Kodutöö 4

Paisktabelid, Räsitabelid (***Hash map***)

Tähtaeg: 3. Detsember 2023 23:59

Seda koduülesannet võib täita kas üksi või paarilisega. Kui valite töötamise paaris, veenduge, et mõlema osaleja nimed oleksid dokumentatsioonis mainitud. Kogu kood, joonised ja dokumentatsioon tuleks üles laadida enda GitHubi hoidlasse. Hoidla nimi võib olla näiteks algoritmid2023.
Ülesandeid tuleb lahendada paberil või implementeerida kood, vabalt valitud keeles.
Kui ülesanne on lõpetatud, esitage Moodle'is hoidla link.

Ülesanne 1: Räsimine

1. Kirjuta selgitus räsimise (*hashing*) kontseptsioonist - põhiidee ja eesmärk.
2. Kirjelda hea räsifunktsiooni omadusi ja selgita, miks need on olulised.
3. Selgita kokkupõrgete lahendamise tehnikaid, eriti eraldi aheldamist (separate chaining) ja avatud aadressimist (open addressing).

Ülesanne 2: Indeksi Kaardistamise (*Index Mapping)* Rakendamine ja Analüüs

1. Rakenda ***Index Mapping*** algoritmi (triviaalne räsimine) kasutades massiive või järjendeid/liste.
2. Analüüsi oma rakenduse aja- ja ruumikomplekssust.
3. Aruta, kuidas indeksi kaardistamist massiividega saab rakendada reaalses maailmas.

Ülesanne 3: *Separate Chaining* Kokkupõrgete Lahendamiseks

1. Rakenda *separate chaningut* kasutades *linked-liste*.
2. Võrdle *separate chaningu* efektiivsust *open addressing* meetodiga ajalise ja ruumilise komplekssuse mõttes.
3. Aruta *separate chaning* kasutamise plusse ja miinuseid räsitabelites.

Ülesanne 4: *Open Addressing* Tehnikate Uurimine

1. Kirjuta lühike ülevaade avatud aadressimise meetodist kokkupõrgete lahendamisel räsimises.
2. Võrdle (teooria) kolme tehnikat: lineaarne otsing (linear probing), ruuduline otsing (quadratic probing) ja topelträsimine (*double hashing*).
3. Aruta, millistes olukordades iga tehnika oleks kõige efektiivsem.

Ülesanne 5: Topelt Räsimise (*Double hashing)* Rakendamine

1. Rakenda topelt räsimise algoritm ja aruta, kuidas see aitab ületada ühekordse räsimise piiranguid.
2. Analüüsi oma rakenduse aja- ja ruumikomplekssust.
3. Paku välja stsenaarium, kus topelt räsimine oleks eriti efektiivne.

Boonusülesanne: Koormustegur ja *Rehashing* (*Double hashing vs Rehashing)*

1. Selgita, mis on räsitabeli koormustegur ja miks see on oluline.
2. Rakenda lihtsat ***Rehashingu*** protsessi ja aruta, kuidas see aitab säilitada efektiivset räsitabelit.
3. Analüüsi ***Rehashingu*** mõju räsitabeli jõudlusele.