
3 tehtävää, yhteensä 40p.

1. [Yhteensä 16p] Vastaa *lyhyesti* seuraaviin kysymyksiin:

- Mitä tarkoitetaan algoritmin iso- O aika- ja tilavaativuudella. (4p)
- Mainitse nimeltä jokin listatietorakenne (jos siitä on useita eri muunnelmia, kerro täsmällisesti mitä tarkoitat), ja kerro perus- joukko-operaatioiden aikavaativuudet (6 kappaletta) valitsemassasi tietorakenteessa. (4p)
- Miksi haluamme hakupuiden olevan tasapainoisia? Mitkä ehdot hakupuun on täytettävä jotta se olisi punamusta hakupuu? (4p)
- Piirrä jokin sellainen binäärihakupuu (ei välttämättä tasapainoinen, eikä välttämättä punamusta) jossa on vähintään viisi solmua, ja juurella on sekä vasen että oikea lapsi. Piirrä solmuihin myös avaimet, jotka ovat kokonaislukuja. Tee puun juurisolmulle vasen kierto (ja piirrä myös lopputulos). (4p)

2. [Yhteensä 12p] Kokonaislukujoukot on toteutettu linkitettyinä listoina.

Halutaan tehdä algoritmi **Erotus**(L_1, L_2), joka saa parametrina listat L_1 ja L_2 , ja palauttaa listan joka sisältää listan L_1 alkioista sellaiset, jotka eivät esiinny listassa L_2 . Listat L_1 ja L_2 eivät saa muuttua algoritmin suorituksen aikana.

- Selosta ratkaisuidea sanallisesti, kun listat *eivät* ole järjestettyjä. (4p)
- Kirjoita ehdottamasi algoritmin pseudokoodi. (4p)
- Mikä on algoritmisi aikavaativuus? Tilavaativuus? (2p)
- Jos listat olisivat olleet järjestettyjä, voisiko ongelman ratkaista tehokkaammin? Miten? (Lyhyt sanallinen selostus riittää.) Kuinka tehokkaasti? (2p)

3. [Yhteensä 12p] Halutaan liittää binäärihakupuun jokaiseen solmuun x kokonaisluku *size*[x], joka on solmun x aitojen jälkeläisten lukumäärä (siis poislukien solmu x itse, eli lehtisolmulla on 0 jälkeläistä, ja juurella on $n - 1$ jälkeläistä, kun puussa on yhteensä n solmua).

- Miten *size*-arvot saadaan lasketuksi, kun annetuissa hakupuussa nämä kentät ovat aluksi tyhjiä.
 - Selosta sanallisesti miten ratkaisisit ongelman. (3p)
 - Kirjoita ratkaisusi pseudokoodi edellisen kohdan mukaisesti. (3p)
 - Analysoi algoritmisi aikavaativuus. (2p)
- Oletetaan, että edellisen kohdan mukaiset *size*-arvot ovat nyt solmuissa. Halutaan kehittää *teho*-kas algoritmi, joka annetuilla avaimella k palauttaa puussa olevien avainta k pienempien avainten lukumäärän. (Voit olettaa, että mikään avain ei esiinny puussa useammin kuin kerran.)
Selosta lyhyesti ratkaisun periaate (3p), ja arvioi sen aikavaativuutta (1p).