

Ülesanne 3: Binaarse Otsingupuu Teoreetiline Analüüs

Binaarne otsingupuu on *andmestruktuur*, mis võimaldab kiiret otsimist, elementide lisamist ja eemaldamist.

Igal sõlmel on võti ning sõlme vasak ja parem alampuu. kuna igal sammul saame pool võimalikest asukohtadest välistada, liikudes kas vasakule või paremale, olenevalt sellest, kas otsitav võti on suurem või väiksem hetkel vaadeldava sõlme võtmest, võimaldab see meil andmeid efektiivselt otsida.

Võti on andmeelement, mida hoitakse ja mille alusel toimub otsinguprotsess. Iga sõlme võti on unikaalne ja võimaldab võrrelda sõlmi omavahel, et otsustada, millises suunas (vasakule või paremale) otsimist või elementide lisamist jätkata.

Alampuu viide on viide või osuti sõlmele, mis moodustab antud sõlme vasaku või parema alampuu.

Tasakaalustamata puud

Binaarse otsingupuu efektiivsus sõltub puu struktuurist – ideaalis peaks iga sõlme vasak ja parem alampuu olema umbes sama kõrge, mis võimaldab operatsioone teostada kiiresti.

Kui aga puu on tasakaalustamata, võivad mõned oksad olla palju pikemad kui teised ja puu efektiivsus väheneb. Selle tulemusel võib puu sarnaneda rohkem järjendiga ja operatsioonide sooritamiseks kulub rohkem aega.

Et puud tasakaalustada, kasutatakse erinevaid algoritme ja tehnikaid.

Punase-musta puud on isetasakaalustuvad puud, kus iga sõlm on kas punane või must. Selle puu algoritmid tagavad, et puu jääb tasakaalustatuks läbi kindlate reeglite, mis põhinevad sõlmede värvusel ja suhtel.

AVL puud on isetasakaalustuvad puud, mis hoiavad kontrolli all iga sõlme alampuude kõrguste vahet. Kui mõne sõlme alampuude kõrguste vahe ületab lubatud piiri, tehakse puus pöörded ehk liigutatakse sõlmi ümber nii, et puu kõrgused jälle tasakaalu saavutaksid.