

EVOLUTION AND THE THEORY OF GAMES

Please note the following:

- Duration of exam is 2,5 hours;
- Use of lecture notes NOT allowed;

Question 1.

In a 2-person game with strategy set \mathbb{X} let $E(x, y)$ denote the payoff to strategy x against strategy y . Show that if x^* and y^* are both evolutionarily stable strategies (ESS), then the support of x^* is not a subset of the support of y^* .

Question 2.

Two rabbits are sitting side by side in a field. When a fox comes into the field, each rabbit can choose to run or to hide. If one rabbit runs while the other hides, the fox captures the runner with probability P , while the other one gets away quietly. If both rabbits run, then the fox chooses one at random, while the other escapes. If both rabbits keep hiding, the fox eventually finds them anyways, and then both will have to run. The fox chooses one at random, but the capture probability now is $Q > P$, because of the shorter distance. Nice story, but it doesn't really matter: this is a game between two rabbits, and the payoff matrix is given in terms of the escape probability:

	Run	Hide
Run	$1 - \frac{1}{2}P$	$1 - P$
Hide	1	$1 - \frac{1}{2}Q$

(payoffs to the row player)

Find all ESS-s (pure and mixed) depending on the values of P and Q , but always assuming that $0 < P < Q < 1$.

Question 3.

Same situation as in the previous question, but now one rabbit is faster than the other, and they both know it. The capture probability of the faster one is a fraction θ of the capture probability of the slower one. Never mind the story, if you don't like it. The point is that this is an asymmetric game with payoff matrix:

	Run	Hide
Run	$1 - \frac{1}{2}\theta P, 1 - \frac{1}{2}P$	$1 - \theta P, 1$
Hide	$1, 1 - P$	$1 - \frac{1}{2}\theta Q, 1 - \frac{1}{2}Q$

Find all ESS-s depending on the values of P , Q and θ , but always assuming that $0 < P < Q < 1$ and $\theta \in (0, 1)$.

Continued on other side →

Question 4.

Same situation as in Question 2, but now we give each rabbit the choice how close it lets the fox approach before it starts running. Let $s \in [0, 1]$ be the distance between rabbits and the fox. When the fox enters the field, the distance is $s = 1$. The capture probability $\varphi(s)$ is a continuous and decreasing function of the distance with $\varphi(0) = Q$ and $\varphi(1) = P$ and $0 < P < Q < 1$. For arbitrary pure strategies $s_1, s_2 \in [0, 1]$ the payoff to s_1 against s_2 is

$$E(s_1, s_2) = \begin{cases} 1 & \text{if } s_1 < s_2 \\ 1 - \frac{1}{2}\varphi(s_1) & \text{if } s_1 = s_2 \\ 1 - \varphi(s_1) & \text{if } s_1 > s_2. \end{cases}$$

Show that if $P < \frac{1}{2}Q$, then there exists no pure strategy ESS and also not a mixed strategy ESS that can be represented by a probability density $f : [0, 1] \rightarrow \mathbb{R}_+$.

Question 5.

Consider a two-stage Hawk-Dove game where after a $H \times H$ contest the players move to a different stage until the loser has recovered from its injuries while the winner gets the resource for free. The two stages of the game, Γ_1 and Γ_2 , have the following payoff matrices:

Γ_1	Hawk	Dove
Hawk	$\frac{1}{2}(R + \delta \Gamma_2^{\text{row}}) + \frac{1}{2}\delta \Gamma_2^{\text{col}}$	$R + \delta \Gamma_1$
Dove	$\delta \Gamma_1$	$\frac{1}{2}R + \delta \Gamma_1$

(payoffs to the row player)

and

Γ_2	Loser
Winner	$R + (1 - \varepsilon)\delta \Gamma_1 + \varepsilon \delta \Gamma_2^{\text{row}}, (1 - \varepsilon)\delta \Gamma_1 + \varepsilon \delta \Gamma_2^{\text{col}}$

(asymmetric game)

where $R > 0$ is the value of the resource, $\delta \in (0, 1)$ the stage-independent probability of a next round, and $\varepsilon \in (0, 1)$ the probability of recovery in the next round. What is the expected number of rounds of a single play on stage Γ_2 before returning to stage Γ_1 again? Calculate the payoff matrix for the full game $\Gamma = (\Gamma_1, \Gamma_2)$ and find all ESS-s (pure and mixed).

- This exam consists of four problems.
- You have 3 hours to complete the exam.
- You are allowed to bring a calculator.
- **Include intermediate steps and justify your answers.**

Problem 1. This problem consists of two parts.

(1a) Let $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ be a function given by $f(x, y) = \log(1 + e^{2x-4y+3})$. Compute the gradient of f .

(1b) Let $\mathbf{g}: \mathbb{R}^3 \rightarrow \mathbb{R}^2$ be a function given by

$$\mathbf{g}(x, y, z) = \begin{bmatrix} 5 \log(1 + e^{-x+z}) \\ -\log(1 + e^{y+2x}) + z \end{bmatrix}.$$

Compute the Jacobian of \mathbf{g} .

Problem 2. This problem consists of two parts.

(2a) Find all critical points of the function $f: \mathbb{R}^d \rightarrow \mathbb{R}$ given by $f(\mathbf{x}) = \|\mathbf{x}\|^2 - \|\mathbf{x}\|^4$.

(2b) Let $\mathbf{a} \in \mathbb{R}^d$ be fixed. Find all critical points of the function $g: \mathbb{R}^d \rightarrow \mathbb{R}$ given by $g(\mathbf{x}) = \exp(-3\|\mathbf{x}\|^2 + \mathbf{a} \cdot \mathbf{x})$.

Problem 3. Let $\mathbf{y} = (y_1, \dots, y_n) \in \mathbb{R}^n$ be a data vector, $\mathbf{B} \in \mathbb{R}^{n \times d}$ a matrix, and $\lambda > 0$ a scalar. Find the coefficients $\mathbf{a}^* \in \mathbb{R}^d$ that minimise the loss function

$$L(\mathbf{a}) = \|\mathbf{y} - \mathbf{B}\mathbf{a}\|^2 + \lambda \|\mathbf{a}\|^2.$$

Your answer can involve matrix products, inverses, and other operations of linear algebra.

Problem 4. A fair 6-sided die is rolled twice. Let X be a random variable that represents the maximum of the two rolls.

(4a) Write out the probability mass function of X .

(4b) Compute the expected value of X .

[It is not necessary to simplify the expressions or compute their decimal forms.]

The reverse of this sheet contains a collection of potentially useful formulae.

Useful Formulae

$$\frac{d}{dx} x^n = nx^{n-1}$$

$$\frac{d}{dx} c = 0$$

$$\frac{d}{dx} e^x = e^x$$

$$\frac{d}{dx} \sin(x) = \cos(x)$$

$$\frac{d}{dx} [f(x)g(x)] = f'(x)g(x) + f(x)g'(x)$$

$$\frac{d}{dx} f(g(x)) = f'(g(x))g'(x)$$

$$T_n(x) = \sum_{k=0}^n \frac{f^{(k)}(x_0)}{k!} (x-x_0)^k$$

$$\int x^n dx = \frac{1}{n+1} x^{n+1}$$

$$\int x^{-1} dx = \log(x)$$

$$\int \sin(x) dx = -\cos(x)$$

$$\int \cos(x) dx = \sin(x)$$

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \partial_1 f(\mathbf{x}) \\ \vdots \\ \partial_d f(\mathbf{x}) \end{bmatrix} \in \mathbb{R}^d$$

$$\nabla [f(\mathbf{x})g(\mathbf{x})] = g(\mathbf{x})\nabla f(\mathbf{x}) + f(\mathbf{x})\nabla g(\mathbf{x})$$

$$\nabla [f(g(\mathbf{x}))] = f'(g(\mathbf{x}))\nabla g(\mathbf{x})$$

$$\nabla [\mathbf{b}^T \mathbf{A} \mathbf{x}] = \mathbf{A}^T \mathbf{b}$$

$$\nabla [\mathbf{x}^T \mathbf{A} \mathbf{x}] = 2\mathbf{A} \mathbf{x} \quad \text{if } \mathbf{A} \text{ is symmetric}$$

$$\nabla \|\mathbf{x} - \mathbf{a}\| = (\mathbf{x} - \mathbf{a}) / \|\mathbf{x} - \mathbf{a}\|$$

$$\frac{d}{dx} \mathbf{A}(x)^{-1} = -\mathbf{A}(x)^{-1} \left(\frac{d}{dx} \mathbf{A}(x) \right) \mathbf{A}(x)^{-1}$$

$$\mathbf{H}_f(\mathbf{x}) = \mathbf{J}_{\nabla f}(\mathbf{x}) = \begin{bmatrix} \partial_1 \partial_1 f(\mathbf{x}) & \cdots & \partial_1 \partial_d f(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \partial_d \partial_1 f(\mathbf{x}) & \cdots & \partial_d \partial_d f(\mathbf{x}) \end{bmatrix}$$

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{L}(\mathbf{x}) = \mathbf{f}(\mathbf{x}_0) + \mathbf{J}_f(\mathbf{x})(\mathbf{x} - \mathbf{x}_0)$$

$$\int_{\mathbf{g}(A)} f(\mathbf{x}) d\mathbf{x} = \int_A f(\mathbf{g}(\mathbf{y})) |\det(\mathbf{J}_g(\mathbf{y}))| d\mathbf{y}$$

$$P(\cup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P(A_i) \quad \text{if } A_i \text{ are disjoint}$$

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

$$P(A_1 \cap \dots \cap A_n) = P(A_1)P(A_2|A_1)P(A_3|A_1 \cap A_2) \dots P(A_n|A_1 \cap \dots \cap A_{n-1})$$

$$P(A) = \sum_{i=1}^n P(B_i)P(A|B_i) \quad \text{if } B_i \text{ form a partition of } \Omega$$

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

$$P_X(A) = P(X \in A) = P(\{\omega \in \Omega : X(\omega) \in A\})$$

$$p_X(x) = P(X = x) = P(\{\omega \in \Omega : X(\omega) = x\})$$

$$F_X(x) = P(X \leq x) = P(\{\omega \in \Omega : X(\omega) \leq x\})$$

$$X \text{ unif. if } X(\Omega) = \{x_1, \dots, x_k\} \text{ and } p_X(x_k) = \frac{1}{n}$$

$$X \sim \text{Poisson}(\lambda) \text{ if } p_X(k) = e^{-\lambda} \frac{\lambda^k}{k!} \text{ for } k \in \mathbb{N} \cup \{0\}$$

$$\mathbb{E}(X) = \sum_{k=1}^{\infty} x_k p_X(x_k) \text{ when } X(\Omega) = \{x_1, x_2, \dots\}$$

$$\text{Var}(X) = \mathbb{E}[(X - \mathbb{E}(X))^2] = \mathbb{E}(X^2) - [\mathbb{E}(X)]^2$$

$$\text{Var}(aX + b) = a^2 \text{Var}(X)$$

$$\frac{d}{dx} \log(x) = \frac{1}{x}$$

$$\frac{d}{dx} \cos(x) = -\sin(x)$$

$$\frac{d}{dx} \left[\frac{f(x)}{g(x)} \right] = \frac{f'(x)g(x) - f(x)g'(x)}{g(x)^2}$$

$$f(x) \approx L(x) = f(x_0) + f'(x_0)(x - x_0)$$

$$F(b) - F(a) = \int_a^b f(x) dx$$

$$\int e^x dx = e^x$$

$$\int_{g(a)}^{g(b)} f(x) dx = \int_a^b f(g(y))g'(y) dy$$

$$\mathbf{J}_f(\mathbf{x}) = \begin{bmatrix} \partial_1 f_1(\mathbf{x}) & \cdots & \partial_d f_1(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \partial_1 f_m(\mathbf{x}) & \cdots & \partial_d f_m(\mathbf{x}) \end{bmatrix} \in \mathbb{R}^{m \times d}$$

$$\nabla [f(\mathbf{g}(\mathbf{x}))] = \mathbf{J}_g(\mathbf{x})^T \nabla f(\mathbf{g}(\mathbf{x}))$$

$$\nabla [\mathbf{a} \cdot \mathbf{x}] = \mathbf{a}$$

$$\nabla [\mathbf{x}^T \mathbf{A} \mathbf{x}] = (\mathbf{A} + \mathbf{A}^T) \mathbf{x}$$

$$\nabla \|\mathbf{x}\|^2 = 2\mathbf{x}$$

$$\mathbf{J}_f(\mathbf{x}) = \mathbf{A} \quad \text{if } \mathbf{f}(\mathbf{x}) = \mathbf{A} \mathbf{x}$$

$$\frac{d}{dx} \log(\det(\mathbf{A}(x))) = \text{tr} \left(\mathbf{A}(x)^{-1} \frac{d}{dx} \mathbf{A}(x) \right)$$

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{L}(\mathbf{x}) = \mathbf{f}(\mathbf{x}_0) + \nabla f(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{x}_0)$$

$$T_n(\mathbf{x}) = \sum_{|\alpha| \leq n} \frac{D^\alpha f(\mathbf{x}_0)}{\alpha!} (\mathbf{x} - \mathbf{x}_0)^\alpha$$

$$\int_{\mathbb{R}^d} f(\mathbf{A} \mathbf{x} - \mathbf{b}) d\mathbf{x} = |\det(\mathbf{A})|^{-1} \int_{\mathbb{R}^d} f(\mathbf{y}) d\mathbf{y}$$

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

581305-6 Tietokoneen toiminta, 5 op, 12.5.2017

Tämä on kevään 2017 ohjatun itseopiskelukurssin kurssikoe (tehtävät 1-4).
Tehtävät 1-3 ovat minikokeiden 1-3 uusintakokeita. Tehtävä 4 on minikoe 4.
Jos et ole osallistunut aikaisempiin minikokeisiin, vastaa kaikkiin kysymyksiin.

Kirjoita **jokaiseen** vastauspaperiisi seuraavat tiedot: nimi ja nimikirjoitus, opiskelijanumero ja kurssin nimi. Kuhunkin tehtävään riittää 1-2 sivun vastaus.

Kirjoita kunkin tehtävän vastaus **omalle konseptilleen** ja palauta se **omaan pinoonsa!**

1. [10 p] Järjestelmän rakenne, suoritin, väylä.
 - a. [4 p] Mikä on konekäskyjen suoritusyksi? Mitä sen eri vaiheissa tapahtuu?
 - b. [3 p] Laitteisto on suorittamassa ohjelmaa P ja konekäskyn K suoritusaikana tulee tieto, että jokin I/O laite vaatii välittömästi huomiota. Kuinka tämä havaitaan suoritusyksiissä ja kuinka suoritus siirtyy nyt käyttöjärjestelmän koodiin?
 - c. [3 p] Järjestelmässä voi olla käyttöjärjestelmän muistialueita (esim. osoite 2345678), jota käyttöjärjestelmän ohjelmat voivat käsitellä, mutta joita tavalliset ohjelmat (esim. P) eivät. Laitteistossa voi myös olla konekäskyjä (esim. aseta BASE-rekisterin arvo), joita vain käyttöjärjestelmä saa käyttää. Kuinka nämä suojausongelmat on ratkaistu suoritusyksiissä?

2. [10 p] Tiedon esitysmuodot, tiedon tarkistus, muisti.
 - a. [3 p] Mitkä ovat lukujen +11 ja -11 kahden komplementin 32-bittiset Big-Endian esitysmuodot? Entä Little-Endian esitysmuodot?
 - b. [3 p] Mikä on liukulukuesityksen piilobitti ja mitä hyötyä siitä on? Mikä on liukuluvun -0.75 IEEE-standardin mukainen 32-bittinen Big-Endian esitystapa?
 - c. [1 p] Miksi Hamming-koodi on pariteettibittiä parempi muistiväylän suojaamisessa?
 - d. [1 p] Montako ylimääräistä bittiä tarvitaan Hamming-koodissa suojaamaan 72 bitin muistiväylää?
 - e. [2 p] Miksi Hamming-koodi ei sovi verkon tiedonsiirtoväylien suojaamiseen, vaan niissä pitää käyttää esim. CRC:tä (Cyclic Redundancy Code)?

3. [10 p] Käyttöjärjestelmä, ulkoinen muisti, I/O.
 - a. [3 p] Milloin prosessi P siirretään valmis suoritukseen tilasta (ready) suoritustilaan (running)? Kuka sen tekee ja miten siirto tapahtuu?
 - b. [5 p] I/O:n voi toteuttaa käyttäen suoraa I/O:ta (direct I/O), keskeyttävää I/O:ta (epäsuora I/O, indirect I/O, interrupt-driven I/O) ja DMA I/O:ta (Direct Memory Access I/O). Kerro kustakin, kuinka sitä käyttävä laiteajuri (DD) saa tiedon, että sen laiteohjaimelle (laiteohjainprosessille, DC) antama tehtävä on suoritettu. Kerro kustakin, kuka siirtää datan ja kuinka siirrettävä data siirtyy keskusmuistista laitteelle kyseistä I/O:ta käyttäen.
 - c. [2 p] Laiteajuri suorittaa CPU:lla, mutta keskusmuistin lisäksi se voi myös lukea ja kirjoittaa laiteohjaimella olevaa muistia ("laiterekistereitä"). Miten laitteisto tietää, että jotkut viitteet kohdistuvat laiteohjaimen laiterekistereihin eikä keskusmuistiin?

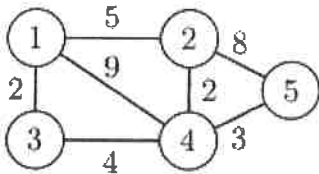
4. [10 p] Käännös, linkitys, lataus, tulkinta, emulointi.
 - a. [2 p] Miksi symbolisen konekielen käännöksen aikana koodi käydään läpi ainakin kaksi kertaa?
 - b. [2 p] Mitä tarkoittaa korkean tason kielen kääntämisen yhteydessä käsite "koodin optimointi"? Mitä siinä oikeastaan optimoidaan?
 - c. [4 p] Mitä etua staattisesta linkityksestä on? Mitä haittaa staattisesta linkityksestä on? Mitä etua dynaamisesta linkityksestä on? Mitä haittaa dynaamisesta linkityksestä on?
 - d. [2 p] Missä Java-ohjelmien suoritustavassa käytetään dynaamista linkitystä? Selitä.

Tentin kesto: 2 h 30 min

Kirjoita jokaisen tehtävän vastaus **erilliselle konseptiarkille**. Merkitse jokaiseen paperiin oma nimesi ja opiskelijanumerosi. Jos käytät lunttilappua, palauta se vastaustesi mukana.

Tehtävä 1 (4 pistettä)

Miten Primin algoritmi käsittelee seuraavan verkon? Näytä sopiva määrä välivaiheita ja lopputulos.



Tehtävä 2 (4 pistettä)

Vastaa lyhyesti jokaiseen seuraavaan kohtaan:

1. Anna esimerkki verkosta, jossa on tasan kolme solmua ja joka on puu.
2. Anna esimerkki verkosta, joka on yhtenäinen mutta ei vahvasti yhtenäinen.
3. Anna esimerkki verkosta, jonka topologinen järjestys on [4, 2, 3, 1].
4. Anna esimerkki verkosta, jonka jokaisen solmun aste on parillinen.
5. Mitä etua leveyshausta on syvyyshakuun verrattuna?
6. Mitä etua syvyyshakuusta on leveyshakuun verrattuna?
7. Miten binäärihakua voi käyttää algoritmien suunnittelussa?
8. Miten dynaamista ohjelmointia voi käyttää repunpakkaus-ongelmassa?

Saat jokaisesta oikeasta vastauksesta 0.5 pistettä, ja tehtävän kokonaispistemäärä on näiden pisteiden summa pyöristettynä alaspäin kokonaisluvuksi.

Tehtävä 3 (6 pistettä)

Sinulle annetaan tiedot kaupungeista ja niiden välisistä kaksisuuntaisista teistä. Jokaisen kahden kaupungin välillä on olemassa jokin reitti teitä pitkin. Tehtäväsi on tutkia, onko mahdollista poistaa jokin tie niin, että poiston jälkeen edelleen jokaisen kahden kaupungin välillä on reitti.

Esimerkiksi jos kaupungit ovat A, B ja C ja tiet ovat A–B, B–C ja A–C, vastaus on "kyllä", koska voit esimerkiksi poistaa tien A–B.

Jos taas kaupungit ovat A, B ja C ja tiet ovat A–B ja A–C, vastaus on "ei", koska molemmat tiet ovat välttämättömiä.

Selosta, miten voit mallintaa ongelman verkkona. Mikä verkon ominaisuus kertoo, onko vastaus ongelmaan "kyllä" vai "ei"?



Tehtävä 4 (6 pistettä)

Tehtäväsi on suunnitella algoritmi, jolle annetaan kokonaisluku n ja joka ilmoittaa, montako tapaa on muodostaa n -merkinen merkkijono, jossa jokainen merkki on 0 tai 1 ja missään kohtaa ei ole kolmea samaa merkkiä peräkkäin.

Esimerkiksi jos n on 4, algoritmin tulee antaa vastaus 10, koska sallitut merkkijonot ovat 0010, 0011, 0100, 0101, 0110, 1001, 1010, 1011, 1100 ja 1101.

Suunnittele tehtävään dynaamisen ohjelmoinnin algoritmi. Algoritmin tulee toimia ajassa $O(n)$ tai tehokkaammin.

Kuvaile ensin algoritmin toimintaidea muutamalla virkkeellä. Anna tämän jälkeen algoritmin koodi ja perustele sen aikavaativuus. Voit käyttää pseudokoodia tai haluamaasi ohjelmointikieltä, ja voit käyttää kurssilla esitettyjä tietorakenteita.

Annoitko jo kurssipalautteen?

Voit antaa kurssipalautteen vielä tänään tai huomenna kurssisivuston ohjeiden mukaisesti. Saat palautteen antamalla yhden ylimääräisen tenttipisteen.

TKT20001 Tietorakenteet ja algoritmit (kevät 2019)

Kurssikoe 1 (4.3.2019)

Tentissä saa olla mukana käsin kirjoitettu yksi A4-kokoinen "lunttilappu", jonka molemmilla puolilla saa olla tekstiä. Muu lisämateriaali (laskimet, taulukot tms.) ei ole sallittua.

Vastaa kuhunkin tehtävään **erilliselle** konseptipaperille.

Kirjoita jokaisen paperin yläkulmaan **kurssin nimi, kokeen päivämäärä, oma nimesi ja opiskelijanumerosi**. Vaikka jättäisit johonkin tehtävään vastaamatta, palauta silti vastauspaperi kyseiseen tehtävään.

Tehtävissä 4 ja 5, joissa pyydetään algoritmia, voit käyttää luentojen ja kurssikirjan tyyppistä pseudokoodia tai muita ymmärrettäviä pseudokoodityylejä tai oikeaa ohjelmointikieltä, esim. Javaa. Jos käytät oikeaa ohjelmointikieltä, selitä erityisen hyvin, mitä ohjelmassasi tapahtuu, äläkä käytä mitään kielen erikoisia piirteitä. Myös sanallinen selitys ilman pseudokoodia kelpaa, kunhan se on riittävän selkeä ja yksityiskohtainen, että vaaditun aikavaativuuden yms. pystyy selvästi toteamaan. Näissä tehtävissä voit käyttää kaikkia kurssilla esitetyjä tietorakenteita ja algoritmeja, niiden tunnettuja aikavaativuuksia jne., kunhan sanot selvästi, mitä milloinkin käytät.

