

MÄÄRITTELYDOKUMENTTI

# TIRAHARJOITUS

## KEKO VERTAILUT

Turunen Pirjo

Tietorakenteet ja algoritmit harjoitustyö

10.5.2013

Helsingin yliopisto, Tietojenkäsittelytieteen laitos

Kristiina Paloheimo, Mika Huttunen

## 1. Ohjelman tarkoitus ja yleiskuvaus

Ohjelma toteuttaa seuraavat keko tietorakenteet ja niiden perusoperaatiot kirjallisuudessa yleisesti esitetyillä aika- ja tilavaatimuksilla;

D-keko, binomikeko ja fibonaccikeko. Tarkoituksena on selvittää ko. keko tietorakenteiden toiminta ja selvittää tietorakenteiden erot aikavaatimusten suhteen ja paremmuus järjestys syötteen suhteen.

## 2. Toimintojen kuvaus

D-keko on täysi d-arinen puu, joka on järjestetty keoksi. Puuta täytetään tasoittain ylhäältä alhaalle vasemmalta oikealle. n-solmuisen d-keon solmut voidaan tallettaa taulukkoon  $A[0..n-1]$  seuraavasti:

Juuri talletetaan paikkaan  $A[0]$ .

Jos solmu on talletettu paikkaan i, niin solmun v lapset  $v_1, \dots, v_d$  talletetaan paikkoihin  $A[id+1], A[id+2], \dots, A[id+d]$ .

Toteutettavat operaatiot:

makeHeap	AV: $O(1)$
findMin	AV: $O(1)$
insert	AV: $O(\log n)$
deleteMin	AV: $O(\log n)$
decreaseKey	AV: $(\log n)$

Binomikeko koostuu joukosta binomi puita, jotka toteuttavat seuraavat periaatteet:

-Jokaisella solmulla on avain.

-Jokainen binomi puu toteuttaa binomikeossa min-heap ominaisuuden

-Jokaiselle ei-negatiiviselle kokonaisluvulle k, binomikeossa on enintään yksi binomipuu, jonka juuri on k.

-Solmujen lukumäärä binomipuu  $B_k$ :lle on  $2^k$ . Puun korkeus k.

-Vanhemman solmun avain on pienempi tai yhtäsuuri kuin lapsi solmun avain.

Toteutettavat operaatiot:

makeHeap	AV: $O(1)$
----------	------------

findMin	AV: $O(\log n)$
insert	AV: $O(\log n)$
deleteMin	AV: $O(\log n)$
decreaseKey	AV: $(\log n)$
merge	AV: $(\log n)$

Fibonaccikekoa kutsutaan laiskaksi tietorakenteeksi. Jokaisella solmulla on osoitin vanhempaan ja yhteen solmun lapsista. Lapsi solmut yhdistetään kaksisuuntaisella linkitetyllä listalla. Jokaisella solmulla on myös asteluku joka kertoo sen lapsien lukumäärän. Solmulla on myös mark flagi, joka kertoo onko solmu merkitty vai merkitsemätön. Eli onko solmu menettänyt lapsen sen jälkeen, kun siitä on viimeksi tehty toisen solmun lapsi.

Toteutettavat operaatiot:

makeHeap	AV: $O(1)$
findMin	AV: $O(1)$
insert	AV: $O(1)$
deleteMin	AV: $O(n)$
decreaseKey	AV: $(n)$
merge	AV: $(1)$

### 3. Testaus suunnitelma

Testimateriaali muodostetaan pienen ja harvan verkon syötteen tapauksesta suuren ja tiheän verkon syötteen tapaukseen. Vertailua tulee tehdä myös perusoperaatioiden suhteesta toisiinsa eri testimateriaalilla. Testausta varten toteutetaan Dijkstran algoritmi ja prioriteettijono toteutuksessa vertaillaan eri kekojen tehokkuutta. Testitapaukset eritellään yksityiskohtaisesti testidokumentissa.

#### 4. LÄHTEET

<http://www.cs.helsinki.fi/u/tapasane/keot.pdf>

<http://www2.it.lut.fi/kurssit/00-01/010534000/luennot/penttonen/osa8.html>

<http://trakla.cs.hut.fi/ebook/ebook-Keko.html>

<http://www.cs.helsinki.fi/courses/58131/2013/k/k/1>

”Johdatus algoritmien suunnitteluun ja analysointiin” Martti Penttonen, Otatieto, ISBN 951-672-249-0

”Introduction to ALGORITHMS” Third edition, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, ISBN 978-0-262-03384-8