Algoritmid ja andmestruktuurid

Kodutöö 5

Tiina Mandel

1. Esitatud eraldi failina Algo\_KT5\_UL1
2. Kuhja (Heap) Struktuuri Teoreetiline Analüüs

Kuhjad on täielike binaarpuude sort, seega igal vanemsõlmel on kõige rohkem kaks last ja kõik tasemed välja arvatud viimane (alumine) on täidetud. Kuhi korraldab elemendid sortimiseks massiivis binaarseks kuhjaks, korduvalt suurimat elementi liigutades. Min-Heap on binaarpuu tüüp, milles iga sõlm on väiksem või võrdne oma lastega. Max-Heap puhul on iga sõlm suurem või võrdne oma vanemast. Kuhja ajakeerukus on O(log n) nii parimal, keskmisel kui halvimal juhul. Kuhjad on efektiivsed suurte andmekogumite jaoks ja efektiivsem võrreldes võrdlusel põhinevate sortidega nagu Bubble Sort ja Insertion Sort. Kuhjad on efektiivsed ja populaarsed tööriistad nii andmete sorteerimiseks kui ka prioriteetjärjekordade haldamiseks, kuna nende omadused ja toimingute efektiivsus sobivad hästi nende ülesannete jaoks.

1. Binaarne otsingupuu (Binary Search Tree, BST) Kahendpuud koosnevad sõlmedest, igal sõlmel on kuni kaks sõlmelast, vasak laps on alati väiksem kui vanem ja parempoolne laps on alati suurem kui vanem. Tasakaalustamata puude puhul võib ajakeerukus olla kuni O(n( näiteks kui puu on nihkes. Seda saaks optimeerida puud tasakaalustades.
2. Punase-Musta Puu Punase-musta puu on tasakaalus kahendpuu, kus sõlmed on jagatud mustadeks ja punasteks, kusjuures punane ja must paiknevad puu visualiseerimisel vaheldumisi. Peale iga sõlme lisamist toimub puu tasakaalustamine, mis aitab säilitada efektiivsuse põhioperatsioonide (sisestamine, otsimine ja kustutamine) jaoks. Juursõlm on alati must. Punase sõlme lastesõlmed on alati mustad (puul ei ole kahte järjestikust punast sõlme). Iga tee juursõlmest ükskõik millise leheni sisaldab sama arvu musti sõlmi (seda nimetatakse mustaks kõrguseks, black height). Lehtsõlmed on nullsõlmed, mida käsitletakse mustadena.  
   Võrdlus: Tavalises binaarses otsingupuus ei ole tasakaalustamise mehhanisme. Punase-musta puu on alati tõhusam või vähemalt sama tõhus kui tavaline binaarne otsingupuu, sest see väldib halvima juhtumi 𝑂(𝑛), aga tavaline binaarne otsingupuu võib siiski olla veidi kiirem väga väikese sõlmede arvuga, kuna punase-musta puu tasakaalustamise kulud võivad väikestes struktuurides lisada pisut overhead’i.  
   Tasakaalustamise mehhanismid koos värvireeglitega aitavad hoida puu alati tasakaalus või peaaegu tasakaalus. Need mehhanismid võimaldavad hoida punase-musta puu efektiivsust, kuna puu kõrgus ei ületa log 𝑛, mis on võtmetähtsusega kõigi põhioperatsioonide kiireks teostamiseks. Punase-musta puud kasutatakse sageli praktikas andmestruktuuridena, näiteks Java ja TreeMap, kus efektiivsus ja tasakaalustamine on olulised.
3. AVL puu on isebalanseeruv binaarne otsingupuu, mis jälgib balansifaktorit ehk vasaku ja parema alampuu kõrguste erinevus on maksimaalselt 1 ja puu pöörab sõlmi, et säilitada tasakaalu. See oli üks esimesi isebalnseeruvaid puid ja sai nime autoritr järgi (Adelson-Velsky Landis). tasakaalustamine tagab, et iga sõlme vasaku ja parema alampuu kõrguste erinevus ei ületa 1. See piirang tagab, et puu jääb alati tasakaalustatuks, ja võimaldab kiireid otsimise, lisamise ja kustutamise operatsioone. AVL puu tasakaalustamine nõuab rohkem pöördeid, kui punase-musta puu tasakaalustamine, aga punase-musta puu otsimine võib olla aeglasem kui puu on vähem tasakaalus kui AVL puu.

AVL puud sobivad paremini, kui otsimine toimub sagedamini kui lisamine/kustutamine või kui andmekogumid on staatilisemad või väiksemad.

Punase-musta puud sobivad paremini, kui andmekogumid on dünaamilised ja nõuavad sagedast lisamist ja kustutamist või reaalajasüsteemides, kus operatsioonide maksimaalne aeg peab olema prognoositav.