

Tietorakennevertailun toteutusdokumentti

Marko Haanranta

15. kesäkuuta 2013

1 Ohjelman yleisrakenne

Tietorakenteiden harjoitustyönä toteutin neljä eri tietorakennetta ja vertailin lisäys ja poisto operaatioiden ajankäyttöä. Tietorakenteina toteutin binäärikeon, binomikeon, fibonaccin keon ja AVL-puun. 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. Muutin toteutustani sen verran, että nyt tallennan taulukkoon Integer-olioita ja näin toteutettuna binäärikeko on hitain toteuttamani tietorakenne.

1.2 Binomikeko

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

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 puusta. Fibonaccin keossa kaikki avaimet lisätään juurisolmulistaan ja varsinainen keko rakennetaan vasta poisto operaation yhteydessä.

1.4 AVL-puu

AVL-puu on tasapainoinen binäärihakupuu eli jokaisella solmulla on kaksi lasta ja vanhempi paitsi juurisolmulla. Puun solmun vasemmassa alipuussa on vain solmua pienemmän avaimen omaavia solmuja. Puun solmun oikeassa alipuussa on vain puun solmua suuremman avaimen omaavia solmuja. AVL-puu pidetään tasapainossa suorittamalla kierto operaatioita, kun puuhun lisätään tai siitä poistetaan solmuja. Sen kaikki perusoperaatiot vievät $O(\log n)$ ajan.

2 Saavutetut aika- ja tilavaativuudet

Kaikki toteuttamani tietorakenteet saavuttivat $O(\log n)$ aikavaativuuden. Fibonacci keon insert ei syystä tai toisesta toiminut odotetun $O(1)$ aikavaativuuden mukaisesti, mistä enemmän testidokumentissa.

3 Suorituskyky- ja O-analyysivertailu

4 Työn puutteet ja parannusehdotukset