Vastaa kaikkien kysymysten kaikkiin kohtiin. Kokeen maksimipistemäärä on 20.

1. [3 pistettä] Mitä etuja pikajärjestämisellä on lomitusjärjestämiseen verrattuna? Entä toisin päin: mitä etuja lomitusjärjestämisellä on pikajärjestämiseen verrattuna? Millaisessa tilanteessa käyttäisit lisäysjärjestämistä; milloin taas et käyttäisi?

Vastaa lyhyesti ja selkeästi vetoamalla kurssilla esitettyihin algoritmien ominaisuuksiin. Vastauksesi tähän tehtävään tulisi normaalikäsiälalla kirjoitettuna mahtua helposti yhdelle sivulle.

2. [3 pistettä] Mitkä ovat hajautustaulun eri operaatioiden aikavaativuudet? Entä tasapainotetun hakupuun? Millaisissa tilanteissa tämän perusteella hajautustaulu on parempi? Entä milloin tasapainotettu hakupuu on parempi? Esitä myös *yleisellä tasolla ja lyhyesti* (parilla virkkeellä) perusajatus, miten näihin aikavaativuuksiin päädytään.

Vastaa lyhyesti ja selkeästi vetoamalla kurssilla esitettyihin tietorakenteiden ominaisuuksiin. Vastauksesi tähän tehtävään tulisi normaalikäsiälalla kirjoitettuna mahtua helposti yhdelle sivulle.

3. [5 pistettä] Mitkä ovat seuraavien algoritmien aikavaativuudet?

Jokaisessa kohdassa ilmoita aikavaativuus parametrin n funktiona käyttäen iso-O-merkintää ja *selitä* lyhyesti, miten päättelit aikavaativuuden. Selitykseksi riittää virke tai pari, joissa mainitset käyttämäsi yleiset periaatteet tms.; älä esitä tarkkoja matemaattisia todistuksia tms.

Algoritmit eivät tee mitään erityisen hyödyllistä. Kolme pistettä "...” tarkoittaa jotain vakioajassa tapahtuvaa laskentaa.

- (a) for i=1 to n
 ...
 for i=1 to n
 for j=1 to n
 ...
 ...
- (b) for i=1 to n
 for j=1 to i
 for k=1 to j
 ...
 ...
- (c) for i=1 to n
 j = 0
 while j < n
 j = j + n/10 + 1
 ...
 ...
- (d) k = 0
 summa = 0
 while summa < n
 k = k + 1
 summa = summa + k
- (e) rekursiivinen(n)
 if n == 0
 return
 rekursiivinen(n-1)
 rekursiivinen(n-1)
 rekursiivinen(n-1)

4. [5 pistettä] Tarkastellaan tilannetta, jossa m hattua on numeroitu $1, \dots, m$. Hattuihin voidaan lisätä ja niistä voidaan poistaa palloja. Lisäysten ja poistojen yhteislukumäärälle on annettu yläraja n . Aluksi jokainen hattu on tyhjä.

Esitä tehokkaat algoritmit seuraavien operaatioiden toteuttamiseksi:

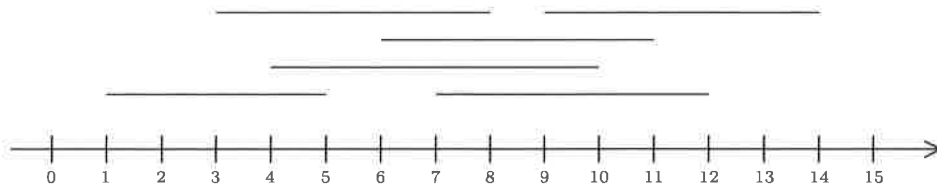
- *lisää(i)*: Lisää yksi pallo hattuun numero i .
- *poista(i)*: Poista yksi pallo hatusta i . Kuitenkin jos hattu i on tyhjä, niin poista yksi pallo numerojärjestyksessä seuraavasta ei-tyhjästä hatusta. Voit olettaa, että jokin seuraavista hatuista sisältää palloja.
- *pallo(i)*: Palauta hatussa i olevien pallojen lukumäärä.

Täysien pisteiden saamiseksi kaikkien operaatioiden tulee toimia pallojen lukumäärän suhteen logaritmisessa ajassa $O(\log n)$ riippumatta siitä, kuinka suuri hattujen lukumäärä m on. Erityisesti aikavaativuuden $O(\log n)$ pitää päteä, kun m on hyvin paljon suurempi kuin n ja siis selvä enemmistö hatuista on tyhjiä.

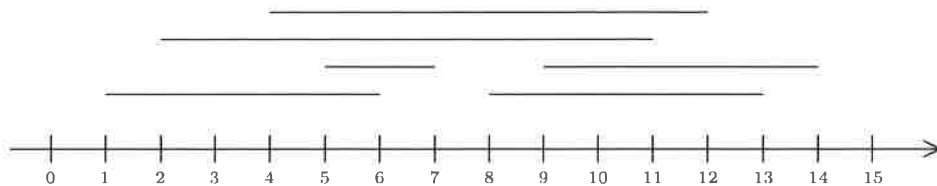
5. [4 pistettä] On annettu joukko lukusuoran välejä $[a, b]$, $0 < a < b$. Tässä $[a, b]$ tarkoittaa suljettua reaalilukuväliä a :sta b :hen, siis pistejoukkoa $\{x \in \mathbb{R} \mid a \leq x \leq b\}$. Tehtävänä on selvittää, onko jokin näistä väleistä jonkin toisen välin aito osaväli; ts. onko joukossa välit $[a, b]$ ja $[c, d]$, joilla $a < c < d < b$. Voit olettaa, että kaikki välien päätepisteet ovat erisuuria kokonaislukuja.

Esitä ongelmaan tehokas ratkaisualgoritmi. Täysien pisteiden saamiseksi algoritmisi tulee toimia ajassa $O(n \log n)$, missä n on välien lukumäärä.

Esimerkki 1: Jos on annettu välit $[1, 5]$, $[7, 12]$, $[4, 10]$, $[6, 11]$, $[3, 8]$ ja $[9, 14]$, niin vastaus on *ei*, sillä mikään väleistä ei sisälly kokonaan toiseen väliin:



Esimerkki 2: Jos on annettu välit $[1, 6]$, $[8, 13]$, $[5, 7]$, $[9, 14]$, $[2, 11]$ ja $[4, 12]$, niin vastaus on *kyllä*, sillä väli $[5, 7]$ sisältyy väliin $[2, 11]$, ja myös väliin $[4, 12]$.



58131 Tietorakenteet ja algoritmit (kevät 2017)

Kurssikoe 1 (6.3.2017)

Tentissä saa olla mukana käsin kirjoitettu yksi A4-kokoinen ”lunttilappu”, jonka molemmilla puolilla saa olla tekstiä.

Vastaa kuhunkin tehtävään erilliselle konseptipaperille. Kirjoita jokaisen paperin yläkulmaan **kurssin nimi, kokeen päivämäärä, nimi, nimikirjoitus ja opiskelijanumero**. Vaikka jättäisit johonkin tehtävään vastaamatta, palauta silti vastauspaperi kyseiseen tehtävään.

Tehtävissä, joissa pyydetään algoritmia, voit käyttää luentojen (Cormenin) tyyppistä pseudokoodia tai muita ymmärrettäviä pseudokoodityylejä tai oikeaa ohjelmointikieltä, esim. Javaa. Jos käytät oikeaa ohjelmointikieltä, selitä erityisen hyvin, mitä ohjelmassasi tapahtuu, äläkä käytä mitään kielen erikoista piirrettä tai valmiita kirjastoja.

Vastaa kaikkien kysymysten kaikkiin kohtiin. Kokeen maksimipistemäärä on 22.

1. [4 pistettä]

(a) Pitääkö paikkansa, että $3n^2 + 5n + 1 = \mathcal{O}(n^3)$?

(b) Pitääkö paikkansa, että $3n^2 + 5n + 1 = \Omega(n^3)$?

Perustele vastauksesi täsmällisesti \mathcal{O} - ja Ω -merkinnän määritelmästä lähtien.

2. [6 pistettä] Syötteenä on annettu kaksi n -alkioista kokonaislukutaulukkoa, A ja B . Tehtävänä on laskea, kuinka moni taulukon B alkioista esiintyy myös taulukossa A . Jos sama kokonaisluku esiintyy useita kertoja taulukossa B , jokainen esiintymä lasketaan mukaan. Useampikertaiset esiintymät taulukossa A eivät vaikuta asiaan.

Esim. jos $n = 10$ ja $A = [8, 2, 11, 5, 3, 7, 3, 9, 5, 11]$ ja $B = [6, 3, 8, 1, 8, 5, 4, 10, 8, 12]$, niin vastaus on 5, koska taulukon B alkioita 3, 8, 8, 5 ja 8 esiintyvät myös taulukossa A .

Esitä ongelmalle algoritmi, jonka pahimman tapauksen aikavaativuus on $\mathcal{O}(n \log n)$. Perustele, että algoritmillasi on tämä aikavaativuus. Mikä on algoritmisi tilavaativuus?

Algoritmille ei tarvitse esittää yksityiskohtaista pseudokoodia, kunhan esität ratkaisuperiaatteen selkeästi niin, että aikavaativuuden voi todeta. Ratkaisussa saa käyttää apuna mitä tahansa kurssilla esitettyjä algoritmeja ja niiden tunnettuja aikavaativuuksia jne., mutta mainitse aina selvästi, mitä algoritmia tai tulosta käytät.

Käännä!

3. [6 pistettä] Sovelluksessa tarvitaan dynaamista joukkoa, jonka avaimet ovat merkkijonoja. Oletetaan, että se toteutetaan hajautustauluna käyttäen jakolaskumenetelmää ja ylivuotolistoja.
- (a) Mitä seikkoja pitää ottaa huomioon talletusalueen koon valinnassa?
 - (b) Selitä yksityiskohtaisesti, miten SEARCH-operaatio toimii, ts. miten annettu merkkijono etsitään rakenteesta.
 - (c) Entä jos ylivuotolistojen sijaan käytettäisiinkin *kaksoishajautusta*: Miten tämä kenties vaikuttaisi talletusalueen koon valintaan? Miten SEARCH nyt toimisi? Mitä etuja ja haittoja tällä olisi ylivuotolistoihin verrattuna?
4. [6 pistettä]
- (a) Anna pseudokoodina algoritmi, joka saa syötteenä osoittimen binääripuun juureen ja palauttaa puun lehtien lukumäärän.
 - (b) Määrittele, millainen on *binäärihakupu*. Mitä ehtoja binäärihakupuun pitää täyttää, että se olisi *AVL-puu*?
 - (c) Anna esimerkki tilanteesta, jossa avaimen lisääminen AVL-puuhun rikkoo AVL-ehdon ja tilanne korjataan suorittamalla kaksoiskierto.

58131 Tietorakenteet ja algoritmit (kevät 2017)

Kurssikoe 2 (10.5.2017)

Tentissä saa olla mukana käsin kirjoitettu yksi A4-kokoinen ”luntilappu”, jonka molemmilla puolilla saa olla tekstiä.

Vastaa kuhunkin tehtävään erilliselle konseptipaperille. Kirjoita jokaisen paperin yläkulmaan **kurssin nimi, kokeen päivämäärä, nimi, nimikirjoitus ja opiskelijanumero**. Vaikka jättäisit johonkin tehtävään vastaamatta, palauta silti vastauspaperi kyseiseen tehtävään.

Tehtävissä, joissa pyydetään algoritmia, voit käyttää luentojen (Cormenin) tyyppistä pseudokoodia tai muita ymmärrettäviä pseudokoodityylejä tai oikeaa ohjelmointikieltä, esim. Javaa. Jos käytät oikeaa ohjelmointikieltä, selitä erityisen hyvin, mitä ohjelmassasi tapahtuu, äläkä käytä mitään kielen erikoista piirrettä tai valmiita kirjastoja.

Vastaa kaikkien kysymysten kaikkiin kohtiin. Kokeen maksimipistemäärä on 22.

1. [4 pistettä]

- (a) Esitä yksityiskohtaisena pseudokoodina, miten minimikeon lisäysoperaatio HEAP-INSERT toimii.
- (b) Esitä pseudokoodina, miten kekojärjestäminen (HEAP-SORT) toimii. Voit olettaa keon perusoperaatiot kuten HEAPIFY annetuiksi, niiden pseudokoodia ei tarvitse esittää. Mitä hyviä ja huonoja puolia kekojärjestämisellä on muihin järjestämisalgoritmeihin verrattuna?

2. [6 pistettä] Syötteenä on annettu suunnattu painottamaton verkko $G = (V, E)$, jonka jokainen solmu on väritetty jollain k eri väristä, sekä solmut s ja t . Tavoitteena olisi löytää lyhin sellainen polku $s \rightsquigarrow t$, joka ei missään vaiheessa kulje kolmea kertaa *peräkkäin* samanvärisen solmun kautta.

- (a) Ongelmaan on ehdotettu seuraavanlaista ratkaisua: Algoritmi toimii muuten kuten leveyshaku, mutta jokaisen solmun kohdalla otetaan talteen tieto siitä, onko siihen saavuttu samanvärisestä solmusta kuin se itse on. Jos tämä pitää paikkaansa, ei vieruslistaa käydessä oteta huomioon kaaria, jotka johtavat samanväriseen solmuun.
Osoita että algoritmi ei toimi antamalla esimerkki verkosta G ja solmuista s, t , joilla algoritmi päättyy väärään tulokseen. Vastauksessa tulee simuloida pääpiirteittäin algoritmin toiminta annetulla syötteellä ja näyttää, että algoritmin tuottama vastaus todellakin on väärin.
- (b) Suunnittele algoritmi, joka löytää lyhimmän kelvollisen polun $s \rightsquigarrow t$ ajassa $O(|V| + |E|)$. Perustele aikavaativuus.

Käännä!

3. [6 pistettä] Kuten tiedämme, Dijkstran algoritmi etsii painotetussa verkossa kaikki lyhimmät polut annetusta lähtösolmusta.
- (a) Selitä Dijkstran algoritmin toimintaperiaate. Sopiva kuvauksen taso on muutama virkkeen selitys algoritmin toimintaperiaatteesta ja sen havainnollistaminen lyhyellä esimerkkisimulaatiolla. Yksityiskohtaista pseudokoodia ei tarvitse esittää. Mainitse kuitenkin, mitä keskeisiä aputietorakenteita algoritmi käyttää.
 - (b) Oletetaan nyt, että suunnatussa verkossa paitsi kaarilla myös solmuilla on paino. Kaikki painot ovat positiivisia kokonaislukuja. Polun pituudeksi määritellään sillä olevien solmujen ja kaarten painojen summa, mukaanlukien alku- ja loppusolmu. Miten muokkaisit tai soveltaisit Dijkstran algoritmia lyhimpien polkujen löytämiseksi tässä tilanteessa? Ratkaisuperiaatteen sanallinen selitys riittää, pseudokoodia ei tarvitse esittää.
4. [6 pistettä] Ilmoita kummastakin seuraavasta väitteestä, pitääkö se paikkansa. Perustele vastauksesi täsmällisesti. Kummassakin kohdassa oletetaan, että G on painotettu suuntaamaton verkko, jonka kaikilla kaarilla on eri paino.
- (a) Jos kahden solmun s ja t välinen lyhin polku sisältää kaaren e , niin kaari e sisältyy myös verkon G pienimpään virittävään puuhun.
 - (b) Jos e on jonkin verkon G syklin kaarista painoltaan suurin, niin se ei kuulu verkon G pienimpään virittävään puuhun.

582206 Models of Computation (Autumn 2016)
3rd exam (23.11.)

To make grading faster, please use a separate sheet of paper for answering each of the problems 1, 2 and 3. Write your own name and student number clearly on top of each sheet.

Answer all parts of all problems. The maximum score for the exam is 12 points.

- (4 points) The language A over the alphabet $\Sigma = \{a, b\}$ consists of all strings that contain more 'b' than 'a' symbols. Give as a state diagram a deterministic Turing machine that recognizes A . If you wish, you may use a multi-tape machine.
- (4 points) Let G be the following context-free grammar:

$$S \rightarrow aSb \mid T$$

$$T \rightarrow cT \mid Td \mid \epsilon$$

Using the method given in the textbook, convert G into a push-down automaton that recognizes the language $L(G)$. Give the PDA as a full state diagram showing all states and transitions. (In particular, do not use any shorthand for pushing several symbols to the stack.) Simulate the computation of your PDA when it accepts the string $aacdbb$.

- (4 points) You are given a context-free grammar G and a natural number k . You may assume that G is in Chomsky normal form. The problem is to decide whether G generates at least one string that has length exactly k . Give an algorithm that solves this problem. Your algorithm does not need to be particularly efficient.

In your solution, give the algorithm using pseudocode or other suitable representation, and also explain briefly its main idea. In particular, point out how the assumption about Chomsky normal form is used (if it is used). However, you don't need to prove the correctness of your algorithm.

From To
1 2 8
2 4 32 64
4 8 32 64
8 32 64
16 64
32 64
64

if need may be
combine and replace
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100]



582206 Laskennan mallit (syksy 2016)
 3. välikoe (23.11.)

Tarkastamisen nopeuttamiseksi vastaa kuhunkin kysymyksistä 1, 2 ja 3 omalle konseptiarkilleen. Kirjoita oma nimesi ja opiskelijanumerosi selvästi kunkin konseptiarkin yläreunaan.

Vastaa kaikkien tehtävien kaikkiin kohtiin. Kokeen maksimipistemäärä on 12 pistettä.

- (4 pistettä) Aakkoston $\Sigma = \{a, b\}$ kieli A koostuu kaikista merkkijonoista, joissa b -merkkejä on enemmän kuin a -merkkejä. Esitä tilakaavio deterministiselle Turingin koneelle, joka tunnistaa kielen A . Voit halutessasi käyttää moninauhaista Turingin konetta.
- (4 pistettä) Olkoon G seuraava yhteydetön kielioppi:

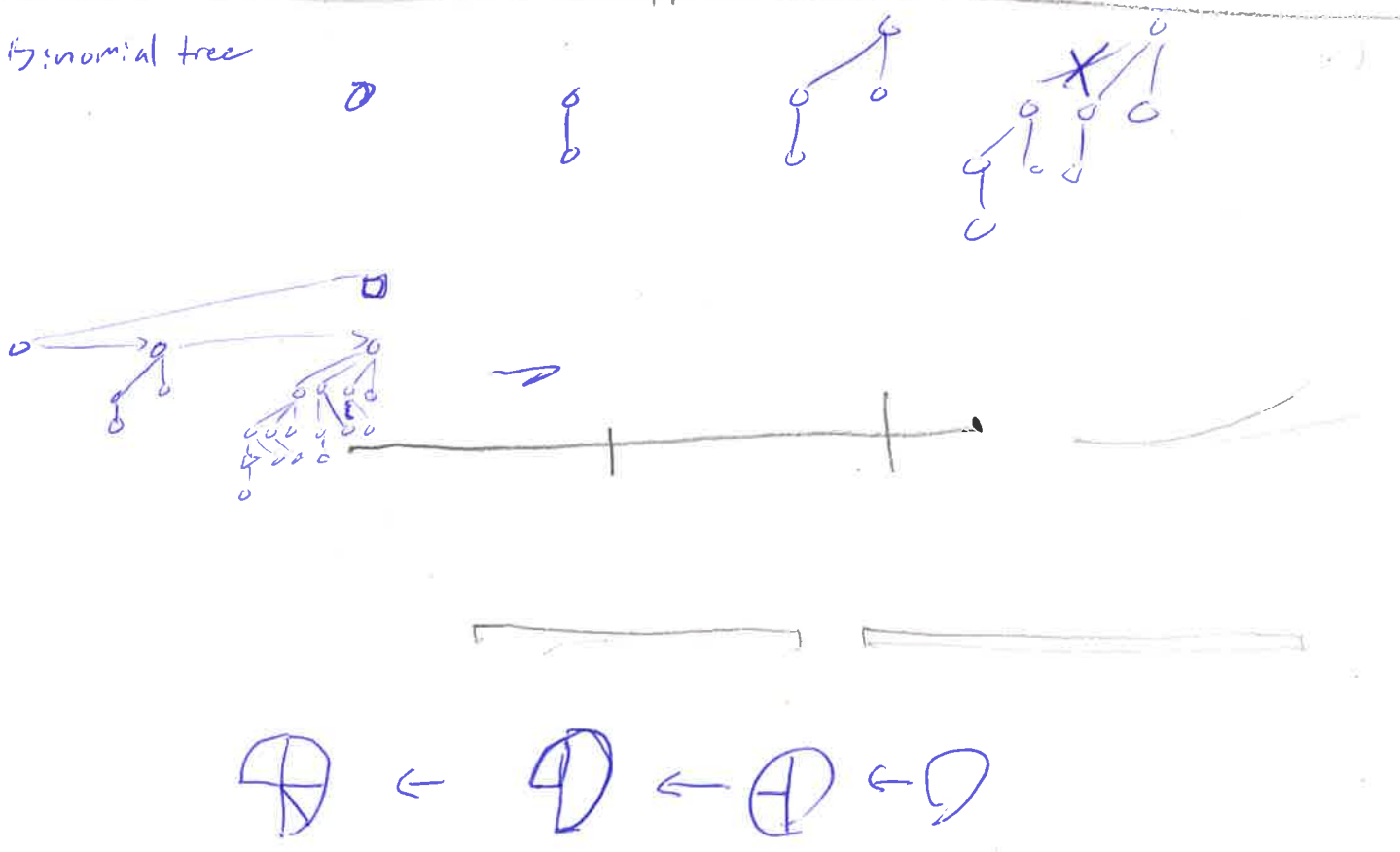
$$\begin{aligned} S &\rightarrow aSb \mid T \\ T &\rightarrow cT \mid Td \mid \varepsilon \end{aligned}$$

Muodosta kurssilla esitetyllä menetelmällä kieliopin G perusteella pinoautomaatti, joka tunnistaa kielen $L(G)$. Esitä automaatti täydellisenä kaikkine tiloineen ja siirtymineen. (Siis erityisesti yhdessä siirtymässä viedään vain yksi merkki pinoon.) Simuloi muodostamasi automaatin laskenta, kun se hyväksyy merkkijonon $aacdbb$.

- (4 pistettä) Syötteenä on annettu yhteydetön kielioppi G ja luonnollinen luku k . Kieliopista G oletetaan, että se on Chomskyn normaalimuodossa. Ongelmana on päättää, sisältääkö kieli $L(G)$ jonkin tasan k merkkiä pitkän merkkijonon. Esitä tähän ongelmaan ratkaisualgoritmi. Algoritmin ei tarvitse olla erityisen tehokas.

