

Boonusülesanne

Koormustegur ja Rehashing (Double hashing vs Rehashing):

1. Selgita, mis on räsi tabeli koormustegur ja miks see on oluline.

V: Räsi tabeli koormustegur on suhe, mis näitab, kui palju tabeli kohad on hõivatud võrreldes koguarvu kohadega. Koormustegur on oluline, kuna valib sobivad meetodeid kokkupõrgete lahendamiseks, nagu näiteks suurendada tabeli suurust või valida teine avatud aadressimise meetod. Mida madalam on koormustegur, seda efektiivsemalt kasutatakse mälu. Kõrge koormustegur võib viia räsi tabeli aeglasema jõudluseni. Valem koormusteguri arvutamiseks:

Koormustegur = Hõivatud kohad/Koguarv kohad

2. Rakenda lihtsat Rehashingu protsessi ja aruta, kuidas see aitab säilitada efektiivset räsi tabelit.

V: Rehashing on protsess, kus räsi tabelit laiendatakse või muudetakse, et säilitada selle efektiivsus, olukorras, kus tabelisse lisatakse uusi elemente. Rehashingu eesmärk on vältida kokkupõrkeid ja tagada, et räsitabel jääks ikka sama heaks. Lihtne rehashingu protsess käib nii, et on algne räsi tabel, kus igale võtmel on räsi, mille põhjal määratakse tabeli indeks. Tabelisse lisatakse uus element võtmega, siis arvutatakse tema räsi. Kui räsi põhjal määratud indeksil on juba olemas väärtus, siis toimub kokkupõrge, mis tuleb lahendada. Rehashingu kasutamiseks on kaks varianti, kas muudame räsi funktsiooni või kohandame tabeli suurust. Mina kasutasin tabeli suuruse muutmist, selleks tuleb tabelit suurendada või vähendada, et vähendada kokkupõrkeid. Pärast rehashingut tuleb olemasolevad elemendid uuesti paigutada vastavalt uutele indeksitele, mis on arvutatud uue tabeli suurusega. Rehashing on mahukas protsess, kuid see ongi loodud selleks, et säilitada räsitabeli efektiivsuses

3. Analüüsi Rehashingu mõju räsitabeli jõudlusele.

V: Rehashing vähendab kokkupõrgete arvu, mis on oluline efektiivse räsi tabeli jaoks. Kui räsi tabelis on vähem kokkupõrkeid, siis on ka vähem aega, mis kulub kokkupõrgete lahendamisele. Rehashingu käigus peavad kõik elemendid liikuma uutele indeksitele vastavalt muutunud räsi funktsioonile või tabeli suurusele. See

on raske, eriti suurte andmemahtude korral. Kui uus räsi funktsioon ei ole piisavalt efektiivne või tekitab palju kokkupõrkeid, võib rehashing muutuda keeruliseks ja vähendada tabeli jõudlust.