

811312A Tietorakenteet ja algoritmit, 2018 - 2019, Harjoitus 5

Harjoituksen aihe ovat hash-taulukot ja binääriset etsintäpuut

Tehtävä 5.1 Tallenna avaimet 10,22,31,4,15,28,17 ja 59 hash-taulukkoon, jonka koko on 11, kun tiivistefunktio on $f(k) = k(\text{mod}11)$ ja

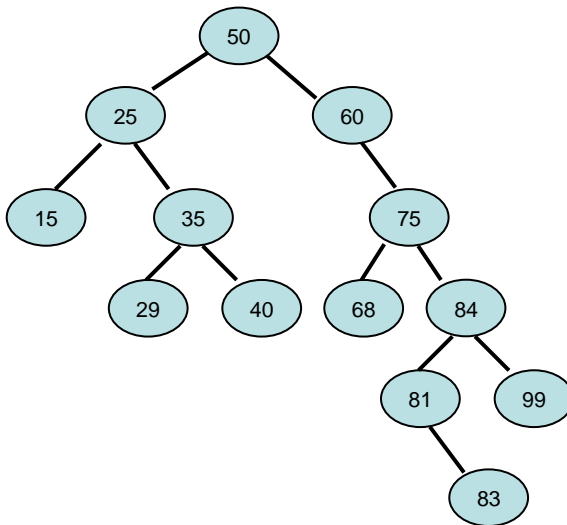
- a) törmäykset hoidetaan **ketjutuksella**;
- b) käytetään **avointa osoittamista** ja **lineaarista luotausta** törmäysten hallintaan;
- c) käytetään **avointa osoittamista** ja törmäysten hallintaan **kaksoishashausta** funktioiden f ja g avulla, missä $g(k) = 1 + (k(\text{mod}10))$.

Tehtävä 5.2 Oletetaan, että binääriseen etsintäpuuhun on tallennettu kokonaislukuja väliltä 1..1000. Puusta etsitään lukua 363. Mitkä seuraavista hakujonoista **eivät voi olla** luvun 363 etsintäjonoja?

Ohje: Tarkastele jokaisen solmun jälkeen lukuväliä, josta etsintää jatketaan.

- a) 925, 202, 911, 240, 912, 245, 363.
- b) 2, 399, 387, 219, 266, 382, 381, 278, 363.
- c) 935, 278, 347, 621, 299, 392, 358, 363.

Tehtävä 5.3 Lisää ensin alla olevasta binääriseen etsintäpuuhun avain 63 ja poista tämän jälkeen avaimet 29, 35, 75 ja 60 tässä järjestyksessä. Esitä ratkaisut kuvioilla ja perustele niitä algoritmeihin BST_POISTA ja BST_TRANSPLANT nojaten.



Ohjelmointitehtävä seuraavalla sivulla ->

Ohjelmointitehtävä

Tehtävä 5.4 Alle linkitetyssä tekstitiedostossa ”task5_4_nums.txt” on tallennettuna 25000 satunnaislukua väliltä [1,1000000]. Jokainen luku on tallennettu omalle rivilleen.

Tehtävänä on lukea luvut sellaiseen tallennusrakenteeseen, että annetusta luvusta voidaan nopeasti todeta, onko se tiedostossa olevien lukujen joukossa. Testaa ohjelmaasi luvuilla 613695, 906429, 180551, 151841, 951585 ja 569127, joista kolme ensimmäistä eivät ole tiedostossa ja kolme viimeistä ovat. Valitse toinen alla olevista vaihtoehdoista:

- a) C-kielisessä ohjelmassa käytetään tallennusrakenteena hash-taulukkoa, jonka koko on 49999. Hash-funktio saadaan käyttämällä jakomenetelmää ja törmäysten hallintaan käytetään lineaarista luotausta. Alle linkitetyssä tiedostossa ”task5_4_base.c” on ohjelma, jota voi käyttää toteutuksessa pohjana. Jos aikaa vielä jää, niin voit testata vielä ohjelmasi suorituskyyä tuottamalla 100 000 satunnaislukua väliltä [1,1000 000] ja mittaamalla näiden hakemiseen kulunut aika.
- b) Python-kielisessä ohjelmassa käytetään tallennusrakenteena binääristä hakupuuta, jonka toteutus on tiedostossa ”bst.py”. Funktio bst_insert lisää puuhun avaimen ja bst_search etsii avainta. Jos avainta ei löydy, paluuarvo on None. Voit aloittaa ohjelman laatimisen tiedostoon ”task5_4_base.py”. Testaa vielä ohjelmasi suorituskyyä tuottamalla 100 000 satunnaislukua väliltä [1,1000 000] ja mittaamalla näiden hakemiseen kulunut aika.