Ratkaisussasi tulee olla algoritmin pseudokoodi- tms. esityksen lisäksi lyhyt selitys sen toimintaperiaatteesta. Osoita selityksessä erityisesti, miten oletusta Chomskyn normaalimuodosta käytetään hyväksi (jos käytetään). Algoritmin oikeellisuutta ei kuitenkaan tarvitse todistaa.

Binomial tree



582206 Laskennan mallit (syksy 2016)
4. välikoe (14.12.)

Tarkastamisen nopeuttamiseksi vastaa kuhunkin kysymyksistä 1, 2 ja 3 omalle konseptiarkilleen. Kirjoita oma nimesi ja opiskelijanumerosi selvästi kunkin konseptiarkin yläreunaan.

Vastaa kaikkien tehtävien kaikkiin kohtiin. Kokeen maksimipistemäärä on 12 pistettä.

1. (4 pistettä) Suuntaamattoman verkon $G = (V, E)$ riippumaton joukko on sellainen solmujoukko, että minkään kahden joukkoon kuuluvan solmun välillä ei ole kaarta. Siis joukko $I \subseteq V$ on riippumaton, jos $(u, v) \notin E$ kaikilla $u, v \in I$.

Riippumattomiin joukkoihin liittyvä formaali kieli *INDEPENDENT-SET* sisältää sopivasti koodattuina kaikki sellaiset verkon G ja luonnollisen luvun k muodostamat parit, että verkossa G on k -solmuinen riippumaton joukko. Siis

$$INDEPENDENT-SET = \{ \langle G, k \rangle \mid \text{verkossa } G \text{ on riippumaton joukko, jossa on } k \text{ solmua.} \}$$

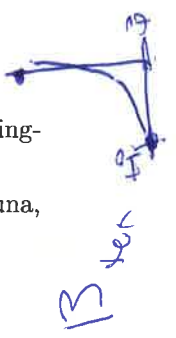
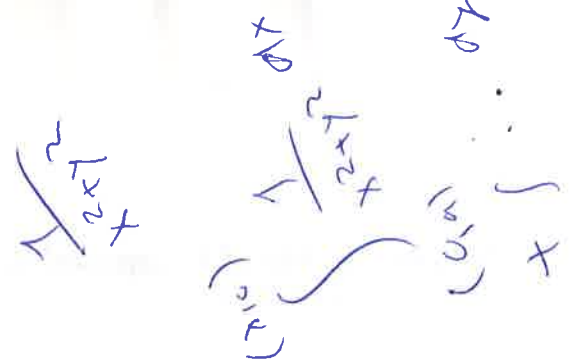
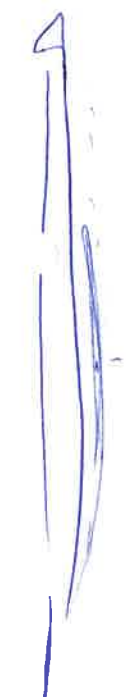
Osoita, että *INDEPENDENT-SET* \in NP. Esitä tarvittava algoritmi pseudokoodin tasolla menemättä Turingin koneen nauhoihin tai muihin vastaaviin yksityiskohtiin.

2. (4 pistettä)

- (a) Esitä täsmällinen määritelmä sille, että kieli on NP-täydellinen.
(b) Klikkiongelma (*CLIQUE*) on eräs kurssilta tuttu NP-täydellinen ongelma. Osoita, että jos *CLIQUE* \in P, niin $P = NP$. Perustele yksityiskohtaisesti lähtien NP-täydellisyysmääritelmästä ja tiedosta, että *CLIQUE* on NP-täydellinen.

3. (4 pistettä)

- (a) Määrittele Turingin koneen pysähtymisongelma $HALT_{TM}$. Onko $HALT_{TM}$ Turing-tunnistettava? Perustele.
(b) Todista, että Turingin koneen pysähtymisongelma on ratkeamaton. Voit pitää tunnettuna, että kieli $D = \{ \langle M \rangle \mid \langle M \rangle \notin L(M) \}$ ei ole Turing-tunnistettava.



How dare you?

$\frac{1}{\delta} = \frac{1}{\delta}$

$\frac{1}{\delta} = \frac{1}{\delta}$

$\frac{1}{\delta} = \frac{1}{\delta}$

$\frac{1}{\delta} = \frac{1}{\delta}$

V_i	d_i	V_i	D_i	V_i	DE	V_i	DE
-8	12	6	8	14	0	14	0
-1	19	8	13	20	8	21	7
6	26	7	26	33	9	29	13
19	27	1	34	47	9	31	25
11	39	-13	41	60	10	37	33
19	45	1	41	68	2	48	36
11	55	14	42			62	36
24	56	22	35				
36	58	35	36				
	66	42	34				
		56	44				
		56	58				

582206 Models of Computation (Autumn 2016)
4th exam (14.12.)

To make grading faster, please use a separate sheet of paper for answering each of the problems 1, 2 and 3. Write your own name and student number clearly on top of each sheet.

Answer all parts of all problems. The maximum score for the exam is 12 points.

1. (4 points) An *independent set* in an undirected graph $G = (V, E)$ is a set of vertices such that there are no edges between vertices in the set. That is, a set of vertices $I \subseteq V$ is independent if $(u, v) \notin E$ for all $u, v \in I$.

We define a related formal language *INDEPENDENT-SET* to consist of suitably encoded pairs of a graph G and natural number k such that G has an independent set of k vertices. That is,

$$INDEPENDENT-SET = \{ \langle G, k \rangle \mid G \text{ has an independent set of } k \text{ vertices.} \}$$

Show that *INDEPENDENT-SET* \in NP. Give the required algorithm using pseudocode without going into details such as the tapes of the Turing machine.

2. (4 points)

- (a) Define precisely what it means for a language to be NP-complete.
 (b) The clique problem (*CLIQUE*) is a familiar NP-complete problem from the course. Show that if *CLIQUE* \in P, then $P = NP$. Give a detailed proof starting from the definition of NP-completeness and the known fact that *CLIQUE* is NP-complete.

3. (4 points)

- (a) Define the Turing machine halting problem *HALT_{TM}*. Is *HALT_{TM}* Turing-recognizable? Justify your answer.
 (b) Prove that the Turing machine halting problem is undecidable. You may use the known fact that the language $D = \{ \langle M \rangle \mid \langle M \rangle \notin L(M) \}$ is not Turing-recognizable.

Handwritten notes:
 $\frac{d}{dx} x^2 = 2x$
 $\frac{d}{dx} x^3 = 3x^2$
 $\frac{d}{dx} x^n = nx^{n-1}$
 $\frac{d}{dx} \frac{1}{x} = -\frac{1}{x^2}$
 $\frac{d}{dx} \ln x = \frac{1}{x}$
 $\frac{d}{dx} e^x = e^x$
 $\frac{d}{dx} a^x = a^x \ln a$
 $\frac{d}{dx} x^a = ax^{a-1}$
 $\frac{d}{dx} x^x = x^x (1 + \ln x)$
 $\frac{d}{dx} \frac{1}{x^2} = -\frac{2}{x^3}$
 $\frac{d}{dx} \frac{1}{x^3} = -\frac{3}{x^4}$
 $\frac{d}{dx} \frac{1}{x^4} = -\frac{4}{x^5}$
 $\frac{d}{dx} \frac{1}{x^5} = -\frac{5}{x^6}$
 $\frac{d}{dx} \frac{1}{x^6} = -\frac{6}{x^7}$
 $\frac{d}{dx} \frac{1}{x^7} = -\frac{7}{x^8}$
 $\frac{d}{dx} \frac{1}{x^8} = -\frac{8}{x^9}$
 $\frac{d}{dx} \frac{1}{x^9} = -\frac{9}{x^{10}}$
 $\frac{d}{dx} \frac{1}{x^{10}} = -\frac{10}{x^{11}}$
 $\frac{d}{dx} \frac{1}{x^{11}} = -\frac{11}{x^{12}}$
 $\frac{d}{dx} \frac{1}{x^{12}} = -\frac{12}{x^{13}}$
 $\frac{d}{dx} \frac{1}{x^{13}} = -\frac{13}{x^{14}}$
 $\frac{d}{dx} \frac{1}{x^{14}} = -\frac{14}{x^{15}}$
 $\frac{d}{dx} \frac{1}{x^{15}} = -\frac{15}{x^{16}}$
 $\frac{d}{dx} \frac{1}{x^{16}} = -\frac{16}{x^{17}}$
 $\frac{d}{dx} \frac{1}{x^{17}} = -\frac{17}{x^{18}}$
 $\frac{d}{dx} \frac{1}{x^{18}} = -\frac{18}{x^{19}}$
 $\frac{d}{dx} \frac{1}{x^{19}} = -\frac{19}{x^{20}}$
 $\frac{d}{dx} \frac{1}{x^{20}} = -\frac{20}{x^{21}}$
 $\frac{d}{dx} \frac{1}{x^{21}} = -\frac{21}{x^{22}}$
 $\frac{d}{dx} \frac{1}{x^{22}} = -\frac{22}{x^{23}}$
 $\frac{d}{dx} \frac{1}{x^{23}} = -\frac{23}{x^{24}}$
 $\frac{d}{dx} \frac{1}{x^{24}} = -\frac{24}{x^{25}}$
 $\frac{d}{dx} \frac{1}{x^{25}} = -\frac{25}{x^{26}}$
 $\frac{d}{dx} \frac{1}{x^{26}} = -\frac{26}{x^{27}}$
 $\frac{d}{dx} \frac{1}{x^{27}} = -\frac{27}{x^{28}}$
 $\frac{d}{dx} \frac{1}{x^{28}} = -\frac{28}{x^{29}}$
 $\frac{d}{dx} \frac{1}{x^{29}} = -\frac{29}{x^{30}}$
 $\frac{d}{dx} \frac{1}{x^{30}} = -\frac{30}{x^{31}}$
 $\frac{d}{dx} \frac{1}{x^{31}} = -\frac{31}{x^{32}}$
 $\frac{d}{dx} \frac{1}{x^{32}} = -\frac{32}{x^{33}}$
 $\frac{d}{dx} \frac{1}{x^{33}} = -\frac{33}{x^{34}}$
 $\frac{d}{dx} \frac{1}{x^{34}} = -\frac{34}{x^{35}}$
 $\frac{d}{dx} \frac{1}{x^{35}} = -\frac{35}{x^{36}}$
 $\frac{d}{dx} \frac{1}{x^{36}} = -\frac{36}{x^{37}}$
 $\frac{d}{dx} \frac{1}{x^{37}} = -\frac{37}{x^{38}}$
 $\frac{d}{dx} \frac{1}{x^{38}} = -\frac{38}{x^{39}}$
 $\frac{d}{dx} \frac{1}{x^{39}} = -\frac{39}{x^{40}}$
 $\frac{d}{dx} \frac{1}{x^{40}} = -\frac{40}{x^{41}}$
 $\frac{d}{dx} \frac{1}{x^{41}} = -\frac{41}{x^{42}}$
 $\frac{d}{dx} \frac{1}{x^{42}} = -\frac{42}{x^{43}}$
 $\frac{d}{dx} \frac{1}{x^{43}} = -\frac{43}{x^{44}}$
 $\frac{d}{dx} \frac{1}{x^{44}} = -\frac{44}{x^{45}}$
 $\frac{d}{dx} \frac{1}{x^{45}} = -\frac{45}{x^{46}}$
 $\frac{d}{dx} \frac{1}{x^{46}} = -\frac{46}{x^{47}}$
 $\frac{d}{dx} \frac{1}{x^{47}} = -\frac{47}{x^{48}}$
 $\frac{d}{dx} \frac{1}{x^{48}} = -\frac{48}{x^{49}}$
 $\frac{d}{dx} \frac{1}{x^{49}} = -\frac{49}{x^{50}}$
 $\frac{d}{dx} \frac{1}{x^{50}} = -\frac{50}{x^{51}}$
 $\frac{d}{dx} \frac{1}{x^{51}} = -\frac{51}{x^{52}}$
 $\frac{d}{dx} \frac{1}{x^{52}} = -\frac{52}{x^{53}}$
 $\frac{d}{dx} \frac{1}{x^{53}} = -\frac{53}{x^{54}}$
 $\frac{d}{dx} \frac{1}{x^{54}} = -\frac{54}{x^{55}}$
 $\frac{d}{dx} \frac{1}{x^{55}} = -\frac{55}{x^{56}}$
 $\frac{d}{dx} \frac{1}{x^{56}} = -\frac{56}{x^{57}}$
 $\frac{d}{dx} \frac{1}{x^{57}} = -\frac{57}{x^{58}}$
 $\frac{d}{dx} \frac{1}{x^{58}} = -\frac{58}{x^{59}}$
 $\frac{d}{dx} \frac{1}{x^{59}} = -\frac{59}{x^{60}}$
 $\frac{d}{dx} \frac{1}{x^{60}} = -\frac{60}{x^{61}}$
 $\frac{d}{dx} \frac{1}{x^{61}} = -\frac{61}{x^{62}}$
 $\frac{d}{dx} \frac{1}{x^{62}} = -\frac{62}{x^{63}}$
 $\frac{d}{dx} \frac{1}{x^{63}} = -\frac{63}{x^{64}}$
 $\frac{d}{dx} \frac{1}{x^{64}} = -\frac{64}{x^{65}}$
 $\frac{d}{dx} \frac{1}{x^{65}} = -\frac{65}{x^{66}}$
 $\frac{d}{dx} \frac{1}{x^{66}} = -\frac{66}{x^{67}}$
 $\frac{d}{dx} \frac{1}{x^{67}} = -\frac{67}{x^{68}}$
 $\frac{d}{dx} \frac{1}{x^{68}} = -\frac{68}{x^{69}}$
 $\frac{d}{dx} \frac{1}{x^{69}} = -\frac{69}{x^{70}}$
 $\frac{d}{dx} \frac{1}{x^{70}} = -\frac{70}{x^{71}}$
 $\frac{d}{dx} \frac{1}{x^{71}} = -\frac{71}{x^{72}}$
 $\frac{d}{dx} \frac{1}{x^{72}} = -\frac{72}{x^{73}}$
 $\frac{d}{dx} \frac{1}{x^{73}} = -\frac{73}{x^{74}}$
 $\frac{d}{dx} \frac{1}{x^{74}} = -\frac{74}{x^{75}}$
 $\frac{d}{dx} \frac{1}{x^{75}} = -\frac{75}{x^{76}}$
 $\frac{d}{dx} \frac{1}{x^{76}} = -\frac{76}{x^{77}}$
 $\frac{d}{dx} \frac{1}{x^{77}} = -\frac{77}{x^{78}}$
 $\frac{d}{dx} \frac{1}{x^{78}} = -\frac{78}{x^{79}}$
 $\frac{d}{dx} \frac{1}{x^{79}} = -\frac{79}{x^{80}}$
 $\frac{d}{dx} \frac{1}{x^{80}} = -\frac{80}{x^{81}}$
 $\frac{d}{dx} \frac{1}{x^{81}} = -\frac{81}{x^{82}}$
 $\frac{d}{dx} \frac{1}{x^{82}} = -\frac{82}{x^{83}}$
 $\frac{d}{dx} \frac{1}{x^{83}} = -\frac{83}{x^{84}}$
 $\frac{d}{dx} \frac{1}{x^{84}} = -\frac{84}{x^{85}}$
 $\frac{d}{dx} \frac{1}{x^{85}} = -\frac{85}{x^{86}}$
 $\frac{d}{dx} \frac{1}{x^{86}} = -\frac{86}{x^{87}}$
 $\frac{d}{dx} \frac{1}{x^{87}} = -\frac{87}{x^{88}}$
 $\frac{d}{dx} \frac{1}{x^{88}} = -\frac{88}{x^{89}}$
 $\frac{d}{dx} \frac{1}{x^{89}} = -\frac{89}{x^{90}}$
 $\frac{d}{dx} \frac{1}{x^{90}} = -\frac{90}{x^{91}}$
 $\frac{d}{dx} \frac{1}{x^{91}} = -\frac{91}{x^{92}}$
 $\frac{d}{dx} \frac{1}{x^{92}} = -\frac{92}{x^{93}}$
 $\frac{d}{dx} \frac{1}{x^{93}} = -\frac{93}{x^{94}}$
 $\frac{d}{dx} \frac{1}{x^{94}} = -\frac{94}{x^{95}}$
 $\frac{d}{dx} \frac{1}{x^{95}} = -\frac{95}{x^{96}}$
 $\frac{d}{dx} \frac{1}{x^{96}} = -\frac{96}{x^{97}}$
 $\frac{d}{dx} \frac{1}{x^{97}} = -\frac{97}{x^{98}}$
 $\frac{d}{dx} \frac{1}{x^{98}} = -\frac{98}{x^{99}}$
 $\frac{d}{dx} \frac{1}{x^{99}} = -\frac{99}{x^{100}}$
 $\frac{d}{dx} \frac{1}{x^{100}} = -\frac{100}{x^{101}}$
 $\frac{d}{dx} \frac{1}{x^{101}} = -\frac{101}{x^{102}}$
 $\frac{d}{dx} \frac{1}{x^{102}} = -\frac{102}{x^{103}}$
 $\frac{d}{dx} \frac{1}{x^{103}} = -\frac{103}{x^{104}}$
 $\frac{d}{dx} \frac{1}{x^{104}} = -\frac{104}{x^{105}}$
 $\frac{d}{dx} \frac{1}{x^{105}} = -\frac{105}{x^{106}}$
 $\frac{d}{dx} \frac{1}{x^{106}} = -\frac{106}{x^{107}}$
 $\frac{d}{dx} \frac{1}{x^{107}} = -\frac{107}{x^{108}}$
 $\frac{d}{dx} \frac{1}{x^{108}} = -\frac{108}{x^{109}}$
 $\frac{d}{dx} \frac{1}{x^{109}} = -\frac{109}{x^{110}}$
 $\frac{d}{dx} \frac{1}{x^{110}} = -\frac{110}{x^{111}}$
 $\frac{d}{dx} \frac{1}{x^{111}} = -\frac{111}{x^{112}}$
 $\frac{d}{dx} \frac{1}{x^{112}} = -\frac{112}{x^{113}}$
 $\frac{d}{dx} \frac{1}{x^{113}} = -\frac{113}{x^{114}}$
 $\frac{d}{dx} \frac{1}{x^{114}} = -\frac{114}{x^{115}}$
 $\frac{d}{dx} \frac{1}{x^{115}} = -\frac{115}{x^{116}}$
 $\frac{d}{dx} \frac{1}{x^{116}} = -\frac{116}{x^{117}}$
 $\frac{d}{dx} \frac{1}{x^{117}} = -\frac{117}{x^{118}}$
 $\frac{d}{dx} \frac{1}{x^{118}} = -\frac{118}{x^{119}}$
 $\frac{d}{dx} \frac{1}{x^{119}} = -\frac{119}{x^{120}}$
 $\frac{d}{dx} \frac{1}{x^{120}} = -\frac{120}{x^{121}}$
 $\frac{d}{dx} \frac{1}{x^{121}} = -\frac{121}{x^{122}}$
 $\frac{d}{dx} \frac{1}{x^{122}} = -\frac{122}{x^{123}}$
 $\frac{d}{dx} \frac{1}{x^{123}} = -\frac{123}{x^{124}}$
 $\frac{d}{dx} \frac{1}{x^{124}} = -\frac{124}{x^{125}}$
 $\frac{d}{dx} \frac{1}{x^{125}} = -\frac{125}{x^{126}}$
 $\frac{d}{dx} \frac{1}{x^{126}} = -\frac{126}{x^{127}}$
 $\frac{d}{dx} \frac{1}{x^{127}} = -\frac{127}{x^{128}}$
 $\frac{d}{dx} \frac{1}{x^{128}} = -\frac{128}{x^{129}}$
 $\frac{d}{dx} \frac{1}{x^{129}} = -\frac{129}{x^{130}}$
 $\frac{d}{dx} \frac{1}{x^{130}} = -\frac{130}{x^{131}}$
 $\frac{d}{dx} \frac{1}{x^{131}} = -\frac{131}{x^{132}}$
 $\frac{d}{dx} \frac{1}{x^{132}} = -\frac{132}{x^{133}}$
 $\frac{d}{dx} \frac{1}{x^{133}} = -\frac{133}{x^{134}}$
 $\frac{d}{dx} \frac{1}{x^{134}} = -\frac{134}{x^{135}}$
 $\frac{d}{dx} \frac{1}{x^{135}} = -\frac{135}{x^{136}}$
 $\frac{d}{dx} \frac{1}{x^{136}} = -\frac{136}{x^{137}}$
 $\frac{d}{dx} \frac{1}{x^{137}} = -\frac{137}{x^{138}}$
 $\frac{d}{dx} \frac{1}{x^{138}} = -\frac{138}{x^{139}}$
 $\frac{d}{dx} \frac{1}{x^{139}} = -\frac{139}{x^{140}}$
 $\frac{d}{dx} \frac{1}{x^{140}} = -\frac{140}{x^{141}}$
 $\frac{d}{dx} \frac{1}{x^{141}} = -\frac{141}{x^{142}}$
 $\frac{d}{dx} \frac{1}{x^{142}} = -\frac{142}{x^{143}}$
 $\frac{d}{dx} \frac{1}{x^{143}} = -\frac{143}{x^{144}}$
 $\frac{d}{dx} \frac{1}{x^{144}} = -\frac{144}{x^{145}}$
 $\frac{d}{dx} \frac{1}{x^{145}} = -\frac{145}{x^{146}}$
 $\frac{d}{dx} \frac{1}{x^{146}} = -\frac{146}{x^{147}}$
 $\frac{d}{dx} \frac{1}{x^{147}} = -\frac{147}{x^{148}}$
 $\frac{d}{dx} \frac{1}{x^{148}} = -\frac{148}{x^{149}}$
 $\frac{d}{dx} \frac{1}{x^{149}} = -\frac{149}{x^{150}}$
 $\frac{d}{dx} \frac{1}{x^{150}} = -\frac{150}{x^{151}}$
 $\frac{d}{dx} \frac{1}{x^{151}} = -\frac{151}{x^{152}}$
 $\frac{d}{dx} \frac{1}{x^{152}} = -\frac{152}{x^{153}}$
 $\frac{d}{dx} \frac{1}{x^{153}} = -\frac{153}{x^{154}}$
 $\frac{d}{dx} \frac{1}{x^{154}} = -\frac{154}{x^{155}}$
 $\frac{d}{dx} \frac{1}{x^{155}} = -\frac{155}{x^{156}}$
 $\frac{d}{dx} \frac{1}{x^{156}} = -\frac{156}{x^{157}}$
 $\frac{d}{dx} \frac{1}{x^{157}} = -\frac{157}{x^{158}}$
 $\frac{d}{dx} \frac{1}{x^{158}} = -\frac{158}{x^{159}}$
 $\frac{d}{dx} \frac{1}{x^{159}} = -\frac{159}{x^{160}}$
 $\frac{d}{dx} \frac{1}{x^{160}} = -\frac{160}{x^{161}}$
 $\frac{d}{dx} \frac{1}{x^{161}} = -\frac{161}{x^{162}}$
 $\frac{d}{dx} \frac{1}{x^{162}} = -\frac{162}{x^{163}}$
 $\frac{d}{dx} \frac{1}{x^{163}} = -\frac{163}{x^{164}}$
 $\frac{d}{dx} \frac{1}{x^{164}} = -\frac{164}{x^{165}}$
 $\frac{d}{dx} \frac{1}{x^{165}} = -\frac{165}{x^{166}}$
 $\frac{d}{dx} \frac{1}{x^{166}} = -\frac{166}{x^{167}}$
 $\frac{d}{dx} \frac{1}{x^{167}} = -\frac{167}{x^{168}}$
 $\frac{d}{dx} \frac{1}{x^{168}} = -\frac{168}{x^{169}}$
 $\frac{d}{dx} \frac{1}{x^{169}} = -\frac{169}{x^{170}}$
 $\frac{d}{dx} \frac{1}{x^{170}} = -\frac{170}{x^{171}}$
 $\frac{d}{dx} \frac{1}{x^{171}} = -\frac{171}{x^{172}}$
 $\frac{d}{dx} \frac{1}{x^{172}} = -\frac{172}{x^{173}}$
 $\frac{d}{dx} \frac{1}{x^{173}} = -\frac{173}{x^{174}}$
 $\frac{d}{dx} \frac{1}{x^{174}} = -\frac{174}{x^{175}}$
 $\frac{d}{dx} \frac{1}{x^{175}} = -\frac{175}{x^{176}}$
 $\frac{d}{dx} \frac{1}{x^{176}} = -\frac{176}{x^{177}}$
 $\frac{d}{dx} \frac{1}{x^{177}} = -\frac{177}{x^{178}}$
 $\frac{d}{dx} \frac{1}{x^{178}} = -\frac{178}{x^{179}}$
 $\frac{d}{dx} \frac{1}{x^{179}} = -\frac{179}{x^{180}}$
 $\frac{d}{dx} \frac{1}{x^{180}} = -\frac{180}{x^{181}}$
 $\frac{d}{dx} \frac{1}{x^{181}} = -\frac{181}{x^{182}}$
 $\frac{d}{dx} \frac{1}{x^{182}} = -\frac{182}{x^{183}}$
 $\frac{d}{dx} \frac{1}{x^{183}} = -\frac{183}{x^{184}}$
 $\frac{d}{dx} \frac{1}{x^{184}} = -\frac{184}{x^{185}}$
 $\frac{d}{dx} \frac{1}{x^{185}} = -\frac{185}{x^{186}}$
 $\frac{d}{dx} \frac{1}{x^{186}} = -\frac{186}{x^{187}}$
 $\frac{d}{dx} \frac{1}{x^{187}} = -\frac{187}{x^{188}}$
 $\frac{d}{dx} \frac{1}{x^{188}} = -\frac{188}{x^{189}}$
 $\frac{d}{dx} \frac{1}{x^{189}} = -\frac{189}{x^{190}}$
 $\frac{d}{dx} \frac{1}{x^{190}} = -\frac{190}{x^{191}}$
 $\frac{d}{dx} \frac{1}{x^{191}} = -\frac{191}{x^{192}}$
 $\frac{d}{dx} \frac{1}{x^{192}} = -\frac{192}{x^{193}}$
 $\frac{d}{dx} \frac{1}{x^{193}} = -\frac{193}{x^{194}}$
 $\frac{d}{dx} \frac{1}{x^{194}} = -\frac{194}{x^{195}}$
 $\frac{d}{dx} \frac{1}{x^{195}} = -\frac{195}{x^{196}}$
 $\frac{d}{dx} \frac{1}{x^{196}} = -\frac{196}{x^{197}}$
 $\frac{d}{dx} \frac{1}{x^{197}} = -\frac{197}{x^{198}}$
 $\frac{d}{dx} \frac{1}{x^{198}} = -\frac{198}{x^{199}}$
 $\frac{d}{dx} \frac{1}{x^{199}} = -\frac{199}{x^{200}}$
 $\frac{d}{dx} \frac{1}{x^{200}} = -\frac{200}{x^{201}}$
 $\frac{d}{dx} \frac{1}{x^{201}} = -\frac{201}{x^{202}}$
 $\frac{d}{dx} \frac{1}{x^{202}} = -\frac{202}{x^{203}}$
 $\frac{d}{dx} \frac{1}{x^{203}} = -\frac{203}{x^{204}}$
 $\frac{d}{dx} \frac{1}{x^{204}} = -\frac{204}{x^{205}}$
 $\frac{d}{dx} \frac{1}{x^{205}} = -\frac{205}{x^{206}}$
 $\frac{d}{dx} \frac{1}{x^{206}} = -\frac{206}{x^{207}}$
 $\frac{d}{dx} \frac{1}{x^{207}} = -\frac{207}{x^{208}}$
 $\frac{d}{dx} \frac{1}{x^{208}} = -\frac{208}{x^{209}}$
 $\frac{d}{dx} \frac{1}{x^{209}} = -\frac{209}{x^{210}}$
 $\frac{d}{dx} \frac{1}{x^{210}} = -\frac{210}{x^{211}}$
 $\frac{d}{dx} \frac{1}{x^{211}} = -\frac{211}{x^{212}}$
 $\frac{d}{dx} \frac{1}{x^{212}} = -\frac{212}{x^{213}}$
 $\frac{d}{dx} \frac{1}{x^{213}} = -\frac{213}{x^{214}}$
 $\frac{d}{dx} \frac{1}{x^{214}} = -\frac{214}{x^{215}}$
 $\frac{d}{dx} \frac{1}{x^{215}} = -\frac{215}{x^{216}}$
 $\frac{d}{dx} \frac{1}{x^{216}} = -\frac{216}{x^{217}}$
 $\frac{d}{dx} \frac{1}{x^{217}} = -\frac{217}{x^{218}}$
 $\frac{d}{dx} \frac{1}{x^{218}} = -\frac{218}{x^{219}}$
 $\frac{d}{dx} \frac{1}{x^{219}} = -\frac{219}{x^{220}}$
 $\frac{d}{dx} \frac{1}{x^{220}} = -\frac{220}{x^{221}}$
 $\frac{d}{dx} \frac{1}{x^{221}} = -\frac{221}{x^{222}}$
 $\frac{d}{dx} \frac{1}{x^{222}} = -\frac{222}{x^{223}}$
 $\frac{d}{dx} \frac{1}{x^{223}} = -\frac{223}{x^{224}}$
 $\frac{d}{dx} \frac{1}{x^{224}} = -\frac{224}{x^{225}}$
 $\frac{d}{dx} \frac{1}{x^{225}} = -\frac{225}{x^{226}}$
 $\frac{d}{dx} \frac{1}{x^{226}} = -\frac{226}{x^{227}}$
 $\frac{d}{dx} \frac{1}{x^{227}} = -\frac{227}{x^{228}}$
 $\frac{d}{dx} \frac{1}{x^{228}} = -\frac{228}{x^{229}}$
 $\frac{d}{dx} \frac{1}{x^{229}} = -\frac{229}{x^{230}}$
 $\frac{d}{dx} \frac{1}{x^{230}} = -\frac{230}{x^{231}}$
 $\frac{d}{dx} \frac{1}{x^{231}} = -\frac{231}{x^{232}}$
 $\frac{d}{dx} \frac{1}{x^{232}} = -\frac{232}{x^{233}}$
 $\frac{d}{dx} \frac{1}{x^{233}} = -\frac{233}{x^{234}}$
 $\frac{d}{dx} \frac{1}{x^{234}} = -\frac{234}{x^{235}}$
 $\frac{d}{dx} \frac{1}{x^{235}} = -\frac{235}{x^{236}}$
 $\frac{d}{dx} \frac{1}{x^{236}} = -\frac{236}{x^{237}}$
 $\frac{d}{dx} \frac{1}{x^{237}} = -\frac{237}{x^{238}}$
 $\frac{d}{dx} \frac{1}{x^{238}} = -\frac{238}{x^{239}}$
 $\frac{d}{dx} \frac{1}{x^{239}} = -\frac{239}{x^{240}}$
 $\frac{d}{dx} \frac{1}{x^{240}} = -\frac{240}{x^{241}}$
 $\frac{d}{dx} \frac{1}{x^{241}} = -\frac{241}{x^{242}}$
 $\frac{d}{dx} \frac{1}{x^{242}} = -\frac{242}{x^{243}}$
 $\frac{d}{dx} \frac{1}{x^{243}} = -\frac{243}{x^{244}}$
 $\frac{d}{dx} \frac{1}{x^{244}} = -\frac{244}{x^{245}}$
 $\frac{d}{dx} \frac{1}{x^{245}} = -\frac{245}{x^{246}}$
 $\frac{d}{dx} \frac{1}{x^{246}} = -\frac{246}{x^{247}}$
 $\frac{d}{dx} \frac{1}{x^{247}} = -\frac{247}{x^{248}}$
 $\frac{d}{dx} \frac{1}{x^{248}} = -\frac{248}{x^{249}}$
 $\frac{d}{dx} \frac{1}{x^{249}} = -\frac{249}{x^{250}}$
 $\frac{d}{dx} \frac{1}{x^{250}} = -\frac{250}{x^{251}}$
 $\frac{d}{dx} \frac{1}{x^{251}} = -\frac{251}{x^{252}}$
 $\frac{d}{dx} \frac{1}{x^{252}} = -\frac{252}{x^{253}}$
 $\frac{d}{dx} \frac{1}{x^{253}} = -\frac{253}{x^{254}}$
 $\frac{d}{dx} \frac{1}{x^{254}} = -\frac{254}{x^{255}}$
 $\frac{d}{dx} \frac{1}{x^{255}} = -\frac{255}{x^{256}}$
 $\frac{d}{dx} \frac{1}{x^{256}} = -\frac{256}{x^{257}}$
 $\frac{d}{dx} \frac{1}{x^{257}} = -\frac{257}{x^{258}}$
 $\frac{d}{dx} \frac{1}{x^{258}} = -\frac{258}{x^{259}}$
 $\frac{d}{dx} \frac{1}{x^{259}} = -\frac{259}{x^{260}}$
 $\frac{d}{dx} \frac{1}{x^{260}} = -\frac{260}{x^{261}}$
 $\frac{d}{dx} \frac{1}{x^{261}} = -\frac{261}{x^{262}}$
 $\frac{d}{dx} \frac{1}{x^{262}} = -\frac{262}{x^{263}}$
 $\frac{d}{dx} \frac{1}{x^{263}} = -\frac{263}{x^{264}}$
 $\frac{d}{dx} \frac{1}{x^{264}} = -\frac{264}{x^{265}}$
 $\frac{d}{dx} \frac{1}{x^{265}} = -\frac{265}{x^{266}}$
 $\frac{d}{dx} \frac{1}{x^{266}} = -\frac{266}{x^{267}}$
 $\frac{d}{dx} \frac{1}{x^{267}} = -\frac{267}{x^{268}}$
 $\frac{d}{dx} \frac{1}{x^{268}} = -\frac{268}{x^{269}}$
 $\frac{d}{dx} \frac{1}{x^{269}} = -\frac{269}{x^{270}}$
 $\frac{d}{dx} \frac{1}{x^{270}} = -\frac{270}{x^{271}}$
 $\frac{d}{dx} \frac{1}{x^{271}} = -\frac{271}{x^{272}}$
 $\frac{d}{dx} \frac{1}{x^{272}} = -\frac{272}{x^{273}}$
 $\frac{d}{dx} \frac{1}{x^{273}} = -\frac{273}{x^{274}}$
 $\frac{d}{dx} \frac{1}{x^{274}} = -\frac{274}{x^{275}}$
 $\frac{d}{dx} \frac{1}{x^{275}} = -\frac{275}{x^{276}}$
 $\frac{d}{dx} \frac{1}{x^{276}} = -\frac{276}{x^{277}}$
 $\frac{d}{dx} \frac{1}{x^{277}} = -\frac{277}{x^{278}}$
 $\frac{d}{dx} \frac{1}{x^{278}} = -\frac{278}{x^{279}}$
 $\frac{d}{dx} \frac{1}{x^{279}} = -\frac{279}{x^{280}}$
 $\frac{d}{dx} \frac{1}{x^{280}} = -\frac{280}{x^{281}}$
 $\frac{d}{dx} \frac{1}{x^{281}} = -\frac{281}{x^{282}}$
 $\frac{d}{dx} \frac{1}{x^{282}} = -\frac{282}{x^{283}}$
 $\frac{d}{dx} \frac{1}{x^{283}} = -\frac{283}{x^{284}}$
 $\frac{d}{dx} \frac{1}{x^{284}} = -\frac{284}{x^{285}}$
 $\frac{d}{dx} \frac{1}{x^{285}} = -\frac{285}{x^{286}}$
 $\frac{d}{dx} \frac{1}{x^{286}} = -\frac{286}{x^{287}}$
 $\frac{d}{dx} \frac{1}{x^{287}} = -\frac{287}{x^{288}}$
 $\frac{d}{dx} \frac{1}{x^{288}} = -\frac$

