

TKT20001 Tietorakenteet ja algoritmit (kevät 2018)

Kurssikoe 2 (9.5.2018)

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 pseudokoodia, 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.

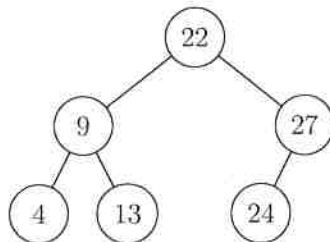
Vastaa kaikkien kysymysten kaikkiin kohtiin. Kokeen maksimipistemäärä on 22.

1. [6 pistettä]

- (a) Mitä tarkoittaa *vahvasti yhtenäinen komponentti*?
- (b) Millainen on *vuoverkko* ja siinä *vu*? Mitä ovat vuoverkon *maksimivuo* ja *minimileikkaus*?

Kummassakin kohdassa esitä täsmälliset määritelmät ja selvennä niitä esimerkillä.

2. [4 pistettä] Allaolevaan AVL-puuhun lisätään avaimet 15, 17, 2 ja 10, tässä järjestyksessä. Esitä puu kunkin lisäyksen jälkeen ja selitä myös mahdolliset tasapainotusoperaatiot.

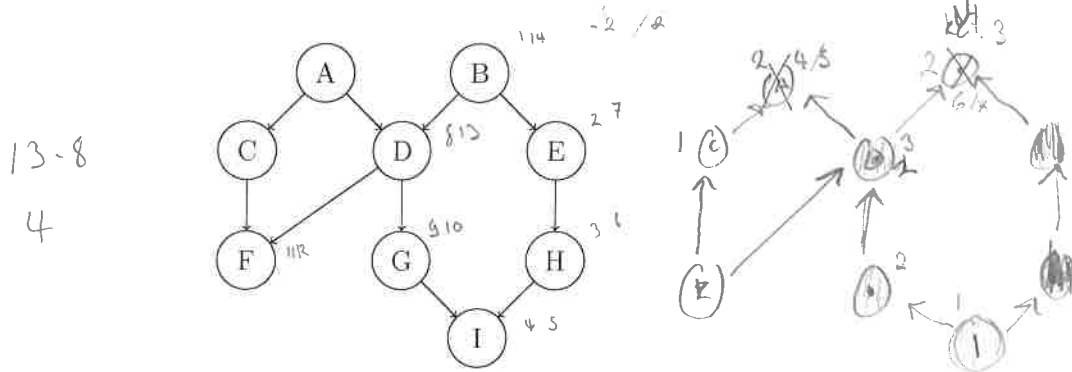


Käännä!

3. [6 pistettä] Syötteenä on annettu suunnattu sykkitön verkko $G = (V, E)$. Tehtävänä on löytää verkosta solmu, josta on polku mahdollisimman moneen muuhun solmuun. Jos useasta eri solmusta on sama suurin määrä saavutettavia solmuja, mikä tahansa niistä kelpaa ratkaisuksi.

Esitä yksityiskohtaisena pseudokoodina tehokas ratkaisualgoritmi ongelmalle. Syöte oletetaan esitettyksi vieruslistoina, ja täysien pisteiden saamiseksi algoritmin aikavaativuuden tulee olla $\mathcal{O}(|V| + |E|)$.

Esimerkki: Allaolevalla verkolla algoritmin pitäisi palauttaa solmu B, koska se on ainoa solmu, josta voi saavuttaa 6 muuta solmua (solmut D, E, F, G, H ja I).



4. [6 pistettä] Mitä tarkoittaa verkon pienin virittävä puu? Millaisten verkkojen tapauksessa sitä yleensä tarkastellaan? Kuvaile jokin tehokas algoritmi verkon pienimmän virittävän puun löytämiseksi.

Kuvauksessasi selitä algoritmin perusajatus esimerkin avulla. Selitä myös tarvittavat aputietorakenteet: mitkä ne ovat, mihin niitä tarvitaan ja millaisia operaatioita niihin kohdistuu. Sinun ei tarvitse esittää tietorakenteiden toteutusta, itse algoritmin pseudokoodia, aikavaativuusanalyysia tms. Samoin esimerkissä ei tarvitse esittää algoritmin toimintaa alusta loppuun jollain verkolla, vaan sen verran, että ajatus tulee selväksi.

1 trans
DFS aikojen kanssa

A → F + 1
A → F + 1

A →	C, D + 2	A →	/
B →	D, E + 2	B →	/
C →	F + 1	C	
D →	F, G + 2	D	
E →	H + 1	E	
F →	/	F	
G →	I + 1	G	
H →	I + 1	H	
I →	/	I	

2