

Ülesanne 2

Kuhja (Heap) struktuuri teoreetiline analüüs:

Kirjelda min-kuhja ja max-kuhja struktuuride teoreetilisi omadusi.

V: Min-kuhja struktuurilised omadused on näiteks, et selle juur on alati väiksem element kogu kuhjas. Iga sõlm on väiksem kui tema lapsed, ehk iga alamkuhi on samuti min-kuhi. Uus element lisatakse kuhja ja siis muudetakse selle asukohta, et säilitada min-kuhja omadused. Kui kõige väiksem element eemaldatakse, vahetatakse alles jäänud sõlmede kohad, et taastada min-kuhja omadused.

Max-kuhja struktuurilised omadused on näiteks see, et juur alati kõige suurem element kogu kuhjas. Iga sõlm kuhjas on suurem kui tema lapsed, ehk iga alamkuhi on samuti max-kuhja. Uus element lisatakse kuhja, siis muudetakse selle asukohta, et säilitada max-kuhja omadused. Kui kõige suurem element eemaldatakse, siis taastatakse max-kuhja omadused, samamoodi nagu min-kuhjas.

Analüüsi nende struktuuride aja- ja ruumi kompleksust.

V:

	Aja kompleksus	Ruumi kompleksus
Sisestamine	Min-kuhja ja Max-kuhja puhul on sisestamise aja kompleksus $O(\log n)$, kus n on kuhja suurus.	Sisestamisel ei ole täiendavat mälukulu, kuna uus element lisatakse olemasoleva kuhja lõppu.
Eemaldamine	Min-kuhja ja Max-kuhja puhul on eemaldamise aja kompleksus	Eemaldamisel ei ole täiendavat mälukulu, kuna eemaldatakse juur ja kuhja struktuur

	$O(\log n)$, kus n on kuhja suurus.	taastatakse.
Laste leidmine	Min-kuhja ja Max-kuhja puhul on parema ja vasaku lapse leidmise ja komplekssus konstantne $O(1)$.	Neid väärtusi hoitakse otseselt kuhja struktuuris, nii et ei ole täiendavat mälukulu.
Ehitamine kuhjast	Min-kuhja ja Max-kuhja puhul on kuhja ehitamise aja komplekssus $O(n)$, kus n on elementide arv.	Täiendav mälukulu on minimaalne, sest olemasolev kuhja struktuur muudetakse jooksvalt.
Sorteerimine	Min-kuhja ja Max-kuhja puhul on kuhja sorteerimise aja komplekssus $O(n \log n)$.	Sorteerimise käigus ei ole täiendavat mälukulu, kuna kuhja struktuur muudetakse kohapeal.

Aruta, kuidas kuhja struktuurid sobivad andmete sorteerimiseks ja prioriteet järjekordade haldamiseks.

V: Binaarsed kuhjad on kasulikud sorteerimisel, kuna nad võimaldavad efektiivset sorteerimist, kasutades korduvuse omadust. Binaarsed kuhjad on kasulikud ka prioriteetide järjekorra haldamisel, kus soovime kiiresti ligi pääseda kõrgema või madalama prioriteediga elemendile.