Ohjelmistotekniikan menetelmät, kurssikoe 17.12.2014

Vastaa tehtäviin 1, 2 ja 3 erillisille konsepteille. Kirjoita jokaiseen palauttamaasi konseptiin kurssin nimi, kokeen päivämäärä, nimi ja opiskelijanumero.

Kokeessa saa olla mukana yksi (1) käsinkirjoitettu A4-kokoinen paperi muistiinpanoja.

Vastaukset palautetaan tehtäväkohtaisiin pinoihin. **Vaikka jättäisit johonkin tehtävään vastaamatta, tulee vastauspaperi siinäkin tapauksessa palauttaa.**

1. (4p) Kirjoita noin 2 sivun mittainen essee aiheesta "Ohjelmistojen elinkaari ja ohjelmistotuotantoprosessin vaiheet vesiputousmallissa ja ketterissä menetelmissä".

Kirjoita kokonaisia hyvin muotoiltuja lauseita, pelkillä ranskalaisilla viivoilla ei pisteitä ole luvassa.

2. (9p) Luokka- ja sekvenssikaavioita.

Joulupukin pajassa on juuri nyt työtahti kovimmillaan. Varoke Oy on tilannut Heebo Motel -palveluunsa pajasta virtuaalisen version joulun kunniaksi.

Lelupajassa on tonttuja, joilla on eri taitoja. Jotkut ovat puuseppiä, toiset metalliseppiä, kolmannet elektroniikka-asentajia. Osa tontuista on niin vanhoja ja viisaita että heillä on useampi taito. Pajan Mestaritonttu osaa tehdä kaikkea. Tontut tekevät leluja toivelistojen mukaisesti. Kun lelu tulee valmiiksi, se merkitään toivelistalla tehdyksi. Tonttu myös merkitsee siihen puumerkkinsä tulostarkkailua ja tammikuun yt-neuvotteluja varten. Kun listan kaikki lelut on merkattu valmiiksi, sihteeritonttu lähettää listasta raportin logistiikkakeskukseen käsiteltäväksi.

- (a) (3p) Mallinna yllä kuvattu järjestelmä luokkakaaviona. Merkitse osallistumisrajoitteet ja suunnat mahdollisimman tarkasti. Itsestäänselviä oliomuuttujia kuten "nimi" ei tarvitse merkitä. Metodeja ei tarvitse merkitä.
- (b) (3p) Tehtäväpaperin lopusta löytyy katkelma Java-koodia. Takaisinmallinna koodi sekvenssikaaviona. Main-luokan main-metodiin on merkitty kohta, josta alkaen sekvenssikaavio piirretään.
- (c) (3p) Mallinna tehtäväpaperin lopusta löytyvän javakoodin pohjalta oliokaavio tilanteesta jossa ohjelma on kun main-metodin viimeinen rivi on suoritettu.

3. (11p)

Yliopiston Flamma-järjestelmän inspiroimana tietojenkäsittelytieteen laitos on päättänyt hankkia uuden uutisten hallinnointiin tarkoitetun sisällönhallintajärjestelmän nimeltään "Pisara". Julkishallinnon luonteesta johtuen kehitystyö on ulkoistettu laitoksen ohjelmistotuotantoprojekteilte, onhan laitoksen motto "Teen itse, säästän ja opin".

Järjestelmä sallii tietyn käyttäjäjoukon (käytännössä ylempi hallinto sekä tiedotus-osasto) luoda, muokata ja poistaa uutisia. Uutiset koostetaan elementeistä, joilla on selkeä hierarkia.

Jokainen uutinen koostuu joukosta lukuja. Luvut puolestaan sisältävät kirjoittajan määräämässä järjestyksessä otsakkeita ja kappaleita. Sekä otsakkeet että kappaleet koostuvat sisältöelementeistä. Sisältöelementti on tässä yhteydessä yleistermi, joka kattaa yksittäisen sanan, linkin sekä kuvan. Järjestelmän linkit voivat viitata joko johonkin verkko-osoitteeseen tai suoraan toiseen järjestelmän tuntemaan uutiseen sen uniikin tunnistenumeron avulla. Sisäisen linkin etu on siis se, että vaikka uutisen verkko-osoite muuttuisikin, pysyy sen uniikki tunniste samana ja siten sisäinen linkki toimii aina, olettaen että viitteen kohteena olevaa uutista ei poisteta. Lisäksi sekä uutisella että jokaisella sen luvulla on kullakin aina yksi otsake, joka toimittaa kyseisen elementin pääotsakkeen roolia.

Uutisessa näkyy uutisen alkuperäisen luoja ja viimeisimpänä uutista muokanneen henkilökunnan jäsenen nimi.

Järjestelmä myös sallii kirjautuneiden käyttäjien lisätä uutisille lipukkeita ("tägejä"). Kaikki käyttäjät (myös vierailijat) voivat hakea uutisia lipukkeiden perusteella.

- (a) (3 + 1p) Etsi järjestelmästä käyttäjät ja ainakin viisi käyttötapausta. Nimeä käyttötapaukset kuvaavasti ja selkeästi, ja piirrä käyttötapauksista käyttötapausta-kaavio. Kuvaa lisäksi yksi käyttötapauksista tarkemmalla tasolla, ns. Cockburnin käyttötapauspohjan tai luentomonisteen tekstuaalisten kuvausten tyyliin.
- (b) (4p) Laadi järjestelmän kuvauksen perusteella määrittelyvaiheen (eli kohdealueen) luokkakaavio. Merkitse yhteyksiin kytkentärajoitteet. Nimeä yhteydet ja yhteysroolit tarvittaessa. Attribuutteja ei tarvitse merkitä, mutta nimeä luokkasi niin, että niiden rooli käy lukijalle selväksi. Muista, että toiminnallisuutta ei kannata määrittelyvaiheen luokkamalliin laittaa!
- (c) (3p) Laadi oliokaavio, jossa kuvaat järjestelmän seuraavassa tilassa: Järjestelmällä on kaksi käyttäjää ("Jukka" sekä "Jyrki") sekä yksi julkistettu uutinen. Uutisen pääotsikko on "Moi kaikki!". Uutisessa on yksi luku, jonka pääotsikko on "Testi-uutinen". Luvussa on yksi kappale jonka sisältö on sana "heips", jota seuraa kuva tiedostosta "kissat-on-kivoja.png". Uutisen alkuperäinen luoja on Jukka, mutta myös Jyrki on muokannut uutista myöhemmin.

Tehtävään 2 (b) liittyvä ohjelmakoodi:

```
public class Joulupukki {
    // ...

    public void annaLahja(Lapsi lapsi) {
        int kiltteys = lapsi.getKiltteys();

        if (kiltteys >= 10) {
            Lahja lahja = new Hammaslaakarikaynti();
            lapsi.otaLahja(lahja);

        } else {
            Lahja lahja = new Risupakkaus();
            lapsi.otaLahja(lahja);
        }
    }

    // ...
}
```

```
import java.util.ArrayList;
import java.util.List;

public class Lapsi {

    private String nimi;
    private int kiltteys;
    private List<Lahja> lahjat;

    public Lapsi(String nimi) {
        this.nimi = nimi;
        this.kiltteys = 10;
        this.lahjat = new ArrayList<>();
    }

    public void otaLahja(Lahja lahja) {
        this.lahjat.add(lahja);
    }

    public void kiroile() {
        System.out.println(this.nimi + ":-##!@@$$");
        this.kiltteys--;
    }

    public int getKiltteys() {
        return kiltteys;
    }
}
```

```
public interface Lahja {  
  
    void avaa();  
}
```

```
public class Risupakkaus implements Lahja {  
  
    @Override  
    public void avaa() {  
        // ...  
    }  
}
```

```
public class Hammaslaakarikaynti implements Lahja {  
  
    @Override  
    public void avaa() {  
        // ...  
    }  
}
```

```
public class Main {  
  
    public static void main(String[] args) {  
  
        Joulupukki arto = new Joulupukki();  
  
        Lapsi doge = new Lapsi("dogeman");  
        Lapsi loezi = new Lapsi("loeziman");  
  
        // mallinna sekvenssikaavio tästä tilasta lähtien  
        arto.annaLahja(doge);  
        doge.kiroile();  
        arto.annaLahja(doge);  
        doge.kiroile();  
        arto.annaLahja(loezi);  
        // mallinna oliokaaviona edeltävän rivin suorituksen jälkeinen tila  
    }  
}
```

Tietokantojen perusteet, kurssikoe 8. 3. 2017 / Arto Hellas

Vastaa tehtäviin 1, 2 ja 3 erillisille papereille. Kirjoita jokaiseen palauttamaasi paperiin nimesi ja henkilötunnuksesi sekä kurssin nimi. Jos teit tenttiä varten lunttilapun (käsin kirjoitettu kaksipuolinen a4-paperi), palauta se tentin yhteydessä.

Tehtävä 1. (10p) Kerro mistä kussakin alla olevassa käsitteessä tai käsiteparissa on kyse. Anna jokaisesta myös vähintään yksi esimerkki.

- Funktionaalinen riippuvuus
- Pää- ja viiteavain
- SQL-injektio
- Toinen normaalimuoto
- Verkkotietokanta

Tehtävä 2. (7+3p) Lasten ja nuorten kehityksestä kiinnostunut Hossein B. kertoo seuraavaa:

Haluaisin käyttöni mobiilisovelluksen, joka tarjoaa mahdollisuuden seurata ja pitää kirjaa lasten kasvusta. Sovelluksen tulisi mahdollistaa lapsen kasvun (pituus, paino) seuraaminen. Sovelluksen tulisi kysyä esimerkiksi päivittäin, viikoittain tai kuukausittain käyttäjältä päivityksiä lapsen kasvuun liittyen. Kirjausaikavälit riippuvat lapsen iästä.

