

## Ülesanne 2: Kuhja Struktuuri Teoreetiline Analüüs

### **MIN** kuhi

Täielik binaarpuu, kus iga sõlme väärtus on väiksem või võrdne oma laste väärtustega.

See tähendab, et puu kõige ülemises sõlmes asub kõige väiksema väärtusega element kõigist puus olevatest elementidest.

Efektiivne struktuur minimaalse elemendi kiireks leidmiseks.

### **MAX** kuhi

Erinevalt MIN kuhjast on iga sõlme väärtus suurem või võrdne oma laste väärtustega.

Juurelement, mis asub puu tipus, on suurima väärtusega võrreldes ülejäänud puus olevate elementidega, mistõttu on max-kuhja kasutamine kõige suurema elemendi kiireks leidmiseks tõhus.

### Aja- ja ruumikomplekssus

#### Sisestamine **$O(\log n)$**

*Tuleneb vajadusest elementi vajadusel läbi puu kõrguste ülespoole viia, et säilitada kuhja struktuuri reegleid.*

#### Elemendi eemaldamine **$O(\log n)$**

*Tingitud asjaolust, et pärast elemendi väljavõtmist tuleb puu uuesti korraldada, et see vastaks taas kuhja omadustele.*

#### Minimaalse/maksimaalse elemendi leidmine **$O(1)$**

*Kuna need elemendid asuvad puu juuretasandil, on nende leidmine kiire, kuludes vaid  $O(1)$  aega.*

#### Ruumikomplekssus **$O(n)$**

*Kõigi kuhja elementide hoidmiseks on vajalik lineaarne hulk mälu. Siin tähistab  $n$  elementide koguarvu kuhjas.*

## Struktuuride sobivus andmete sorteerimiseks ja haldamiseks

Kuhjad sobivad hästi andmete sorteerimiseks, kasutades kuhja sortimise algoritmi (heap sort), mis võimaldab andmeid sorteerida  $O(n \log n)$  ajakomplekssusega.

Kuhja sortimine on eriti efektiivne suurte andmekogumite puhul, sest erinevalt  $O(n^2)$  ajakomplekssusega sortimisalgoritmidest, nagu selection sort või bubble sort, ei suurene kuhja sortimise ajakulu andmete hulga kasvades eksponentsiaalselt.

Prioriteetjärjekord on andmestruktuur, mis võimaldab elemente hoida ja hallata vastavalt nende prioriteedile. Kuhjad, olgu need siis min-kuhjad või max-kuhjad, on prioriteetjärjekordade implementeerimisel äärmiselt populaarsed.

Kuna kuhja eesmärk on hoida kõrgeima või madalaima prioriteediga elementi juuresel, on prioriteedi alusel ligipääs lihtsustatud. See tähendab, et kõige olulisem element on alati kohe kättesaadav, mis on kriitilise tähtsusega paljudes rakendustes, näiteks operatsioonisüsteemide protsessiplaneerijates, simuleerimistes ja andmevoogude töötlemises.

Prioriteetjärjekordade kasutamisel kuhjade abil on oluline märkida, et kuigi kuhjad pakuvad efektiivset toimivust, võib nende kasutamine nõuda täiendavaid kaalutlusi, nagu kuhja tüübi valik (min- või max-kuhja) sõltuvalt rakenduse nõuetest ning võimalikud mõjud süsteemi üldisele toimivusele suurte andmekogumite puhul.