

Tietorakenteet ja algoritmit (Syksy 2012)

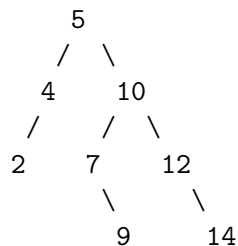
Harjoitus 7 (2.11.2012)

1. Olkoon annettu kaksi yhteen suuntaan linkitettyä listaa, jotka molemmat sisältävät kokonaislukuja suuruusjärjestyksessä (pienin ensin). Kirjoita algoritmi *lomita*, joka muodostaa uuden yhteen suuntaan linkitetyn listan, joka sisältää kummankin alkuperäisen listan sisältämät luvut suuruusjärjestyksessä. Jos luku sisältyy kumpaankin alkuperäiseen listaan, saa se sisältyä lomitettuun listaan vain kerran (alkuperäisissä listoissa kukin luku esiintyy korkeintaan kerran).

Käytä osoittimia, ei taulukkoesitystä. Et voi olettaa, että listallasi on mitään valmiita operaatioita (insert, delete, ...) tai että listan oheen olisi talletettu ylimääräistä tietoa (kuten listan pituutta n). Huom: muista ottaa tyhjä lista huomioon.

2. Binäärihakupuut

- (a) Selitä lyhyesti miten AVL-puuhun lisätään alkioita. Lisää allaolevaan puuhun ensin alkio 8 ja poista näin saadusta puusta juurisolmu. Näytä puun lopputilanne kunkin lisäyksen ja poiston tultua suoritetuksi sekä välitilanteet, joissa tehdään jokin tasapainotusoperaatio.
- (b) Tee algoritmi, joka laskee kuinka leveä binääripuu on. Toisin sanoen kuinka monta alkioita on suurimmillaan yhdellä puun tasolla. Esimerkiksi alla olevan puun leveys on 3, sillä sen toiseksi alimmalla tasolla ovat alkiot 2,7,12 ja se on puun levein kohta. Mikä on algoritmisi aika- ja tilavaativuus?



3. Otetaan aluksi tyhjä hajautustaulu, jonka koko m on 11 ja hajautusfunktio $h(k) = k \bmod m$. Taulukkoon lisätään avaimet 10, 22, 31, 4, 15, 28, 17 ja 88 tässä järjestyksessä.
 - (a) Piirrä ja selitä, miten taulukko täyttyy, kun yhteentörmäykset käsitellään yli-voitoketjuilla.
 - (b) Piirrä ja selitä miten taulukko täyttyy avoimessa hajautuksessa, kun yhteentörmäykset käsitellään lineaarisella kokeilulla ja $h' = h$.
 - (c) Kuten edellä, mutta yhteentörmäykset käsitellään neliöisellä kokeilulla, missä $h' = h$, $c_1 = 1$ ja $c_2 = 3$.
 - (d) Kuten edellä, mutta yhteentörmäykset käsitellään kaksoishajautuksella $h'(k) = k \bmod m$ ja $h''(k) = 1 + (k \bmod (m - 1))$

4. Tavoitteena on luoda hajastustaulu yliopiston opiskelijoista avaimena opiskelijanumero. Helsingin yliopistossa on noin 35000 opiskelijaa. Yhteentörmäykset käsitellään ylivuotoketjuilla, joiden pituuden oletusarvon toivotaan olevan $2,0:n$ ja $3,0:n$ välillä. Ehdota perusteluineen sopivaa hajastustaulun kokoa m , kun käytetään jakojäännös-menettelmää.
5. (a) Ongelmana on löytää maksimikeosta pienin alkio. Keko oletetaan toteutetuksi normaaliin tapaan taulukkona. Esitä ongelmalle algoritmi, joka tutkii mahdollisimman vähäisen määrän keossa olevia avaimia.
(b) Ongelmana on tulostaa maksimikeosta kaikki annettua arvoa x suuremmat avaimet. Keon sisältöä ei saa muuttaa. Esitä ongelmalle algoritmi, jonka aikavaativuus on lineaarinen tulostettavien alkioden lukumäärän suhteen.
6. Kekoon halutaan lisätä operaatio `heap-inc-all-keys(k)`, joka kasvattaa kaikkia keossa olevia lukuja $k:n$ verran. Kuinka operaatio voidaan toteuttaa niin, että se on vakio-aikainen ja muiden keon operaatioiden aikavaativuudet säilyvät ennallaan?
7. Esitä minimikeolle algoritmi `heapify` iteratiivisessa muodossa (ts. ilman rekursiota). Huomio: Minimikeko, ei maksimikeko, kuten materiaalissa.