Saman sovelluksen voisi antaa muillekin käyttöön. Kun sovellus tallentaa keräämänsä tiedot keskitettyyn tietokantaan, saadaan käyttöön suuria määriä kasvusta kertovaa informaatiota. Tätä voidaan käyttää esimerkiksi lasten ja nuorten pituuskasvuun liittyvien ennustemallien päivittämiseen. Luonnollisesti, jotta tämä olisi mahdollista, sovelluksen tulee mahdollistaa myös muiden tietojen kuten syntymäajan ja sukupuolen kirjaaminen, sekä vanhempiin liittyvien taustatietojen kuten vanhempien pituuksien kirjaaminen.

Jotta mobiilisovellukselle saataisiin paljon käyttäjiä, tulee siinä olla myös jonkinlainen tapa innostaa käyttäjiä sen käyttöön. Eräs trendi, jota tässä voisi käyttää on pelillistäminen, eli pelillisten elementtien kuten saavutusten lisääminen osaksi sovellusta. Sovelluksessa voisi olla kirjauksiin ja kasvuun liittyviä saavutuksia kuten "päivän kirjaukset tehty", "viikon kirjaukset tehty", sekä "lapsi kasvanut sentin", "lapsi kasvanut 5 senttiä" jne. Sovellukseen voisi lisätä myös isäi ajatellen muita saavutuksia, kuten "lapsi pissasi neuvolatädin päälle".

Sovelluksen tulisi mahdollistaa vähintäänkin seuraavien tietojen näyttäminen:

- Miten (oman) lapsen paino ja pituus on kasvanut annettuna aikavälinä?
- Mitkä ovat kaikkien lapsien keskipaino ja keskipituus esimerkiksi yhden vuoden iässä?
- Kuinka usein tietty henkilö on tehnyt kirjauksia?
- Kuinka monta prosenttia sovelluksen käyttäjistä on saanut "lapsi pissasi neuvolatädin päälle"-saavutuksen?
- Minkä pituinen poikalapsi on keskimäärin 3 vuoden iässä jos äidin pituus on 167 cm?

Tee ylläolevasta kuvauksesta tietosisältöanalyysi, eli etsi järjestelmään liittyvät käsitteet. Luo käsitteiden pohjalta luokkakaavio (käsitekaavio). Yhteyksien ja osallistumisrajoitteiden

Tietokantojen perusteet, kurssikoe 8. 3. 2017 / Arto Hellas

Vastaa tehtäviin 1, 2 ja 3 erillisille papereille. Kirjoita jokaiseen palauttamaasi paperiin nimesi ja henkilötunnuksesi sekä kurssin nimi. Jos teit tenttiä varten lunttilapun (käsin kirjoitettu kaksipuolinen a4-paperi), palauta se tentin yhteydessä.

merkintä on oleellista, attribuuteista tulee merkitä vain oleellisimmat. Vastauspaperiin riittää vain lopullinen kaavio, eli kaikkia käsiteanalyysin vaiheita ei tarvitse tässä kirjata. Huom! Tämän kaavion ei tule olla tietokantakaavio.

Kun olet piirtänyt käsitekaavion, piirrä tämän jälkeen tekemästäsi käsitekaaviosta erillinen tietokantakaavio. Voit käyttää joko kurssilla käytettyä merkintätapaa tai tehtävän 4 merkintätapaa. Merkitse tietokantakaavioon kaikki attribuutit, pääavaimet sekä viiteavaimet.

Tehtävä 3. (10p) Tietokantojen suunnitteluun ja käsittelyyn erikoistuneen yrityksen "Schöne Feste und Spiele mit ganz gut Statistik" tietokantaekspertti Prof. Dr. auf der Relation on luonut elektronisten pelien tilastoja varten seuraavanlaisen tietokantakaavion.

Joukkue((pk) id, nimi)
Pelaaja((pk) id, nimi, (fk) joukkue_id -> Joukkue)
Tietokonepeli((pk) id, nimi)
OtteluTulos((fk) pele_id -> Tietokonepeli, (fk) kotijoukkue_id -> Joukkue, (fk) vierasjoukkue_id -> Joukkue, kotijoukkue_pisteet, vierasjoukkue_pisteet, paivamaara)

Taulu "Tietokonepeli" kuvaa tietokonepeliä, esimerkiksi Dota tai Sims. Taulu "OtteluTulos" kuvaa kahden joukkueen välisen yksittäisen ottelun tulosta.

Kirjoita seuraavat SQL-kyselyt:

- Kysely, joka listaa kaikki joukkueen "The Filthy Underwears" pelaajat aakkosjärjestyksessä.
- Kysely, joka kertoo joukkueen "Ninjas in Pyjamas" pelaajien lukumäärän.
- Kysely, joka kertoo kuinka monta tasapeliä joukkue "Mirage" on pelannut (tasapeleissä sekä kotijoukkueella että vierasjoukkueella on sama pistemäärä).
- Kysely, joka kertoo joukkuekohtaisen kotona pelattujen otteluiden lukumäärän, eli kuinka monta ottelua kukin joukkue on pelannut kotijoukkueena. Huomioi tässä myös ne joukkueet, jotka eivät ole olleet osallisena yhdessäkään ottelussa.
- Kysely, joka listaa niiden joukkueiden nimet, jotka ovat olleet osallisena yhteensä vähintään 10 ottelussa.

Kokeessa saa pitää mukana käsinkirjoitettua A4-kokoista kaksipuolista "lunttilappua", joka on palautettava koepaperin mukana. Huomaa että jokaisen koetilaisuuteen osallistuvan on palautettava vähintään tyhjä koepaperi, johon on kirjattu opiskelijanumero ja nimi.

Merkitse jokaiseen vastauspaperiin kurssin nimi, päivämäärä, opiskelijanumerosi, nimesi ja allekirjoitus. Kirjoita vastaukset s e l k e ä l l ä käsialalla.

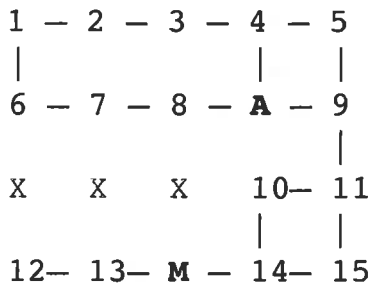
HUOM: Tarkista että olet saanut kaikki tehtäväarkit. Vastaa neljään (4) tehtävään viidestä (5). Jos vastaat kaikkiin tehtäviin, arvostelussa otetaan huomioon ensimmäiset neljä. Jokaisen tehtävän maksimipistemäärä on 10.

1. Tekoölyn filosofiaa ja historiaa

- a. (6 p) Esseekysymys. Sopivan pituinen vastaus on noin yhden kokonaisen sivun mittainen.
Aihe: "Deep learning ratkaisee kaiken?"
Deep learning eli syvät neuroverkot ovat viimeisten viiden vuoden aikana saaneet valtavasti huomiota, koska niiden avulla on saavutettu suuria edistysaskeleita. Uskotaan, että deep learning on ratkaisu lähes kaikkiin tekoölyn ongelmiin, kunhan jäljellä olevat esteet lähes kaiken ratkaisevassa tekoölyssä voidaan ylittää. Tämän väitetään olevan mahdollista jo lähivuosina. Pohdi näitä odotuksia ottaen huomioon tekoölyn historiassa ilmenneet suuret kehityskaaret.
- b. (2 p) Pitääkö väittää "tekoölytutkimuksen ensisijainen tarkoitus on kopioida ihmis-älykkyys" paikkansa vai ei? Perustele vastauksesi lyhyesti.
- c. (2 p) Elokuissa tekoöly usein saavuttaa tietoisuuden ja hyökkää ihmisten kimppuun. Kuvitellaan, että Suomen hallitus on huolissaan tästä riskistä ja ajaa lakia, jolla kaikki tekoölyalgoritmit kielletään. Millaisia käytännön seurauksia tällaisella lailla olisi? Mainitse muutama esimerkki. Mitä ongelmia tällaisen lain valvomiseen liittyisi?

2. A*-haku

Simuloi A*-hakua seuraavassa verkossa:



$\sqrt{1} = 1$	$\sqrt{11} \approx 3.32$
$\sqrt{2} \approx 1.41$	$\sqrt{12} \approx 3.46$
$\sqrt{3} \approx 1.73$	$\sqrt{13} \approx 3.61$
$\sqrt{4} = 2$	$\sqrt{14} \approx 3.74$
$\sqrt{5} \approx 2.24$	$\sqrt{15} \approx 3.87$
$\sqrt{6} \approx 2.45$	$\sqrt{16} = 4$
$\sqrt{7} \approx 2.65$	$\sqrt{17} \approx 4.12$
$\sqrt{8} \approx 2.83$	$\sqrt{18} \approx 4.24$
$\sqrt{9} = 3$	$\sqrt{19} \approx 4.36$
$\sqrt{10} \approx 3.16$	$\sqrt{20} \approx 4.47$

missä numeroidut solmut 1–15 ovat tavallisia solmuja, ‘X’ on saavuttamattomissa oleva solmu, ‘M’ on maalisolmu, ‘A’ on aloitussolmu, ja viivat solmujen välillä ovat sallittuja siirtymiä. (Verkossa ‘|’ tai ‘-’ tarkoittaa yhtä siirtymää). Jokaisen sallitun siirtymän kustannus on 1 yksikkö.

Käytä heuristiikkana euklidista (eli “lennuntie-”) etäisyyttä $\sqrt{(\Delta_x^2 + \Delta_y^2)}$ missä $\sqrt{\quad}$ tarkoittaa neliöjuurta, Δ_x on etäisyys leveysuunnassa ja Δ_y on etäisyys pystysuunnassa. Esim. alkusolmulle heuristiikka antaa siis arvion $\sqrt{(2^2 + 1^2)} = \sqrt{5} \approx 2.24$ (ks. oheinen taulukko).

- a. (4 p) Listaa kussakin etsinnän vaiheessa sen hetkinen käsiteltävä solmu ja solmulistan sisältö siinä järjestyksessä, jossa siitä poimitaan solmuja.

esim.

M: [14, 13]

- b. (4 p) Merkitse a-kohdan ratkaisussa kunkin solmun kohdalle arvot, joiden mukaan solmulista on järjestetty.

esim.

M: [14 (3), 13 (3.83)]

- c. (2 p) Minkä reitin A*-haku tuottaa tulokseksi? Millä tavoin A*-haku on parempi kuin esimerkiksi leveysuuntainen haku? Miten tämä etu tulee ilmi tämän tehtävän reittihaussa?

3. Luonnollisen kielen käsittely (NLP)

a. (2 p) Olkoon kontekstittoman kieliopin säännöt seuraavat:

$$\begin{array}{lll} S \rightarrow P S & P \rightarrow \text{tosi} & N \rightarrow \text{ratkaisu} \\ S \rightarrow A N & A \rightarrow \text{hieno} & \end{array}$$

Saadaanko näitä sääntöjä soveltamalla aloitussymbolista S muodostettua lauseet ”ratkaisu”, ”tosi tosi tosi hieno ratkaisu” ja ”tosi tosi hieno hieno ratkaisu”? Perustelee.

b. (4 p) Sovella CYK-algoritmia täyttääksesi oheinen taulukko. *Vinkki:* Solut (3,3), (4,4) ja (3,4) on täytetty valmiiksi. Muista, että solu (3,4) viittaa lauseen sanoihin 3–4 eli ”hieno ratkaisu”.

(1,4)				
(1,3)	(2,3)			
(1,2)	(2,3)	S	(3,4)	
(1,1)	(2,2)	A	(3,3)	N (4,4)
tosi	tosi	hieno	ratkaisu	

c. (2 p) Esitä edellisen kohdan jäsenystaulukosta saatava jäsenyspuu (tai puut).

d. (2 p) Esitä esimerkki lauseesta, joka on monitulkinainen. Millä perusteella voidaan pyrkiä päättämään ”oikea” tulkinta automaattisesti?

4. Todennäköisyysmallinnus

a. (3 p) Mikä on Bayesin kaava? Anna käytännön esimerkki sen soveltamisesta käyttäen sopimaksi katsomiasi, konkreettisia tapahtumia (ei siis pelkästään A , B , ...). Käytä valitsemiasi mielekkäitä numeroarvoja, joista mikään ei saa olla tasan 0 tai 1.

b. (2 p) Miksi Bayesin kaava on tärkeä tekoälyssä?

c. (3 p) Miten Bayes-verkosta generoidaan dataa?

d. (2 p) Miten arvioidaan ehdollista todennäköisyyttä $P(\text{”On kevät”} \mid \text{”Istun luentosalissa”, ”Ulkona on harmaata”})$, kun on generoitu 100 000 monikkoa edellä mainittujen tapahtumien välisiä suhteita kuvaavasta Bayes-verkosta?

5. Neuroverkot

- a. (4 p) Leiki opettajaa. Etsi neuroverkkoaiheinen kysymys, kun vastaus on annettu!
- i. $f(z) = 0$, kun $z < 1$, $f(z) = 1$, muuten.
 - ii. Boltzmannin kone
 - iii. paljon suurempi skaala, epäsynkronisuus (eri neuronit suorittavat laskentaa eri tahdissa) ja tyypillisesti binääriset viestit
 - iv. visualisointi
- b. (3 p) Kuvaile valitsemasi neuroverkon toimintaperiaate. Mitkä ovat verkon syötteet ja mitä tapahtuu, kun syöte esitetään verkolle?
- c. (2 p) Selitä kyseisen verkon oppimismenetelmä ja kerro samalla, minkä muotoista opetusaineistoa verkon opettamiseen käytetään. Riittää selittää toimintaperiaate, ei tarvitse esittää yksityiskohtia. (Ei pseudokoodia, vaan on selitettävä sanallisesti.)
- d. (1 p) Mikä on tyypillinen kyseisen neuroverkon sovellus?

FLUM

Ohjelmistotuotanto, kurssikoe 9.5.2015

Kirjoita jokaiseen palauttamaasi konseptiin kurssin nimi, kokeen päivämäärä, nimesi ja opiskelijanumerosi.

Kaikkien tehtävien vastaukset tulee tehdä omille konsepteille.

Vastaa tehtäviin ytimekkäästi. Sopiva vastauksen pituus kuhunkin tehtävästä on noin 1-2 sivua. **Täydet pisteet voi saada ainoastaan kirjoittamalla järkevästi muotoillun esseen, ranskalaiset viivat eivät riitä.**

1. (4p) Kerro mitä tarkoitetaan product backlogilla ja sprint backlogilla. Minkälainen on hyvä product backlog? Entä hyvä sprint backlog? Miten scrum-tiimin eri rooleissa olevat henkilöt hyödyntävät backlogeja?

2. (2+1.5+1.5p)

(a) Ketterien menetelmien yhteydessä puhutaan usein käsitteistä definition of done ja hyväksymiskriteerit (engl. acceptance criteria). Selitä mitä käsitteillä tarkoitetaan. Anna konkreettinen esimerkki molemmista

(b) Mitä tarkoitetaan regressiotestaamisella? Miten regressiotestaus yleensä hoidetaan kun ohjelmistoja kehitetään ketterien periaatteiden mukaan?

Mitä tarkoitetaan tutkivalla testaamisella? Minkälaisiin tilanteisiin tutkiva testaus sopii ja mihin se ei sovi?

3. (5p) Alla on lueteltu kuusi kurssilla vastaan tullutta termiä. Selitä lyhyesti **viiden** termin merkitys, missä niitä käytetään ja mikä niiden hyödyt ovat?

- minimal viable product
- work in progress limit
- velositeetti
- staging-palvelin tai staging-ympäristö
 - stub- ja mock-oliot
 - mikropalvelmalli/arkkitehtuuri (microservice)

HUOM: jätä yhden termin merkitys selittämättä. Jos kerrot jokaisesta termistä, arvostellaan viisi huonointa vastausta.

4. (1+2+3p)

(a) Mitä tarkoitetaan ohjelman ulkoisella ja sisäisellä laadulla?

(b) Mitä sisäisen laadun kannalta ongelmallisia asioita tehtäväpaperin lopun koodissa on?

(c) Selitä miten refaktorisit tehtäväpaperin lopun koodia soveltaen suunnittelumalleja tai muita tilanteeseen sopivia ratkaisuja. Vastaukseen tulee liittää mukaan luonnosmaista koodia, pelkkä suunnittelumallien nimien luetteleminen ei riitä. Älä kuitenkaan kirjoita "javaboilerplatea" (luokka+näkyvyysmäärittelyt ym.) kokonaisuudessaan

```

public class Main {
    public static void main(String[] args) {
        Company ibm = new Company("database");
        // adding employees etc...

        // print hour workers in csv-format
        for (Employee hourWorker : ibm.hourWorkes()) {
            System.out.println(hourWorker.formatAs("csv"));
        }

        // print employees with age less than 50 in json-format
        for (Employee youngEmployee : ibm.ageLessThan(50)) {
            System.out.println(youngEmployee.formatAs("json"));
        }
    }
}

```

```

public class Company {
    private List<Employee> employees;
    private String source;

    public Company(String source) {
        this.source = source;
        if ( source.equals("file")) {
            readEmployeesFromFile();
        } else if ( source.equals("database")) {
            readEmployeesFromDatabase();
        }
    }

    public List<Employee> hourWorkes(){
        List<Employee> result = new ArrayList<>();

        for (Employee employee : employees) {
            if ( employee.isHourWorker() ) {
                result.add(employee);
            }
        }

        return result;
    }

    public List<Employee> withMonthSalaryAtLeast(int amount){
        List<Employee> result = new ArrayList<>();

        for (Employee employee : employees) {
            if ( !employee.isHourWorker() && employee.getSalary()>amount ) {
                result.add(employee);
            }
        }

        return result;
    }

    public List<Employee> ageLessThan(int limit){
        List<Employee> result = new ArrayList<>();

```

konvertointaja m.

iterointia ← interface

read/save ← interface

Format As

↑
payment identiset

enum data source

I Company data source

I Finder

I employee formatter

enum format

Static factory ? no.

Polym Iterointia

```

    for (Employee employee : employees) {
        if ( employee.getAge() < limit ) {
            result.add(employee);
        }
    }

    return result;
}

// also many more simillar finder methods...

public void setNewSalary(String name, int salary){
    for (Employee employee : employees) {
        if ( employee.getName().equals(name)) {
            employee.setSalary(salary);
            return;
        }
    }
    throw new IllegalStateException(name+" was not found");
}

public void newEmployee(Employee e){
    for (Employee employee : employees) {
        if ( employee.getName().equals(e.getName())) {
            throw new IllegalStateException(e.getName()+" was already added");
        }
    }

    employees.add(e);
    if ( source.equals("file")) {
        saveEmployeesToFile(e);
    } else if ( source.equals("database")) {
        saveEmployeesToDatabase(e);
    }
}

// details of the following not shown
private void readEmployeesFromFile() { ... }
private void readEmployeesFromDatabase() { ... }
private void saveEmployeeToFile(Employee e) { ... }
private void saveEmployeeToDatabase(Employee e) { ... }
}

public class Employee {
    private String name;
    private int age;
    private String sex;
    private int salary;
    private int hoursWorked;
    private boolean hourWorker;

    public Employee(String name, int age, String sex, int salary, boolean monthly) {
        this.name = name;
        this.age = age;
        this.sex = sex;
        this.salary = salary;
        this.hourWorker = monthly;
    }
}

```

} interface

```

public int salaryToPay(){
    int amount = 0;

    if ( hourWorker ) {
        amount = hoursWorked*salary;
        hoursWorked = 0;
    } else {
        amount = salary;
    }

    return amount;
}

public void addHours(int hours ){
    if ( hourWorker ) {
        hoursWorked += hours;
    }
}

public String formatAs(String format){
    if ( format.equals("csv")) {
        String payment = "";
        if ( hourWorker ) {
            payment = salary+"/hour";
        } else {
            payment = salary+"/month";
        }

        return name+";"+age+";"+sex+";"+payment;
    } else if ( format.equals("json")) {
        String payment = "";
        if ( hourWorker ) {
            payment = salary+"/hour";
        } else {
            payment = salary+"/month";
        }
        return "{ name:"+name+", age:"+age+", sex: "+sex+", payment:"+payment+" }";
    } else if ( format.equals("xml")) {
        // details not shown
        return "";
    }
    // more formats will be added later

    throw new IllegalStateException("format "+format+" not supported");
}

// getters and setters
public int getAge() { return age; }
public String getName() { return name; }
public String getSex() { return sex; }
public boolean isHourWorker() { return hourWorker; }
public int getSalary() { return salary; }
public void setSalary(int salary) { this.salary = salary; }
}

```

University of Helsinki
Department of Computer Science

CSM14101 Software Architectures

Course Exam 25.10.2018

Examiner: Antti-Pekka Tuovinen

No other materials or implements except for writing are allowed in the examination.

Write on each separate sheet of paper the name of the course and the date of the exam, your name (clearly), your student number and your signature. Students who are taking their whole degree in English must answer in English, but others can answer also in Finnish or Swedish.

The exam yields 32 points in total. Each question is worth 8 points.

The expected length of an answer is about one hand written page (75 - 100 words). The answers must consist of complete sentences (not just bullet points).

1. Explain how software architecture can be used in a *prescriptive role* and in a *descriptive role* in software development by explaining shortly two uses from both role categories.
2. What does *Information Hiding* mean as a software design principle? Give an example (no code needed, explain on a general level).
3. Explain the *solution characteristics* of the Microservice Architecture pattern. What kind of elements there are, what are their responsibilities, how do they communicate, and how does this affect organizing the development work?
4. Explain the basic idea of *scenario-based* architecture assessment. What is good about scenarios?

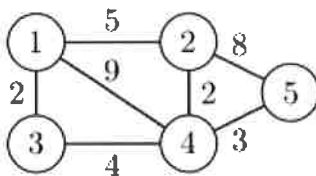
Data Structures and Algorithms, Fall 2018
Exam 18.12.2018 / Antti Laaksonen

Duration: 2 h 30 min

Write the answer to each problem **in a separate paper**. Write down your name and student number on each paper. If you use a cheat sheet, return it with your answers.

Problem 1 (4 points)

How does Prim's algorithm process the following graph? Show a suitable number of intermediate steps and the final result.



Problem 2 (4 points)

Write a short answer to each question:

1. Give an example of a graph that has exactly three nodes and is a tree.
2. Give an example of a graph that is connected but not strongly connected.
3. Give an example of a graph whose topological sort is [4, 2, 3, 1].
4. Give an example of a graph where each node has an even degree.
5. What is the benefit of using breadth-first search, compared to depth-first search?
6. What is the benefit of using depth-first search, compared to breadth-first search?
7. How can binary search be used in algorithm design?
8. How can dynamic programming be used in the knapsack problem?

You will get 0.5 points for each correct answer, and the total score for the problem will be the sum of these points, rounded down to an integer.

Problem 3 (6 points)

You are given a set of cities and bidirectional roads between them. There is a route between each pair of two cities using the roads. Your task is to find out if it is possible to remove a road so that after the removal, there is still a route between each pair of two cities.

For example, if the cities are A, B and C and the roads are A–B, B–C and A–C, the answer is "yes", because you can remove, for example, the road A–B.

However, if the cities are A, B and C and the roads are A–B and A–C, the answer is "no", because both the roads are necessary.

Describe how you can model the problem as a graph. Which property of the graph indicates if the answer is "yes" or "no"?

