

Tehtävät 1–3 ovat uusintatentti, eli korvaavat välikokeet. Yhteispistemäärään lisätään siis harjoituspisteet.

Tehtävä 4 korvaa harjoituspisteet. Yhteispistemäärään lisätään siis välikokeiden pisteet (tai tehtävien 1–3 pisteet).

Voit siis tehdä vain toisen vaihtoehdoista (tai halutessasi molemmat).

1. [Yhteensä 16p] Vastaa *lyhyesti* seuraaviin kysymyksiin (kukin kohta 4p):

- (a) Algoritmin X aikavaativuus kuuluu luokkaan $O(f(n))$. Mitä tämä *täsmällisesti* ottaen tarkoittaa?
- (b) Mikä on binäärihaun perusidea? Aikavaativuus?
- (c) Mitä etuja ja haittoja hajautuksella on verrattuna tasapainoisiin etsintäpuihin?
- (d) Vertaile seuraavia algoritmeja: Dijkstran algoritmi, Bellman-Fordin algoritmi, Floydin algoritmi: kerro niiden aikavaativuudet, ja mainitse lyhyesti millaisille verkoille / ongelmille ne soveltuvat.

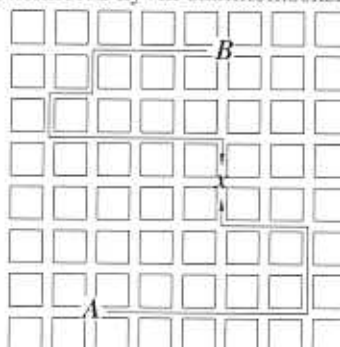
2. [Yhteensä 14p] Olkoon annettu kahteen suuntaan linkitetty lista L , joka sisältää n kappaletta kokonaislukuja. Listaa ei ole järjestetty.

- (a) [5p] Kirjoita algoritmi (pseudokoodi), joka poistaa listasta peräkkäiset samanarvoiset luvut, jättäen jäljelle vain ensimmäisen. Siis jos listassa on aluksi luvut 4, 4, 4, 2, 3, 3, 1, 5, 5, 2, 4, 4 (tässä järjestyksessä), niin lopputuloksena listassa on luvut 4, 2, 3, 1, 5, 2, 4 (tässä järjestyksessä). Algoritmi saa käyttää vain $O(1)$ verran aputilaa. Mikä on aikavaativuus?
- (b) [4p] Kuten kohta (a), mutta nyt jokaisesta alkioista säilytetään vain sen ensimmäinen esiintymä, riippumatta siitä, ovatko esiintymät peräkkäisiä vai ei. Tulos olisi siis 4, 2, 3, 1, 5. Saat vapaasti käyttää kursilla annettuja lista-operaatioita (mutta et mitään aputiitorakenteita) kirjoittamatta niiden pseudokooodeja. Algoritmi ei tarvitse olla erityisen tehokas. Mikä on sen aikavaativuus?
- (c) [3p] Kuten kohta (b), mutta saat käyttää **aputiitorakenteena** hakupuuta (mutta syöte ja tulos ovat edelleen listassa). Lisäksi tiedetään, että eriarvoisia lukuja on listassa vain k kappaletta (k on siis tuloksen pituus, esimerkissämme $k = 5$; mutta lukuja on syötteessä yhteensä edelleen n kappaletta). Pseudokoodia ei tarvita, vaan lyhyt sanallinen selitys riittää. Nyt algoritmin pitäisi olla mahdollisimman tehokas. Mikä on aikavaativuus ja aputilavaativuus? Perustele lyhyesti.
- (d) [2p] Kuten kohta (c), mutta hakupuuta korvataan hajautuksella. Mikä on nyt aikavaativuus ja aputilavaativuus? Perustele.

