

Boonus Ülesanne

Analüüsi ja võrdle erinevaid binaarpuude tasakaalustamise algoritme (näiteks AVL, punase-musta, Splay puud, B-tree) teoreetiliselt. Selgita, kuidas need algoritmid aitavad optimeerida andmestruktuuride jõudlust erinevates rakendustes.

AVL puu

AVL puu on range tasakaalustatud binaarne otsingupuu, kus iga sõlme tasakaalustatusefaktor on piiratud mingi arvu vahemikuga.

Otsingu, sisestuse ja kustutamise operatsioonide kompleksus on $O(\log n)$, mis teeb sellest suurepärase valiku rakendustesse, kus otsingute sagedus on suur ja muudatuste arv on väiksem.

Sobib hästi, kui rakenduses on peamiselt otsingud ja harvemini muudatused, aga on vähem paindlikum kui punane-must puu, mis tähendab, et mõnedes olukordades võib olla AVL puu kasutamine rohkem ressursi- ja ajakulukas

Punane-must puu

Punase-musta puu on tasakaalustatud binaarne otsingupuu, millel on vähem ranged tasakaalureeglid võrreldes AVL-puuga. See põhineb värvireeglitel (punased ja mustad sõlmed).

Jõudlus otsingu, sisestamise ja kustutamise puhul $O(\log n)$. Punase-musta puu vähem rangemad tasakaalureeglid võivad, aga viia vähematele tasakaalustamistoimingutele kui AVL-puus.

Sobib hästi, kui rakenduses on peamiselt otsingud ja harvemini muudatused, aga on paindlikum kui AVL puu, mis tähendab, et mõnedes olukordades võib olla punane-must puu parem valik.

Splay puu

Splay puu on struktuur, mis toob hiljuti kasutatud sõlmed puu tippu. See eelistab sagedamini kasutatud elemente, mis võib parandada otsingute jõudlust.

Splay puu operatsioonide keerukus varieerub sõltuvalt viimati kasutatud sõlmedest. Kuigi üksiku operatsiooni keerukus võib olla kõrge (halvimal juhul $O(n)$), võib suure kasutussageduse korral otsingute efektiivsus paraneda.

Splay puu võib olla eelistatud, kui rakenduses on vajalik sagedasti kasutatud elementide kiiret otsimist. Splay puu negatiivne osa on see, et kui teha pidevalt palju erinevaid otsinguid, siis operatsioonide keerukus suureneb.

B puu

B puu on tasakaalustatud puu, mis on optimeeritud suurte andmemahtude jaoks. See jagab andmestruktuuri mitmeks blokiks või alampuuks, mis võimaldab tõhusalt töödelda suuri andmeid.

B puu sobib hästi disk-põhiste rakenduste jaoks, kus puu jagamine alampuudeks vähendab I/O-operatsioonide arvu. Operatsioonide keerukus sõltub alampuude astmest ja andmemahtudest.

B-puu sobib hästi suurte andmemahtude ja disk-põhiste rakenduste jaoks, kus I/O-operatsioonide optimeerimine on oluline, näiteks andmebaasid.