Problem 4 (6 points)

You are given an integer n and your task is to calculate the total number of ways to construct a string of length n whose each character is 0 or 1 and there are no three consecutive positions where each character is the same.

For example, if n is 4, the correct answer is 10, because the valid strings are 0010, 0011, 0100, 0101, 0110, 1001, 1010, 1011, 1100 and 1101.

Your task is to design a dynamic programming algorithm for the task. The algorithm must work in $O(n)$ time or more efficiently.

First explain the idea of the algorithm in 1–2 sentences. After this, give a code for the algorithm and explain why it works efficiently. You can use pseudocode or any programming language, and you can use the data structures discussed during the course.

Did you already give course feedback?

You can still give feedback today or tomorrow according to the instructions on the course site. You will get one extra exam point for giving feedback.

Ohjelmistoprojektien johtaminen ja ryhmädynamiikka: kurssikoe 3.3.2020

Kirjoita jokaisen vastauspaperisi alkuun kurssin nimi ja kokeen päivämäärä sekä nimesi, opiskelijanumerosi ja **myös allekirjoituksesi**.

1. Määrittele seuraavat ryhmädynamiikan käsitteet ja ilmiöt. Pohdi *lisäksi* lyhyesti niiden merkitystä ohjelmistotuotantoryhmän kannalta. (6x2 = 12 pistettä)
 - a. sosio-emotionaalinen funktio ja tehtäväfunktio
 - b. normien merkitys yksilölle ryhmän jäsenenä
 - c. normien merkitys ryhmälle
 - d. roolit ja roolikonfliktit
 - e. sosiaalinen vertailu: merkitys ja mekaniikka
 - f. Abilenen paradoksi
2. Laadi ryhmän dynamiikan laatua varmistava *tiivis luettelomuotoinen* ohjeistus ohjelmistotuotantoryhmän projektipäällikölle tai vastaavalle seuraavissa tilanteissa. Vastauksissa arvostetaan sosiaalipsykologisen käsitteistön oikeaa ja luontevaa käyttöä. (10 pistettä)
 - a. Ryhmä järjestäytyy. Ulkopuolinen taho on valinnut ryhmän jäsenet.
 - b. Ryhmä ajautuu vakaviin ristiriitohin työtehtäviin liittymättömien henkilösuhteiden takia, "kemat eivät toimi".
 - c. Ryhmä ajautuu vakaviin ristiriitohin teknisten ongelmien ja keskinäisen syyttelyn takia.
 - d. Projekti on päättymässä lähes onnistuneesti. Vielä on tehtävä viimenen ponnistus ehkä vapaa-aikaakin uhraten.

DATA11007 Statistics for Data Science

Exam 25 January 2023

Antti Honkela, Mikko Heikkilä and Marlon Tobaben

Answer all problems. Each problem is worth 6 points.

Allowed equipment: writing instruments.

Voit vastata kysymyksiin myös suomeksi.

1. True or false? **Answers to the attached answer sheet.** (Correct answer $\frac{1}{2}$ points, wrong answer $-\frac{1}{4}$ points, empty 0 points)

- i. The variance of the sum of any two random variables is the sum of their variances.
- ii. According to the central limit theorem, the sum $\sum_{i=1}^n X_i$ of independent random variables X_1, X_2, \dots , each following the uniform distribution $X_i \sim \text{Uniform}(-1, 1)$, converges in distribution to a Normal distribution.
- iii. The mean squared error (MSE) of an estimator $\hat{\theta}_n$ can be written as

$$\text{MSE} = \text{bias}(\hat{\theta}_n) + \mathbb{V}(\hat{\theta}_n).$$

- iv. The expected value of a consistent estimator is equal to the true parameter value.
- v. A maximum likelihood estimator is always unbiased.
- vi. Bootstrap sampling can be used to estimate the variance of plug-in estimators.
- vii. p -value is equal to the probability of observing under the null hypothesis a value of the test statistic that is at least as extreme as what was observed.
- viii. Statistically significant differences are always practically important.
- ix. According to the Bayesian method, predictions are computed as averages over the prior distribution.
- x. A minimax estimator always has the smallest Bayes risk for all prior distributions.
- xi. When treatments are randomised, any consistent estimator of association is a consistent estimator of the causal effect.
- xii. Maximising the number of obtained responses is the most important consideration in survey design.

2. Briefly explain the following terms

- i. Confidence interval
- ii. Multiple testing problem
- iii. Simpson's paradox
- iv. Missing not at random (MNAR)

Design and Analysis of Algorithms (Autumn 2023), 5 cr

Course exam 25.10.2012 – Model solutions

Examiner: Veli Mäkinen

Remember to write your name on each answer sheet. You can answer also in Finnish or in Swedish.

Course grading: $15+15+15+15=60$. The assignment yielding least points will be replaced by exercise points if that gives a higher score.

1. Recurrences

Consider recurrence $T(n) = 2T(n/2) + n \log_2 n$. The following *flawed* substitution proof aims to prove $T(n) = O(n \log^2 n)$. Spot the errors in the proof and then fix them to give a correct analysis. Before fixing the induction proof, check first using the recursion tree method if $O(n \log^2 n)$ is the correct bound.

Flawed analysis.

Assume $T(m) \leq cm \log_2^2 m$ for $m < n$.

Then by induction

$$\begin{aligned} T(n) &\leq 2cn/2 \log_2^2 \frac{n}{2} + n \log_2 n \\ &\leq cn \log_2^2 n + n \log_2 n \\ &= O(n \log^2 n). \end{aligned}$$

Solution.

First, the recursion tree yields $T(n) = n \sum_{i=0}^{\log_2 n} \log \frac{n}{2^i} < n \sum_{i=0}^{\log_2 n} \log n = O(n \log^2 n)$, so the bound is indeed correct, just the analysis is flawed.

The analysis estimates the equation too much upwards failing to show that $T(n) \leq cn \log_2^2 n$ as it should. Also base case is missing. More careful analysis is sufficient to fix the flaws.

Assume again $T(m) \leq cm \log_2^2 m$ for $m < n$.

Then by induction

$$\begin{aligned} T(n) &\leq 2cn/2 \log_2^2 \frac{n}{2} + n \log_2 n \\ &= cn(\log_2^2 n - 2 \log_2 n + 1) + n \log_2 n \\ &\leq cn \log_2^2 n, \end{aligned}$$

when $-2cn \log_2 n + cn + n \log_2 n \leq 0$. This holds for $c \geq 1$ and $n \geq 2$, as then $cn \leq cn \log_2 n$.

Assume the natural base case $T(1) = 0$. Induction assumes $T(1) \leq c1 \log_2^2 1 = 0$, which holds in the base case.

Grading.

- 5 points for checking that the bound is correct using the recursion tree method.
- 7 points for fixing the induction step correctly.
- 3 points for fixing the constants by checking the base case.

Common mistake.

A common mistake was to subtract a lower order term, e.g., $T(m) \leq cm \log_2^2 m - bm \log_2 m$ for $m < n$, but showing only $T(n) \leq cn \log^2 n$, while one should show $T(n) \leq cn \log^2 n - bn \log_2 n$ in this case.

2. Flows

A *cycle cover* of a directed graph $G = (V, E)$ is a set of vertex-disjoint simple cycles that cover all the vertices, i.e., every vertex appears exactly once in the cycle cover. In the Cycle Cover problem we are asked whether G has a cycle cover or not.

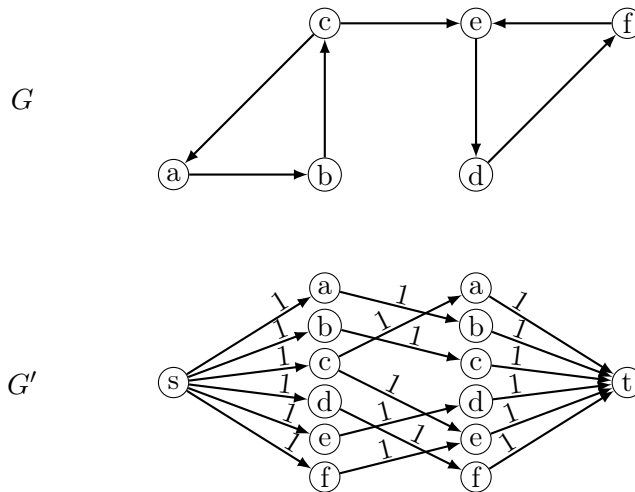
Show that the Cycle Cover problem can be solved in polynomial time by reducing it to the maximum-flow problem. Remember to provide a complete proof. *Hint.* Start with a bipartite graph with one copy of nodes on the left and another copy on the right.

Solution.

- (a) We want to reduce the Cycle Cover problem to Maximum-flow problem. Thus we take an input of the Cycle Cover problem, i.e a graph $G = (V, E)$, and we create an input of Maximum-flow problem. We create two different copies of V , V_{out} and V_{in} , we take $G' = (V', E')$ the directed graph with

$$\begin{aligned} V' &= \{s\} \cup V_{out} \cup V_{in} \cup \{t\} \\ E' &= (\{s\} \times V_{out}) \cup (V_{out} \times V_{in} \cap E) \cup (V_{in} \times \{t\}) \end{aligned}$$

where s and t are new nodes corresponding to the *source* and the *target* and the capacity c with for all $e' \in E'$, $c(e') = 1$.



As $|V'| = 2|V| + 2$ and $|E'| = 2|V| + |E|$, the size of G' is in $O(|V| + |E|)$.

Now we want to prove that G has a cycle cover **if and only if** G' has a feasible flow of size $|V|$ with only integral values.

- (b) (\Rightarrow) Assume that G has a cycle cover C represented as a set of arcs defining the cycles. As a cycle cover is formed of vertex-disjoint simple cycles, we can decompose C in a set of vertex-disjoint simple cycles $\{c_1, \dots, c_m\}$. As $\{c_1, \dots, c_m\}$ is vertex-disjoint, we can partition V in V_1, \dots, V_m using the set of vertex cover by each cycle. As for each V_i the number of vertices in V_i corresponds to the number of arcs in c_i , the total number of arcs is equal to the number of vertices in V , i.e. $|C| = |V|$.

We take the application f from E' to \mathbb{N} which is initialized to 0 and for each $(u, v) \in C$, we add 1 to $f((s, u))$, $f((u, v))$ and $f((v, t))$. As for each $(u, v) \in C$ we create a path from s to t , f is a flow (respects the flow conservation property).

As C is formed of vertex-disjoint simple cycles, for all (u, v) and $(u', v') \in C$, we have $u \neq u'$ and $v \neq v'$. Hence, we can prove by contradiction that for all $e' \in E'$, $f(e') \in \{0, 1\}$. Indeed, we assume that there exists (u, v) such that $f((u, v)) \geq 2$. As E is not a multiset (G has no multiple edges between the same pair of vertices), we know that if $(u, v) \in E$, $f((u, v)) \in \{0, 1\}$, thus it is impossible. If $u = s$, there exists

w and w' such that (v, w) and $(v, w') \in C$ which is impossible. If $v = t$, there exists w and w' such that (w, u) and $(w', u) \in C$ which is also impossible. Hence, the flow f is feasible.

As for all $e' \in E'$, $f(e') \in \{0, 1\}$, the size of f corresponds to the number of $u \in V_{out}$ such that $f((s, u)) = 1$, which corresponds to the number of arcs of C . As $|C| = |V|$, the size of f is $|V|$. Hence there exists a feasible flow of size $|V|$ with only integral values.

- (c) (\Leftarrow) Assume that G' has a feasible flow f of size $|V|$ with only integral values. As f is a feasible flow with only integral values, we have that for all $e' \in E'$, $f(e') \in \{0, 1\}$. As the size of f is $|V|$, $|\{(s, u) \in E'\}| = |V_{out}|$ and $|V_{out}| = |V|$, for each vertex u of V_{out} , we have $f((s, u)) = 1$. As f is a flow and satisfies the flow conservation property, for each $u \in V_{out}$, there exists a unique $(u, v) \in E$ such that $f(u, v) = 1$. We denote by C the following set of $|V|$ elements $\{(u, v) : u \in V_{out}, (u, v) \in E \text{ and } f(u, v) = 1\}$. We are going to prove that C is a cyclic cover of G . As for all $u \in V_{out}$ there exists $(u, v) \in C$ and V_{out} is a copy of V , C covers all the vertices of V . By the definition of C , for each $u \in V$, there exists a unique v such that $(u, v) \in C$ and as $|V| = |C|$, C is a set of cycles and thus a cycle cover of G .

Grading. 5 points for each item. In detail

- (a) **1 point** for taking an input of Cycle Cover, **1 point** for creating a flow network, **1 point** for giving a good network and **2 points** to give the good equivalence between the problems.
- (b) **1 point** for taking a cyclic cover, **1 point** to create a flow with this input, **1 point** to prove that is a flow and **2 points** to prove that this flow is feasible.
- (c) **1 point** for taking a flow with the good properties, **1 point** to create a cyclic cover with the flow, **1 point** to prove that cover G and **2 points** to prove that is a set of cycles.

3. Dynamic programming

Recall the algorithm to compute the length of the longest increasing subsequence in quadratic time. Modify the algorithm so that it computes the length of the longest increasing subsequence that can be partitioned to k pieces that can be found in that order as subarrays of the input. For example, with input $A = 1, 3, 5, 2, 6, 7, 2, 4, 9, 12, 15$, and $k = 3$, the longest increasing subsequence $= 1, 3, 5, 6, 7, 9, 12, 15$ is also the solution to this restricted case as it can be partitioned into $[1, 3, 5], [6, 7], [9, 12, 15]$, where the pieces are found in that order in A . With $k = 2$ the optimal solution is $[1, 3, 5], [9, 12, 15]$, and thus differs from the unrestricted optimum. *Hint.* Consider computing a two-dimensional table $T[0 \dots n][0 \dots k]$ with value $T[i][j]$ defined as the length of the longest increasing subsequence in the subarray $A[1..i]$ consisting of j contiguous pieces such that the last piece ends in the i th element, or $-\infty$ if no such subsequence exists. Find a recurrence to compute the values in a suitable evaluation order. Consider also initialization and finalization.

Solution.

We solve the problem using dynamic programming, computing for all $0 \leq i \leq n$ and $0 \leq j \leq k$ the value $T[i][j]$ defined as the length of the longest increasing subsequence in the subarray $A[1..i]$ consisting of j contiguous pieces such that the last piece ends in the i th element, or $-\infty$ if no such subsequence exists. For convenience, we define that if there are no pieces, the last piece ends in $A[0] = -\infty$. By this definition, the final result is given by $\max_{0 \leq i \leq n} T[i][k]$.

From the definition we get that $T[0][0] = 0$ and if one of the equalities $i = 0$ or $j = 0$ holds, then $T[i][j] = -\infty$. Let us now compute $T[i][j]$ for $1 \leq i \leq n$ and $1 \leq j \leq k$ recursively.

Constructing an increasing subsequence consisting of j contiguous pieces ending in $A[i]$ can be done in two ways: extending the last piece of a subsequence with j contiguous pieces ending in $A[i-1]$ provided that $A[i-1] < A[i]$ or by creating a new piece into an increasing subsequence consisting of $j-1$ contiguous pieces ending in $A[a]$ where $0 \leq a < i$ and $A[a] < A[i]$. If $A[i-1] \geq A[i]$, we get the recurrence

$$T[i][j] = \max_{\substack{0 \leq a < i \\ A[a] < A[i]}} T[a][j-1] + 1$$

and if $A[i-1] < A[i]$, we get the recurrence

$$T[i][j] = \max \left\{ T[i-1][j] + 1, \max_{\substack{0 \leq a < i \\ A[a] < A[i]}} T[a][j-1] + 1 \right\}.$$

Here we use the convention that a maximum over an empty set is $-\infty$. By definition, the value $T[i][0]$ is 0 if $i = 0$ and $-\infty$ otherwise. The straightforward implementation of this recurrence runs in $O(kn^2)$.

Grading.

- 8 points from the correct recurrence.
- 5 points from the correct initialization.
- 2 points from the correct finalization.

4. NP-completeness of vertex cover revisited

In this assignment you should complete an NP-completeness proof of vertex cover using the following sketch of a reduction from 3-CNF-SAT. In *vertex cover* problem, one asks if there is a subset of nodes of size k in an undirected graph that cover all edges. That is, each edge should have at least one endpoint inside the subset forming a cover. Recall that we already proved vertex cover NP-complete, but for this assignment, assume you do not know it yet.

Recall also that the input of 3-CNF-SAT problem is a boolean formula of the following form:

$$\phi = (\ell_{1,1} \vee \ell_{1,2} \vee \ell_{1,3}) \wedge (\ell_{2,1} \vee \ell_{2,2} \vee \ell_{2,3}) \wedge \cdots \wedge (\ell_{m,1} \vee \ell_{m,2} \vee \ell_{m,3}),$$

where each literal $\ell_{i,j}$ is either x_t or $\neg x_t$, for some $t > 0$, and $\{x_1, x_2, \dots\}$ is the set of variables that may occur in ϕ . Recall that we proved that deciding if 3-CNF-SAT formula ϕ has a truth assignment is an NP-complete problem, and for this assignment, you can assume it known.

Consider a reduction that creates a graph with the $3m$ literals as nodes. Add edges to connect all literals inside the same clause to each other. Then for each literal add also an edge to its negation in all other clauses. (Note that this is the complement graph of the one we considered at the lecture for another problem.) That is, for

$$\phi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$$

we would create a graph with six nodes $v_{1,1}, v_{1,2}, v_{1,3}, v_{2,1}, v_{2,2}, v_{2,3}$ corresponding to the literals in the two clauses in the same order. Edges would be $(v_{1,1}, v_{1,2}), (v_{1,1}, v_{1,3}), (v_{1,2}, v_{1,3}), (v_{2,1}, v_{2,2}), (v_{2,1}, v_{2,3}), (v_{2,2}, v_{2,3}), (v_{1,1}, v_{2,1}), (v_{1,3}, v_{2,3})$.

Complete the details of this reduction to prove vertex cover NP-complete. *Hint.* Think of vertex cover of size $k = 2m$.

Solution.

Vertex cover belongs to NP because it admits a certificate of the following form: the set S of vertices in the cover. To check such a certificate in polynomial time, we inspect each edge of G and check whether one of its endpoints belongs to S .

To prove that vertex cover is NP-hard, observe that the reduction proposed can be done in polynomial time. We still need to prove that ϕ is satisfiable $\Leftrightarrow G$ (i.e., the graph proposed in the reduction) has a vertex cover of size $2m$.

“ \Rightarrow ”: Let $f(\cdot)$ be an assignment of truth values to the variables of ϕ (i.e., $f(x_t)$ is either 0 or 1) rendering ϕ true. For each clause c_i of ϕ , let $\ell_{i,j}$ be one literal in c_i set to true by f . We select in the cover S the two nodes of c_i *different* from $\ell_{i,j}$. Observe that the size of S is $2m$. The set S is also a vertex cover because:

- all edges inside the same clause are covered by S , as we select two nodes from each clause;
- any possible edge between clauses c_i and c_j is of the form $(x_t, \neg x_t)$. Thus, at least one of its endpoints is selected in the cover for c_i or for c_j (i.e., not both of x_t and $\neg x_t$ were set to true by f).

“ \Leftarrow ”: Let S be a vertex cover of G of size $2m$. Observe that any vertex cover of G must select at least 2 nodes from each clause. Since $|S| = 2m$, then S selects exactly 2 nodes from each clause. We construct a truth assignment f by setting to true that literal of c_i not in S , for each c_i . If consistent (i.e., we do not assign true to both literals x_t and $\neg x_t$), then f renders true each clause, and thus renders ϕ satisfiable. To see that it is consistent, assume for a contradiction that both x_t and $\neg x_t$ were assigned to true in distinct clauses c_i and c_j , respectively. Thus, neither literal x_t or $\neg x_t$ is in S . Thus the edge $(x_t, \neg x_t)$ is not covered by S , a contradiction.

Grading.

- 1 point: arguing that vertex cover belongs to NP and that the reduction given in the assignment works in polynomial time.
- 2 points: mentioning the equivalence ϕ is satisfiable $\Leftrightarrow G$ has a vertex cover of size $2m$ (either explicitly or implicitly)
- 6 points: proving the “ \Rightarrow ” implication
- 6 points: proving the “ \Leftarrow ” implication

Note: This is an "**open book exam**": You are allowed to access any material, including the course material in online or printed form. However, it is strictly **forbidden to copy answers** (except for brief, clearly marked quotations), **or to interact with other people** to discuss the exam during the exam.

You can answer in Finnish, Swedish, or English.

Answer four (4) problems out of five (5). If you answer all problems, only the first four will be taken into account. Each problem is worth max 10 points.

1. Philosophy, History, and Ethics of AI

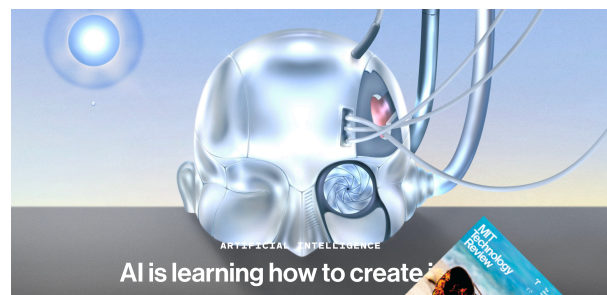
- a. (4 p) Essay. A good length is about 2/3 of a page. Topic: "The next generation of AI: Hybrid neuro-symbolic systems?"

It has been suggested that neither the "good-old fashioned AI" (GOFAI) (methods based mostly on symbolic representations) or the modern AI (methods such as machine learning and neural networks) alone can solve all problems, and that what we'll need to get to the next generation of AI is hybrid systems that combine the "best of both worlds".

Briefly discuss how you see the future of AI and to what extent such hybrid "neuro-symbolic" systems might be the missing piece.

- b. (3 p) What do you think about the following headline and illustration (see right; MIT Tech Review, May 2021):

"AI is learning how to create itself – Humans have struggled to make truly intelligent machines. Maybe we need to let them get on with it themselves."



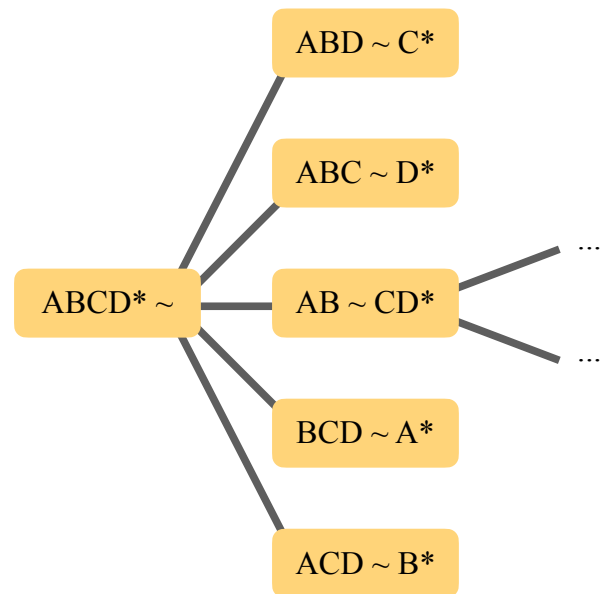
How well does this reflect your understanding of the capabilities of current AI?