3. [Yhteensä 10p] Tee vain *toinen* seuraavista vaihtoehdoista (joko (a) tai (b)):

- (a) Olkoon annettu painottoman suunnattu sykliton verkko $G = (V, E)$, joka on esitetty vieruslistoina. Suunnittele mahdollisimman tehokas algoritmi, joka etsii mahdollisimman *pitkän* polun annetusta lähtösolmusta s , annettuun maalisolmuun t . Pseudokoodia ei ole pakko esittää, kunhan esität ratkaisun (sanallisesti) riittävän yksityiskohtaisesti. Kerro algoritmi aikavaativuus Täysien pisteiden saamiseksi algoritmin on lisäksi toimittava ajassa $O(|V| + |E|)$. Perustele lyhyesti miksi algoritmi toimii oikein.

- (b) Henkilöt A ja B ovat jossain risteyksessä kaupungissa G . Kaupunki on rakennettu hyvin säännölliseksi, kaikki korttelit / kadut ovat saman pituisia (katso kuva). A ja B ovat liikkeellä autolla, ja haluavat tavata jossain (missä tahansa) risteyksessä x siten, että vastaavan reitin *kokonaispituus* olisi mahdollisimman lyhyt (toisin sanoen A :n kulkema matka voi olla paljon lyhyempi tai paljon pidempi kuin B :n kulkema matka). Suuri osa kaduista on kuitenkin *yksisuuntaisia*, joten mikä tahansa reitti ei kelpaa. Tehtävänä on etsiä lyhin mahdollinen reitti ajamatta katuja väärään suuntaan. Mallinna ongelma verkkona, ja suunnittele algoritmi joka etsii sopivan risteyksen x ja vastaavan lyhimmän reitin mahdollisimman tehokkaasti. Mikä on algoritmi aikavaativuus? Täysien pisteiden saamiseksi algoritmin on toimittava lineaarisessa ajassa kaupungin pinta-alan suhteen.



4. Vastaa *lyhyesti* (1-2 lauseella) seuraaviin kysymyksiin (kukin kohta 2p):

- (a) Väite: Jos hakupuuta ei ole tasapainoitettu, niin kaikki perus-operaatiot (annetun alkion / minimin / maksimin etsiminen, jne.) toimivat puun korkeuteen verrannollisessa ajassa. Totta vai ei? Jos ei, niin miksi ei?
- (b) Väite: Binäärihaku linkitettyssä listassa on tehokkaampi kuin peräkkäishaku. Totta vai ei? Jos ei, niin miksi ei?
- (c) Väite: Jos jokaiselle binääripuun solmulle x pätee: $x.left.key < x.key < x.right.key$, niin puu on hakupuuta. Totta vai ei? Jos ei, niin mikä on oikea ehto?
- (d) Väite: Jos hakupuun solmujen avaimet tulostetaan esijärjestyksessä, niin avaimet tulostuvat suuruusjärjestyksessä. Totta vai ei? Jos ei, niin onko olemassa jokin muu järjestys jolle väite pätee?
- (e) Erään binääripuun solmujen avaimet ovat leveysuuntaisessa järjestyksessä lueteltuna 3, 2, 5, 1. Saman puun avaimet jälkijärjestyksessä ovat 1, 2, 5, 3. Väite: voidaan päätellä, että puu on hakupuuta. Totta vai ei? Jos ei, niin anna vastaesimerkki.
- (f) Väite: Mikä tahansa hajautusfunktio toimii huonosti *jollakin* syötteellä. Totta vai ei? Jos ei, niin miten voidaan valita sellainen funktio joka toimii hyvin kaikilla syötteillä?
- (g) Verkko voidaan tallettaa vieruslistana tai vierusmatriisina. Mainitse kummallekin vaihtoehdolle yksi hyvä ja yksi huono puoli.
- (h) Onko taulukko

12	14	31	16	15	40	41	19
----	----	----	----	----	----	----	----

 minimikeko? Jos ei, niin mikä avain rikkoo kekoehdon?
- (i) On annettu suunnattu sykli-ton verkko, jonka jokaisen kaaren paino on 42. Mainitse nimeltä mahdollisimman tehokas algoritmi, jonka avulla voidaan laskea lyhin polku annetusta lähtösolmusta kaikkiin muihin verkon solmuihin.
- (j) Väite: suurella täyttöasteella avoin hajautus toimii paremmin kuin ylivuotolistojen käyttö. Totta vai ei? Jos ei, niin miksi ei?
-