

Määrittely dokumentti:

Työssä vertaillaan kolmen puurakenteen, listan ja hajautustaulun nopeutta eri tilanteissa. Puurakenteet ovat binääri-hakupuu, AVL -puu ja punamusta-puu. Ohjelma vertailee kuinka tietorakenteet suoriutuvat joukkoihin liittyvistä operaatioista.

Ohjelma toteutetaan Java-ohjelmointikielellä. Ohjelman käyttöliittymä tulee olemaan tekstipohjainen. Käyttäjän voi lisätä tietorakenteisiin haluamansa määrän solmuja ja ajaa erilaisia operaatioita. Ohjelma tulostaa operaatioihin kuluneen kokonaisajan ja lisäksi joissain operaatioissa ohjelma tulostaa väliaikoja.

Binäärihakupuun, AVL -puun ja punamustan-puun lisäksi ohjelma tarvitsee listaa ja hajautusrakennetta (kuten HashMap). Vertailussa voidaan käyttää myös listaa ja hajautusrakennetta.

Hajautusrakenteessa lisäämisen ja poiston tulee toimia ajassa $O(1)$. Etsimisen tulisi toimia vähintään ajassa $O(n)$, mutta käyttäjän antaessa kunnollisen hajautus funktion tulisi etsinnän toimia keskimäärin ajassa $O(1)$.

Listassa tulisi lisäämisen taulukon alkuun tai loppuun toimia ajassa $O(1)$. Lisättäessä johonkin muuhun taulukon kohtaan aikavaativuus on $O(n)$, sillä muita alkioita joudutaan siirtämään tilan tekemiseksi. Etsimisen ja sisältämisen tulisi toimia ajassa $O(n)$. Hakemisen indeksin perusteella tulisi toimia ajassa $O(1)$ listan alusta ja lopusta ja muualta ajassa $O(n)$.

Puurakenteiden lisäys, poisto ja etsintä operaatioiden tulisi toimia keskimäärin ajassa $O(\log n)$. Binäärihakupuussa pahimmantapauksen aikavaativuuden tulee olla $O(n)$, kun se AVL-puussa ja punamustassa-puussa on $O(\log n)$. Puut tarjoavat myös operaatiot min, max, succ ja pred, joiden tulee toimia myös ajassa $O(\log n)$.

Kaikki tietorakenteet ovat geneerisiä. Ne tarjoavat kaikki myös joitain perus operaatioita, kuten clear ja size muiden operaatioiden lisäksi. Näiden operaatioiden aikavaativuudet (kuten myös muiden operaatioiden) löytyvät tietorakenteen dokumentaatiosta.