- c. (3 p) In what ways might AI be a threat to human rights? Mention at least two threats and propose ways to avoid these threats when developing and deploying AI systems.

2. Problem Solving by Search

Let's solve a puzzle: Alice (A) and Bob (B) are the proud owners of two remote-controlled robot dogs, named Curveball (C) and Dumbo (D). They want to cross a river with a tiny boat that can only fit only a) one person or b) up to two robot dogs (not, for example, one person and one robot dog). The boat has an electric outboard motor and the robot dogs are capable of operating the boat on their own. How to get all of them across the river dry?

- a. (5 p) To get started, we'll draw a state-transition diagram. We've already gotten started with it, see right. The notation shows each of our protagonists as the letters A,B,C,D on either the left or the right side of the river ('~'). The star ('*') indicates on which side of the river the boat is in each state.



For example, in the starting state (the leftmost node in the diagram), everyone is still on the left side with the boat. There are five possible transitions: one for each of the four agents traveling to the right side alone, plus one where Curveball and Dumbo go together (the middle one: "AB ~ CD*").

Draw the rest of the state-transition diagram. Remember to draw each state only once, or otherwise your diagram will become infinitely large. Note that all the transitions are two-way. You shouldn't get more than a couple dozen states in your diagram. If you get more, something's off.

- b. (1 p) Using the state-transition diagram, provide a solution to the puzzle in the form of a sequence of transitions from the state "ABCD* ~" to the state "~ ABCD*".
- c. (2 p) What search algorithm would be a good choice in this case? What search algorithm would be a poor choice? Why?
- d. (2 p) What types of real-world problems can be solved by this type of search techniques? What types of real-world problems can't?

3. Reasoning under Uncertainty

Suppose we are building a probabilistic model of the following scenario: The risk of catching a contagious disease¹ depends on the age of a person and whether they have been exposed to an infected person. The symptoms of the disease include fever which occurs more often with elderly patients than with the young. There is a test that can identify the disease but the test isn't perfect.

- a. (3 p) Draw a Bayesian network to describe the above scenario.

The network should contain five nodes: Age (young/old), Exposed (yes/no), Disease (yes/no), Fever (yes/no), and Test (pos/neg). The parents of each node should include the variables that influence the node in question: for example, the parents of the node Disease, should be Age and Exposed. (You can assume, for the sake of simplicity, that Age and Exposed are independent of each other, and that the test accuracy is the same for everyone.)

- b. (3 p) Come up with imaginary, but reasonable, conditional probability tables for the network.
- c. (2 p) Pretend to draw a sample of size $n = 10$ from the network. List the values of the five variables for each of the ten cases (or tuples) like this:

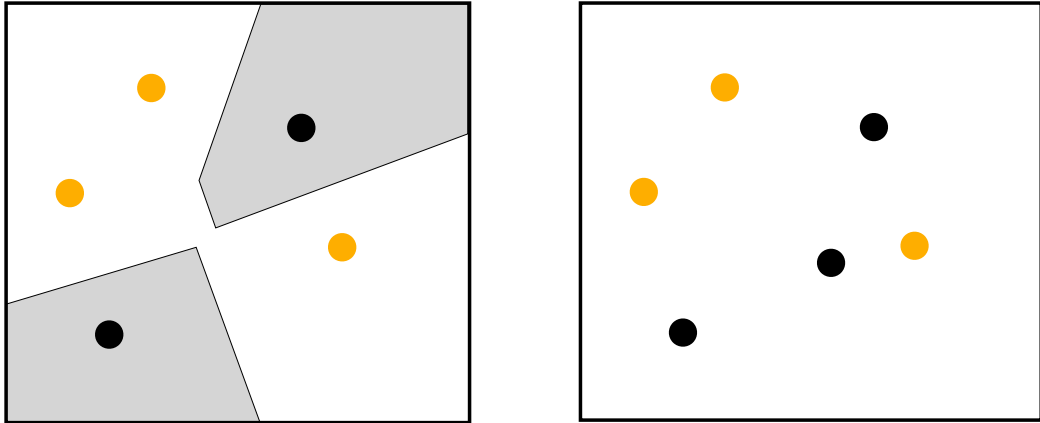
AGE	EXPOSED	DISEASE	FEVER	TEST
young	yes	no	no	neg
old	yes	yes	yes	pos
...				

- d. (2 p) What could such a network be useful for? How would it be used?

¹ Any similarity to actual persons or events is purely coincidental.

4. Machine Learning

Consider a simple binary classification task where we classify two-dimensional data points into two classes: orange vs black. Given labeled training data, a nearest neighbour classifier splits the (x,y)-plane into regions based on the class label of the nearest training data point. The left figure below shows an example of the resulting classification boundaries: any test data point in the shaded regions would be classified as black.



- (3 p) Draw a similar diagram showing the classification boundaries for the training data in the figure on the right, where there is one additional black data point.
- (2 p) Do the same with a 3-NN classifier (using the data on the right).
- (3 p) Now suppose we weren't dealing with simple 2D data but, say, images or audio samples. In what kinds of image or audio classification tasks would the k -NN classifier be likely to perform well and in what kinds not so well?
- (2 p) What type of an approach would you try instead in the cases from the previous item where you expect the k -NN classifier to perform poorly?

5. Natural Language Processing

a. (3 p) Consider a context-free grammar with the following rules:

$$\begin{array}{lll} S \longrightarrow S S & B \longrightarrow S B & S \longrightarrow bada \\ S \longrightarrow A B & A \longrightarrow big & B \longrightarrow boom \end{array}$$

S is the starting symbol. From S , we can generate, for example, $S \longrightarrow AB \longrightarrow ASB \longrightarrow \dots \longrightarrow big\ bada\ boom$.

Does the language contain the following three sentences (or strings): “boom boom”, “bada bada bada bada”, “big boom big boom”? Justify your answer.

b. (5 p) Apply the CYK algorithm to fill in the following table in order to determine whether the sentence “big boom bada bada big boom” belongs to the language. *Hint:* Cells (5,5), (6,6), and (5,6) have already been filled in. Recall that, e.g., cell (5,6) covers the words 5–6: “big boom”.

(1,6)					
(1,5)	(2,6)				
(1,4)	(2,5)	(3,6)			
(1,3)	(2,4)	(3,5)	(4,6)		
(1,2)	(2,3)	(3,4)	(4,5)	S (5,6)	
(1,1)	(2,2)	(3,3)	(4,4)	A (5,5)	B (6,6)
<i>big</i>	<i>boom</i>	<i>bada</i>	<i>bada</i>	<i>big</i>	<i>boom</i>

c. (2 p) What is the corresponding parse tree (or parse trees), if any? Does the sentence belong to the language or not?

TKT20001 Tietorakenteet ja algoritmit (kevät 2020)

Kurssikoe 1, malliratkaisut

Tehtävän 1 tarkasti Jyrki Kivinen, tehtävän 2 Jukka Rautaoja, tehtävän 3 Marcus Leivo ja tehtävän 4 Hannu Kärnä.

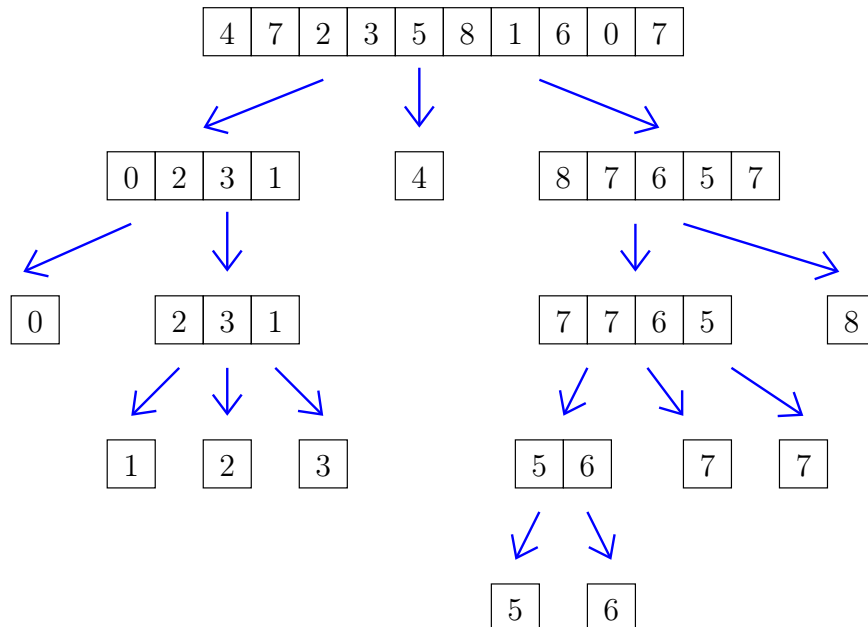
1. [4 pistettä] *Pikajärjestäminen*. Selitä lyhyesti pikajärjestämisen toimintaperiaate. Yksityiskohtaista pseudokoodia tms. ei tarvitse antaa; sopiva vastauksen tarkkuustaso on sanallinen selitys, jota on havainnollistettu parilla kuvalla tai kaaviolla. Mitä tiedetään pikajärjestämisen aika- ja tilavaativuudesta? (Näitä ei tarvitse perustella.)

Hyvän vastauksen pitäisi normaalikäsiälalla mahtua helposti yhdelle sivulle.

Ratkaisu: Pikajärjestämisessä taulukon alkiot järjestetään pienimmästä suurimpaan seuraavasti:

- (1) Jos taulukossa on korkeintaan yksi alkio, lopeta tekemättä mitään.
- (2) Valitse jokin taulukon alkio *jakoalkioksi*.
- (3) Järjestele taulukon alkiot niin, että jakoalkiota pienemmät alkiot tulevat jakoalkion vasemmalle puolelle ja suuremmat oikealle.
- (4) Järjestä jakoalkion vasemmalle puolelle jäävä osataulukko ja oikealle puolelle jäävä osataulukko soveltamalla kumpaankin rekursiivisesti pikajärjestämistä.

Jakoalkion valintaan ja alkioiden järjestelemiseen kohdassa (3) on erilaisia menetelmiä. Seuraava esimerkki havainnollistaa algoritmin toimintaa, kun jakoalkioksi valitaan aina taulukon ensimmäinen alkio (kuva luentomateriaalin sivulta 125):



Algoritmin aikavaativuus on pahimmassa tapauksessa $O(n^2)$ ja keskimäärin $O(n \log n)$, kun taulukon eri järjestykset oletetaan keskenään yhtä todennäköisiksi. (Käytännössä hyvin toteutettu pikajärjestäminen on yleensä yksi nopeimmista ellei

nopein järjestämisalgoritmi.) Tilavaativuus on naiivilla toteutuksella pahimmassa tapauksessa $O(n)$ ja keskimäärin $O(\log n)$; optimoidulla toteutuksella tilavaativuudeksi saadaan myös pahimmassa tapauksessa $O(\log n)$.

Kommentteja vastauksista: Vastauksissa ilmeni hyvin paljon epäselviä tai suoraan virheellisiä käsityksiä. Seuraavassa käydään läpi yleisimpiä niistä, vaikka niistä ei kaikista olekaan arvostelussa vähennetty pisteitä.

Kun algoritmin keskimääräinen aikavaativuus on $O(n \log n)$, se ei tarkoita, että aikavaativuus on $O(n \log n)$ ”yleensä”, ”hyvin toteutettuna” tms. Kysymyksessä on täsmällinen matemaattinen väittäminen: kun syöte valitaan satunnaisesti tiettyjen oletusten vallitessa, niin suoritusajan odotusarvo on $O(n \log n)$. Pikajärjestämisen (ja yleensä järjestämisalgoritmien) tapauksessa nämä oletukset ovat, että taulukon alkio ovat eri suuria ja niiden jokainen järjestys esiintyy yhtä todennäköisesti.

Pikajärjestämistä voisi tarkemmin sanoa perheeksi algoritmeja, koska jakoalkion valintaan ja siihen liittyvään taulukon järjestelemiseen on erilaisia vaihtoehtoja. Perusversiossa jakoalkioksi valitaan taulukon ensimmäinen alkio, mutta todelliseen käyttöön tarkoitettussa toteutuksessa parempi (mutta silti yksinkertainen) valinta olisi taulukon ensimmäisen, viimeisen ja keskimmäisen alkion mediaani, tai yleisemmin esim. viiden tasavälisesti poimitun alkion mediaani. Kaikilla näillä jakoalkion valintamenettelyillä algoritmin aikavaativuus on $O(n \log n)$ keskimäärin ja $O(n^2)$ pahimmassa tapauksessa.

Miksi sitten ensimmäisen alkion valitsemista pidetään huonona, vaikka se johtaa samaan aikavaativuusluokkaan kuin parempina pidetyt? Pääasiallinen syy on, että ensimmäisen alkion valitsevan algoritmin (eräs) pahin tapaus on valmiiksi järjestetty taulukko, jollaisia (ainakin suunnilleen) voi helposti kuvitella esiintyvän käytännön sovelluksissa. Useamman alkion mediaaniin perustuvilla algoritmeilla on niillekin mahdollista löytää tapauksia, joissa aikavaativuus on $O(n^2)$, mutta tämä vaatii huomattavasti keinoitekoisempia konstruktioita, joiden esiintyminen käytännössä on vähemmän luultavaa.

(Teoriassa on mahdollista valita jakoalkioksi koko taulukon mediaani ajassa $O(n)$. Tämä ei kuitenkaan ole käytännöllistä, koska tarvittava algoritmi on melko monimutkainen ja sen käyttäminen johtaisi suuriin vakiokertoimiin, mikä poistaisi keskeisen syyn ylipäänsä käyttää pikajärjestämistä eikä muita $O(n \log n)$ -algoritmeja.)

Monessa vastauksessa oli mainittu, että ensimmäisen alkion valitseminen johtaa aikavaativuuteen $O(n^2)$, jos syötetaulukko on laskevassa järjestyksessä (kun se halutaan nousevaan). Tämä on kyllä totta, mutta oleellisempaa ja helpommin nähtävää on, että valmiiksi oikeassa järjestyksessä oleva taulukko johtaa tähän aikavaativuuteen.

Jakoalkion valinnan lisäksi algoritmin tehokkuuteen vaikuttaa, miten alkioita siirrellään halutun jaon aikaansaamiseksi. Esim. jos taulukon kaikki alkio ovat yhtä suuria, on tietysti samantekevää, mikä niistä valitaan jakoalkioksi. Kurssimateriaalissa esitetyllä jako-proseduurilla on sekin huono puoli, että tässä tilanteessa se taas johtaa aikavaativuuteen $O(n^2)$. Tämä voitaisiin välttää toisenlaisella alkioiden siirtelystrategialla, jota emme kuitenkaan tässä rupea selvittämään.

Tilavaativuuden muodostumisessa tuntui olevan jonkin verran epäselvyyttä. Pikajärjestäminen ei käytä aputaulukoita. Tilavaativuus ei silti ole $O(1)$, sillä rekursion synnyttämä aktivaatietietuepino vaatii tilan $O(\log n)$.

Hyvin monessa vastauksessa algoritmin toiminnan kuvauksessa todettiin, että kun alku- ja loppuosa on erikseen järjestetty rekursiivisesti, osataulukot yhdistetään. Ei niitä yhdistetä. Ne ovat valmiiksi yhdessä. Toisin kuin lomitusjärjestäminen, pika-järjestäminen tekee kaikki alkioiden siirtelyt taulukon alkuperäisen talletusalueen sisällä. Kun alku- ja loppuosa on kumpikin erikseen järjestetty, lopputulos on samalla hetkellä valmis ilman lisätoimia. Tämä osaltaan selittää, miksi kurssimateriaalissa esitetty jako-proseduuri (ja muut kirjallisuudesta löytyvät) on niin epäintuitiivinen: vaatii hieman temppuilua, että haluttu jaottelu saadaan syntymään ilman aputaulukon käyttämistä.

Arvosteluperusteet: Kolme pistettä tuli algoritmin toiminnan selityksestä ja yksi piste aika- ja tilavaativuudesta.

Toiminnan selityksen osalta

- yhteen pisteeseen riitti jonkinlainen kohtuullisen oikeansuuntainen selitys
- kahteen pisteeseen vaadittiin, että selityksen perusteella opiskelija vaikuttaa ymmärtäneen algoritmin jokseenkin oikein
- kolmeen pisteeseen vaadittiin, että toiminta oli kuvattu selvästi ja ymmärrettävästi.

Vaikka tehtävänannossa vihjattiin, että kuva olisi hyödyllinen, sellaista ei kuitenkaan ehdottomasti vaadittu, jos selitys oli muuten hyvin selvä. Tältä kannalta jako-proseduurin pseudokoodin kopioiminen vastauspaperiin ei ole hyvä havainnollistuskäyttö. Erilaisten jakoalkion valintamenettelyjen esittelyä ei ole vaadittu, mutta jos vastauksesta ei mitenkään käy ilmi, että ensimmäisen alkion valitseminen ei ole ainoa mahdollisuus, on vähennetty 0,5 pistettä.

Aikavaativuudesta sai 0,5 pistettä toteamalla, että keskimääräisen tapauksen aika-vaativuus on $O(n \log n)$ ja pahimman tapauksen $O(n^2)$. Nämä kumpikin vaadittiin. Pisteitä ei ole vähennetty siitä hyvin yleisestä puutteesta, että aikavaativuuteen $O(n \log n)$ liittyvä ”keskimääräinen tapaus” on selitetty hyvin epämääräisesti (katso kommentteja tuonnempana). Tilavaativuuden osalta 0,5 pistettä on saanut toteamalla joko, että se on keskimäärin $O(\log n)$ ja pahimmassa tapauksessa $O(n)$ (mikä pätee kurssimateriaalissa esitetyle yksinkertaiselle toteutukselle) tai että se on aina $O(\log n)$ (mikä pätee optimoidulle toteutukselle). Toteamalla tilavaativuudeksi $O(n)$ ilman mitään täsmennystä ei ole saanut pisteitä.

2. [2 + 3 pistettä]

- (a) Määrittele, millainen on *binäärihakupu*. Mitä ehtoja binäärihakupuun pitää täyttää, että se olisi *AVL-puu*?

Ratkaisu: Binäärihakupuussa jokaisen solmun avain on suurempi kuin mikään avain sen vasemmassa alipuussa ja pienempi kuin mikään avain sen oikeassa alipuussa. Binäärihakupu on AVL-puu, jos jokaisessa solmussa vasemman ja oikean alipuun korkeuksien erotus on -1, 0 tai 1.

Pisteytys: kummastakin seuraavasta erikseen yksi piste:

- On kerrottu, että binäärihakupu koostuu solmuista, jotka on järjestetty niin, että puussa on helppo löytää suurin ja pienin alkio.
 - AVL-puun tasapainoehto kerrottu. Jokaisen solmun vasemman ja oikean alipuun korkeusero on enintään 1.
- (b) Esitä yksityiskohtaisena pseudokoodina algoritmi, joka saa parametrina osoittimen binääripuun juureen ja palauttaa puun lehtien lukumäärän.

Algoritmin aikavaativuuden tulee olla $O(n)$, missä n on puun solmujen lukumäärä. Aikavaativuutta ei tarvitse erikseen perustella.

Ratkaisu:

```
LEHDET(solmu)
  if solmu == null
    then return 0
  elseif solmu.vasen == null and solmu.oikea == null
    then return 1
  else return LEHDET(solmu.vasen) + LEHDET(solmu.oikea)
```

Pisteytys: kustkin seuraavasta erikseen yksi piste:

- Algoritmi laskee oikein lehtien eli lapsettomien solmujen lukumäärän.
- Algoritmi palauttaa oikein lehtien lukumäärän.
- Algoritmi ei aiheuta virhettä yrittämällä seurata null-osoitinta eikä ajaudu ikuiseseen rekursioon/toistolauseeseen.

Kommentteja:

- Pisteet eivät ole riippuvaisia toisistaan, toki b-kohdan 1. ja 2. piste ovat lähellä toisiaan.
- Moni laski solmujen lukumäärän tai ei käsitellyt null-osoittimia oikein.

3. [6 pistettä] Mitkä ovat seuraavien algoritmien aikavaativuudet?

Jokaisessa kohdassa anna aikavaativuudelle mahdollisimman tarkka arvio parametrin n funktiona käyttäen iso- O -merkintää ja selitä lyhyesti, miten päättelit aikavaativuuden. Selitykseksi riittää virke tai pari, joissa mainitset käyttämäsi yleiset periaatteet tms.; tarkkoja matemaattisia todistuksia ei tarvita.

Algoritmit eivät tee mitään erityisen hyödyllistä. Kolme pistettä "...” tarkoittaa jotain vakioajassa tapahtuvaa laskentaa. Kohdissa (a)–(e) ilmoita annetun koodinpätkän aikavaativuus, kohdassa (f) kutsun $f(n)$ aikavaativuus.

```
(a)   for i=1 to n
        ...
        for i=1 to n
            for j=1 to n
                ...
```

Ratkaisu: Ensimmäinen silmukka suoritetaan n kertaa, joten sen aikavaativuus on $O(n)$. Toisessa silmukassa sisin osuus suoritetaan $n \cdot n$ kertaa, joten sen aikavaativuus on $O(n^2)$. Laitettaessa nämä kaksi osuutta peräkkäin suurempi aikavaativuus määrää kokonaisuuden aikavaativuuden, joten koko algoritmin aikavaativuudeksi tulee $O(n^2)$.

```
(b)   for i=1 to n
        for j=1 to i
            ...
```

Ratkaisu: Sisin osuus suoritetaan i kertaa arvoilla $i = 1, \dots, n$ eli yhteensä $1 + \dots + n = n(n+1)/2 = O(n^2)$ kertaa. Siis aikavaativuus on $O(n^2)$.

```
(c)   j = 0
        while j < n
            ...
            j = j + 3
```

Ratkaisu: Silmukka suoritetaan $\lceil n/3 \rceil$ kertaa, joten aikavaativuus on $O(n)$.

```
(d)   j = 0
        while j < n
            ...
            j = j + n/3 + 1
```

Ratkaisu: Jokainen silmukan suorituskerralla laskuri kasvaa ainakin $n/3$, joten suorituskertojen lukumäärä on korkeintaan 3 eli vakio. Aikavaativuus on $O(1)$.

```
(e)   j = 1
        while j < n*n
            ...
            j = j*2
```

Ratkaisu: Silmukkamuuttujan arvo kaksinkertaistuu joka suorituskerralla. Ylärajan n^2 saavuttamiseen tarvitaan noin $\log_2 n^2 = 2 \log_2 n$ kaksinkertaistusta, joten aikavaativuus on $O(\log n)$.


```
(f)   procedure f(n)
        if n == 0
            return
        f(n-1)
        f(n-1)
```

Ratkaisu: Parametrin n kasvattaminen yhdellä kaksinkertaistaa tehtävän laskennan, joten aikavaativuus on $O(2^n)$.

