

Tietorakennevertailun toteutusdokumentti

Marko Haanranta

15. kesäkuuta 2013

1 Ohjelman yleisrakenne

Tietorakenteiden harjoitustyönä toteutin neljä eri tietorakennetta ja vertailin perusoperaatioiden ajankäyttöä. Tietorakenteina toteutin binäärikeon, binomikeon, fibonaccin keon ja AVL-puun. Tietorakenteiden perusoperaatiot joita työssä vertailtiin olivat avaimen lisäys, poisto ja tietorakenteen pienimmän avaimen palautus. Kaikki keot on toteutettu minimikekoina.

1.1 Binäärikeko

Binäärikeko on toteutuksena hieman yksinkertaisempaa kuin muut toteutetut tietorakenteet. Kekoon lisättävä avain tallennetaan taulukkoon ja sen jälkeen ajetaan heapify metodi joka vie avaimen sen oikealle paikalle keossa. Jos käytetään int muuttujaa avaimen tallentamiseen niin binäärikeko on selkeästi nopeampi kuin muut toteuttamani tietorakenteet. Mutta muutin toteutustani sen verran, että nyt tallennan taulukkoon Integer olioita ja näin toteutettuna binäärikeko on hitain rakenne.

1.2 Binomikeko

Binomikeon alkiot tallennetaan node tyyppisinä oliona. Jokaisella node oliolla on seuraavat kentät: key, parent, child, sibling ja degree. Binomikeko rakentuu linkitetystä listasta, johon on tallennettu binomipuuta. Puiden solmuilla on degree ja juurilistalla ei voi olla kahta puuta joilla on sama degree, jotta keko olisi binomikeko ehdon mukainen.

1.3 Fibonaccin keko

Fibonaccin kekoon lisättävä avain tallentaa FibNode oliona. FibNode oliolla on seuraavat kentät key, parent, child, left, right, degree ja mark. Fibonaccin keko rakentuu puista. Fibonaccin keossa kaikki avaimet lisätään juurilistaan. Fibonaccin keko rakennetaan poisto operaation yhteydessä.

1.4 AVL-puu

AVL-puu on tietorakenne, jonka kaikki perusoperaatiot vievät $O(\log n)$ ajan.

2 Saavutetut aika- ja tilavaativuudet

3 Suorituskyky- ja O-analyysivertailu

4 Työn puutteet ja parannusehdotukset