Kommentteja vastauksista: Todella moni oli yrittänyt perustella kohdan (a) aikavaativuutta sillä, että koodissa oli sisäkkäiset for-silmukat, mutta yleisesti ottaen tämä ei riitä todistamaan että aikavaativuus olisi $O(n^2)$. Kohdassa (b) monet olivat vastanneet aikavaativuudeksi $O(ni)$, mikä ei ole lainkaan mielekäs vastaus. Muuten tehtävä oltiin osattu suhteellisen hyvin.

Pisteytys: 0,5 pistettä oikeasta aikavaativuudesta ja 0,5 pistettä oikeasta perustelusta. Perusteluiden tuli olla tarkkuustasoltaan suunnilleen samaa luokkaa mallivastauksien kanssa.

4. [5 pistettä] Syötteenä on taulukko, jossa on n mielivaltaisen suurta positiivista kokonaislukua. Tehtävänä on etsiä taulukosta kahta lukua, joista pienempi on tasan puolet suuremmasta; siis luvut x ja y , joilla $y = 2x$. Jos tällaiset kaksi lukua löytyy, vastauksena tulee palauttaa niistä pienempi. Jos ratkaisuja on useita, mikä tahansa niistä kelpaa. Jos tällaisia kahta lukua ei ole taulukossa, vastauksena tulee palauttaa -1 .

Suunnittele ongelmaan ratkaisualgoritmi, jonka pahimman tapauksen aikavaativuus on $O(n \log n)$. Ratkaisussasi esitä algoritmi tehtäväpaperin alussa selitettyjen periaatteiden mukaisesti. Selitä myös lyhyesti ratkaisusi toimintaperiaate, erityisesti sen mahdollisesti käyttämät tietorakenteet. Perustele algoritmisi aikavaativuus.

Esimerkki Jos syötteenä on taulukko $[3, 22, 7, 5, 11, 5]$, algoritmi palauttaa arvon 11, sillä $22 = 2 \cdot 11$. Jos taas syötteenä on taulukko $[8, 5, 14, 13, 6, 1]$, algoritmi palauttaa -1 , koska mikään luku ei ole tasan puolet jostain toisesta.

Ratkaisu 1: Käydään taulukko kerran läpi ja talletetaan sen alkiot aluksi tyhjään tasapainotettuun binäärihakupuuhun. Sen jälkeen käydään taulukko toisen kerran läpi ja jokaisen alkion x kohdalla tarkastetaan, löytyykö $2x$ hakupuusta. Jos löytyy, palautetaan x . Jos taulukon läpikäynti ei johda tulokseen, palautetaan -1 .

Algoritmissa käydään n -alkiainen taulukko kaksi kertaa läpi, ja kummallakin läpikäynnillä tehdään jokaista taulukon alkion kohti yksi tasapainotetun hakupuun operaatio. Hakupuussa on enimmillään n avainta. Siis lisäys- ja hakuoperaatio menevät ajassa $O(\log n)$. Kokonaisaikavaativuus on $O(n) \cdot O(\log n) + O(n) \cdot O(\log n) = O(n \log n)$.

Ratkaisu 2: Järjestetään ensin taulukko lomitusjärjestämällä. Sen jälkeen käydään taulukko alkio kerrallaan läpi, ja alkion x kohdalla tarkastetaan binäärihaulla, onko alkio $2x$ taulukossa. Jos $2x$ löytyi, palautetaan x . Jos taulukon läpikäynti ei johda tulokseen, palautetaan -1 .

Lomitusjärjestäminen vie ajan $O(n \log n)$. Koska läpikäynnissä tehdään jokaista taulukon alkion kohti yksi binäärihaku ajassa $O(\log n)$ ja lisäksi vakioaikaista laskentaa, läpikäynti menee ajassa $O(n \log n)$, joka siis on myös algoritmin kokonaisaikavaativuus.

Ratkaisu 3:

```

ETSI TUPLA(A[0...n-1])
  MERGESORT(A)
  vasen = 0
  oikea = 1
  while oikea < n
    if A[oikea] == 2 * A[vasen]
      then return A[vasen]
    if A[oikea] < 2 * A[vasen]
      then oikea = oikea + 1
    else vasen = vasen + 1
  return -1

```

Algoritmin aikavaativuus on $O(n \log n)$, joka tulee lomitusjärjestämisestä. While-silmukassa aina $vasen \leq oikea < n$ ja muuttujista $vasen$ ja $oikea$ toinen kasvaa

yhdeksi, joten silmukka suoritetaan korkeintaan $2n$ kertaa. Koska laskenta silmukan sisällä on vakioaikaista, silmukan aikavaativuus on $O(n)$.

Selvästi jos algoritmi palauttaa jonkin muun arvon kuin -1 , se on kelvollinen ratkaisu. Perustellaan vielä, että jos kelvollinen ratkaisu on olemassa, algoritmi aina löytää sen. Oletetaan tätä varten, että joillain p ja q pätee $A[q] = 2 \cdot A[p]$.

Niin kauan kuin $vasen \leq p$, muuttuja *oikea* ei voi saada arvoa $q+1$. Tällöin nimittäin taulukon järjestyksen takia $A[q] = 2 \cdot A[p] \geq 2 \cdot A[vasen]$, joten tilanteessa $oikea = q$ ehto *oikea*-muuttujan kasvattamiselle ei toteudu.

Vastaavasti kun $oikea \leq q$, muuttuja *vasen* ei voi saada arvoa $p+1$: pätee $A[oikea] \leq A[q] = 2 \cdot A[p]$, joten tilanteessa $vasen = p$ ehto *vasen*-muuttujan kasvattamiselle ei toteudu.

Siis niin kauan kuin $vasen \leq p$, pätee myös $oikea \leq q$; ja niin kauan kuin $oikea \leq q$, pätee myös $vasen \leq p$. Koska aluksi kumpikin ehto pätee, ne pysyvät kumpikin voimassa loppuun asti. Jos algoritmi ei palauta jotain toista ratkaisua, niin joka askelella joko *vasen* tai *oikea* kasvaa. Siis lopulta saavutetaan tilanne, jossa $vasen = p$ ja $oikea = q$, ja algoritmi palauttaa arvon $A[p]$.

Kommentteja vastauksista: Binäärihaku ei toimi järjestämättömässä taulukossa. Pikajärjestäminen voi pahimmassa tapauksessa vaatia ajan $O(n^2)$ ja yksi hajautus-taulun operaatio ajan $O(n)$.

Koska luvut olivat mielivaltaisen suuria, ei tehtävää voinut ratkaista oikein esim. kirjaamalla luvut boolean-tilaan, jossa indeksiä k vastaava arvo kertoo, onko luku k syötteessä. Sitä varten pitäisi voida tallettaa mielivaltaisen suuri taulukko. Siten ei myöskään ole olemassa hajautusfunktiota, joka pahimmassa tapauksessa jakaisi talletettavat arvot tasaisesti hajautustauluun. Oli talletustila miten suuri vain, voidaan riittävän suuria lukuja valitsemalla muodostaa syöte, jonka jokaisella luvulla on sama hajautusarvo.

Jotkin vastaukset perustuivat väärään oletukseen, että $y = 2x$ tulee syötteessä aina myöhemmin kuin x , että x ja y ovat järjestämisen jälkeen taulukossa peräkkäin, x on aina pariton tai y on taulukon suurin arvo.

Joskus etsittiin nousevaan järjestykseen järjestetystä taulukosta x :ää vain y :n oikealta puolelta tai y :tä x :n vasemmalta puolelta. Vastaava tapahtui hajautustaulua käyttäen, jos sinne lisättiin taulukon alkioita pienimmästä alkaen ja samalle etsittiin alkion arvoa kahdella kerrottuna. Tupla-arvo ei vielä sillä hetkellä voi olla talletettuna. Ratkaisu toimi oikein – oli taulukko järjestetty tai ei – jos kunkin lisäyksen jälkeen etsittiin sekä kaksinkertaista arvoa että puolta pienempää.

Jos kuitenkin esim. lisättiin hajautustauluun sekä taulukon alkion arvo että arvo kerrottuna kahdella, ja sitten etsittiin arvon puolikasta, päädyttiin hyväksymään myös $y = 4x$. Esim. syöte on $\{1, 4\}$. Talletetaan 1 ja $2 \cdot 1 = 2$. Etsitään arvon 1 puolikasta, ei löydy. Talletetaan 4 ja $2 \cdot 4 = 8$. Etsitään arvon 4 puolikasta, löydetään ja palautetaan virheellisesti luku 2 , jota ei syötteessä ollut. Tämän tapaisesta virheestä esiintyi muutamaa muunnelmaa.

Seuraavan silmukan suoritusaika ei ole $O(n)$:

```
i = 1, j = 2
while i < n
```

```

        if t[j] == 2*t[i] return t[i]
        if j == n
            i++
            j = i+1
        else j++
    return -1

```

Silmukan lopetus perustuu muuttujaan i , mutta ensisijaisesti silmukassa kasvate- taankin vain muuttujaa j . Muuttujan i arvo kasvaa vasta, kun j on päässyt arvoon n , jolloin j palaa alkuun. Sama algoritmi tutummin kirjoitettuna:

```

for i = 1 to n-1
    for j = i+1 to n
        if t[j] == 2*t[i] return t[i]
return -1

```

Vaikka j käy i :n kasvaessa yhä pienemmän alueen läpi, ei suoritus-aika ole myöskään $O(n \log n)$, kuten useassa vastauksessa arvioitiin, vaan $O(n^2)$. Keskimäärin sisem- pi silmukka suoritetaan noin $n/2$ kertaa, ei $\log n$ kertaa. Muita toistuvia virheitä aikavaativuuden arvioissa oli, että koko syötteen talletus tasapainotettuun binääri- hakupuuhun tapahtuisi ajassa $O(n)$.

Järjestetyn taulukon pystyy tarkastamaan lineaarisessa ajassa kahta osoitinta käyt- täen mallivastauksen Ratkaisun 3 tapaan. Jos kuitenkin toista osoitinta askelletaan alusta loppuun päin ja toista lopusta alkuun päin tai muuten väärin, jää joko joitakin mahdollisia lukupareja tutkimatta tai sitten algoritmi muuttuu aikavaativuudeltaa neliölliseksi, kun toinen osoitin joudutaan välillä palauttamaan alkupisteeseensä. Lä- pikäynnin optimoinnit eivät pudota aikavaativuutta luokkaan $O(n)$, jos edelleen kuta- kin arvoa verrataan useaan muuhun arvoon, joiden määrä kasvaa suorassa suhteessa syötteen kokoon. Joissakin ratkaisuissa oli käytetty apuna listaa tai toista tauluk- koa, joka oli joko suora kopio järjestetystä taulukosta tai sen arvot kaksinkertaisena / puolitetuna. Myös näin oli mahdollista päätyä löytämään lukupari tavoitellussa ajassa $O(n)$, ohittamaan mahdollisia lukupareja tai käyttämään aikaa $O(n^2)$.

Arvosteluperusteet:

5 pistettä: kattava kuvaus ja aikavaativuusarvio algoritmista, jonka pa- himman tapauksen aikavaativuus on $O(n \log n)$

4,5 pistettä: pieni virhe yllä mainitussa

4 pistettä:

- kattava kuvaus ja aikavaativuusarvio algoritmista, jonka keskimää- räinen aikavaativuus on $O(n \log n)$: käytettiin hajautukseen perus- tuvaa tietorakennetta tai pikajärjestämistä tai määrittelemätöntä järjestämisalgoritmia *tai*
- puutteellinen / virheellinen kuvaus ja aikavaativuusarvio algorit- mista, jonka pahimman tapauksen aikavaativuus on $O(n \log n)$

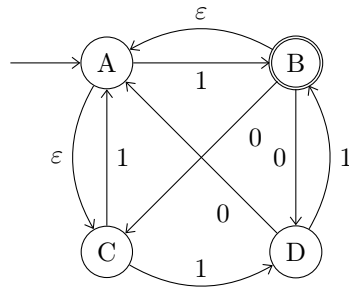
3,5 pistettä:

- pienen virheen sisältävä kuvaus ja aikavaativuusarvio algoritmista, jonka keskimääräinen aikavaativuus on $O(n \log n)$ *tai*

- muuten virheetön algoritmi ja aikavaativuusarvio, jonka toimivuus perustuu siihen, että voidaan käyttää taulukkoa, jonka pituus on yhtä suuri kuin syötteen arvoalue, vaikka arvoalue oli tehtävänannossa määritelty mielivaltaisen suureksi
- 3 pistettä:** puutteellinen / virheellinen kuvaus ja aikavaativuusarvio algoritmista, jonka keskimääräinen aikavaativuus on $O(n \log n)$, esim. vaativuutta ei ole arvioitu lainkaan
- 2,5 pistettä:** algoritmi koostuu peräkkäisistä osista, joiden aikavaativuus on $O(n \log n)$, mikä on analysoitu oikein, mutta ratkaisussa on vakava ajatusvirhe tai puute
- 2 pistettä:** pyritty $O(n \log n)$ -ratkaisuun käyttämällä tehokkaaksi tunnettuja algoritmeja ja tietorakenteita, mutta oleellisesti puutteellinen / virheellinen vastaus, vähän enemmän oikeaa kuin 1,5 pisteen vastauksessa
- 1,5 pistettä:**
- täsmällisesti kuvattu $O(n^2)$ aikavaativuuden ratkaisu *tai*
 - pyritty $O(n \log n)$ -ratkaisuun käyttämällä tehokkaaksi tunnettuja algoritmeja ja tietorakenteita, mutta oleellisesti puutteellinen / virheellinen vastaus; ei ole ilmeistä tapaa korjata ratkaisu toimivaksi, vaan tarvitaan oleellinen muutos, johon ei löydy viitteitä vastauksesta
- 1 piste:**
- aikavaativuuden $O(n^2)$ ratkaisu, jossa puutteita *tai*
 - oleellisesti virheellinen ratkaisu, jossa kuitenkin oikeaa ajatusta ja pyritty tehokkuuteen.

TKT20005 Beräkningsmodeller (hösten 2020)
Mellantentamen I (28.9)

- (4+4 *poäng*) Språket A i alfabetet $\{a, b, c\}$ består av de strängar, som *inte* innehåller delsträngen ab .
 - Ange (rita upp) en deterministisk ändlig automat, som känner igen språket A .
 - Ange ett reguljärt uttryck, som beskriver språket A .
- (6 *poäng*) Omvandla den icke-deterministiska ändliga automaten nedan till en deterministisk sådan genom att tillämpa metoden presenterad under kursen (på föreläsningarna eller i kursboken). Mellansteg behöver inte ritas ut, förutsatt att det av slutresultatet klart framgår, att det uppnåtts med den angivna metoden.



- (2+2+2 *poäng*) Låt X och Y vara språk i alfabetet $\{0, 1\}$. Ange för varje delfråga nedan antingen exempel på två språk X och Y , som uppfyller givna villkor eller alternativt visa, att dylika X och Y inte existerar. Som hjälpmedel kan användas alla på kursen bevisade egenskaper hos reguljära språk, såsom slutenhetsegenskaperna och det faktum, att t.ex. språket $\{0^n 1^n \mid n \in \mathbf{N}\}$ inte är reguljärt.
 - X är reguljärt, Y är inte reguljärt och $X \cup Y$ är reguljärt.
 - X är inte reguljärt, Y är inte reguljärt och $X \cup Y$ är reguljärt.
 - X är reguljärt, Y är reguljärt och $X \cup Y$ är inte reguljärt.
- (4 *poäng*) Visa, att språket $C = \{a^m b^n a^m b^n \mid m, n \in \mathbf{N}\}$ i alfabetet $\{a, b\}$ inte är reguljärt. Tillåtna hjälpmedel är pumpningslemmat och andra på kursen bevisade allmänna egenskaper för reguljära språk, dock inte resultat som direkt säger att ett visst språk inte är reguljärt.

TKT20005 Beräkningsmodeller (hösten 2020) Mellantentamen II (14.10)

Besvara alla deluppgifter. Denna mellantentamen ger maximalt 24 poäng.

1. (4 + 4 poäng)

(a) Ange en kontextfri grammatik för språket

$$\{ a^m b^{2m} a^{2n} b^n \mid m, n \in \mathbf{N} \}.$$

i alfabetet $\{ a, b \}$.

(b) Ange en kontextfri grammatik för språket

$$\{ a^p b^q a^r \mid p, q, r \in \mathbf{N} \text{ och } q > p + r \}$$

i alfabetet $\{ a, b \}$.

2. (6 poäng) Låt P vara ett Javaprogram, x en av dess metoder och s någon av möjliga indata till det. Målet är att bestämma, om metoden x kallas på en enda gång under exekveringen av P , när indata är s . Existerar det för detta beräkningsproblem en lösningsalgoritm, som löser det korrekt för alla fall av (P, x, s) ? Bevisa ditt påstående. För beviset kan användas vilka som helst resultat som presenterats under kursen.

3. (2 + 4 + 4 poäng) En *dominerande mängd* i en oriktad graf är en mängd av dess noder så att varje nod i grafen antingen hör till ifrågavarande mängd eller är granne till en nod som gör det.

Med andra ord, om $G = (V, E)$ är en oriktad graf och $S \subseteq V$ en mängd av noder i G , så är S en dominerande mängd, om för varje nod $v \in V$ mängden $\{v\} \cup \{u \in V \mid (u, v) \in E\}$ innehåller åtminstone ett element ur mängden S .

(a) Ge exempel på en graf, som har en dominerande mängd som består av tre noder, men inte har en som består av två. Motivera varför grafen som angivits har denna egenskap.

(b) I *dominerande mängd-problemet* ges som input en oriktad graf G och ett naturligt tal k . Målet är att bestämma, huruvida grafen G har en dominerande mängd med exakt k noder. Uttryck dominerande mängd-problemet som ett formellt språk.

Visa, att dominerande mängd-problemet hör till klassen NP. Som lösning räcker en schematisk beskrivning av en algoritm som behövs för detta och en kort motivering för att algoritmen har behövliga egenskaper.

(c) I själva verket är det känt, att dominerande mängd-problemet är NP-fullständigt. Hur hänger alltså P vs. NP-problemet samman med lösningsalgoritmerna för det här problemet? Motivera kort med hjälp av resultat presenterade under kursen.

Uppgift 1: Scrum (4p)

Scrum definierar en mängd roller, artefakt, möten och spelregler. Berätta vilka metoder Scrum erbjuder för att lösa problemsituationerna nedan. Det räcker inte att du listar termer. Berätta vad varje botemedel som Scrum erbjuder handlar om och hur den hjälper med problemet som beskrivs.

- Silva och Jussi grälar ofta om ifall enhetstest bör skapas för nya funktionaliteter.
- Jami, som är chef för företagets marknadsavdelning, får ofta briljanta ideér som bör implementeras genast oavsett vad som är på hälft.
- För ofta märker man att ett par programutvecklare, speciellt Kristian och Petrus, har båda kodat varsin version av samma user story. Många dagars arbete kan gå till spillo.
- Ibland är det oklart vilka storyn som borde utföras under sprinten. Det beror lite på vilken av företagets chefer man råkar fråga, speciellt vdn Luukkainens åsikt avviker från de andra chefernas linje.
- Samma problem uppstår i projektet från månad efter månad, t.ex. Git används hur som helst, testerna fungerar ibland och ibland inte. Namngivningen av klasser och metoder är ganska arbiträrt.
- Kalle tycker det är roligt att få programmera helt i fred. Den dåliga sidan är nog den, att ingen riktigt vet när något överhuvudtaget blir färdigt. Ibland tar det månader.
- Dom här sprinterna är roliga. Det är bara lite kaotiskt ibland när ingen riktigt vet vad man borde göra under sprinten. Det är ett mysterium hur utförandet av de olika user storyna påverkar den existerande koden.
- Vi har litet det problemet, att fast vi oftast får allt implementerat, kan det inte i slutet av sprinten sättas i produktion eftersom test-teamet i Indien fungerar med flera veckors fördröjning

Uppgift 2: Kravhantering (4p)

Din kamrat har fastställt följande user story som beskriver *betalningsfunktionaliteten för köp i en nätbutik*. Vad är bra och vad är dåligt och vad fattas från user storyn? Hur skulle man kunna förbättra user storyn?

När köpen i varukorgen betalas (förutom kreditkort fungerar MobilePay och andra möjliga betalningsalternativ), skall det i databasens ORDERS-tavla skrivas leveransadressen. Även tavlan PRODUCTS, som upprätthåller lagersaldot, skall uppdateras. Det lönar sig att observera, att försäljningen under Black Friday är väldigt. Programvaran skall ändå fungera snabbt även under en enorm användarpik. Det lönar sig att med samma programmera en möjlighet att bestämma hur leveransen arrangeras, genom posten eller någon annan leverantör t.ex. UPS. Denna story skall implementeras genast, den är livsviktig.

Uppgift 3: Termer (3p)

Är följande påståenden sanna eller falska?

- Programvarans validering sker genom enhetstestning
- Om användaren är väldigt nöjd, är troligtvis programvarans externa kvalitet i skick
- Testernas tillräckliga radtäckning kan säkras genom utforskande testning
- Med hjälp av ett Burn down -schema kan man uppskatta hur mycket programmerarna har arbetstimmar kvar till förfogande i sprinter som är på gång
- Att följa tillgänglighetsdirektivet (<https://vm.fi/sv/tillganglighetsdirektivet>) är ett så kallat icke-funktionellt krav
- Om man producerar en så kallad minimal viable product, skapar man inte oftast omfattande automatiserade tester för den
- Genom velociteten kan det vara möjligt att uppskatta när programvaran eller en del av den blir färdig
- Även i agila metoder dokumenteras kraven
- Kodgranskning är vattenfall, den hör inte till agil programvaruutveckling
- Scrum of Scrums är ett sätt att försöka se efter att ett företags olika teams kan arbeta enligt ett gemensamt mål
- Inkrementell arkitektur garanterar att kodens interna kvalitet hålls i skick
- Lean startup -metoden förbjuder att ta tekniskt skuld

Uppgift 4: Lean/DevOps (4p)

I samband med Lean pratar man om Just in Time (JIT) -produktion, samt om att minimera cykeltiden för produktionen.

Klargör vilken nytta en kort cykeltid har i programvaruproduktion, hur man försöker minska på cykeltiden i programvaruproduktion och vad DevOps har med saken att göra.

I ditt svar bör du även berätta vad begreppen Just in Time (JIT) -produktion, cykeltiden för produktion och DevOps betyder.

Uppgift 5 (del 1&2): Kodens kvalitet (5p)

Vad menas med programvarans interna kvalitet? (1p)

Vilka problematiska faktorer gällande programvarans interna kvalitet finns i koden nedan? (2p)

Förklara hur du skulle refaktorera koden nedan genom att tillämpa designmönster eller andra lämpliga lösningar. Svaret bör innehålla skiss av kod. Det räcker inte med att lista namn på designmönster. Skriv ändå inte "javaboilerplate" (dvs. klass+tillgänglighetsbestämning osv.) i sin helhet. (2p)