

# Adatbázisok

## 3. gyakorlat – Fizikai szervezés

- Egy 1000 rekordból álló állományt ritka index szervezéssel tárolunk. A rekordhossz 850 bájt, egy blokk kapacitása (a fejrészt nem számítva) 4000 bájt. A kulcs 50 bájtos, egy mutatóhoz 18 bájt kell.
  - Hány rekord fér el egy blokkban?
  - Hány blokkot foglal el az indexstruktúra és mennyit a teljes állomány?
  - Melyik szinten, melyik blokkokban és blokkok között követeljük meg a rendezettséget?
  - Mennyi ideig tart legfeljebb egy rekord tartalmának kiolvasása, ha feltételezzük, hogy az index struktúra már benne van az operatív tárban? (egy blokkművelet ideje 5 ms)
  - Mennyi ideig tart legfeljebb egy rekord tartalmának kiolvasása, ha az index struktúra nem fér el az operatív tárban? (egy blokkművelet ideje 5 ms)
- Egy 7 vödörrel rendelkező hash tábla leképező függvénye  $h(k) = k \bmod B$ . A következő rekordok érkeznek, amelyeket szeretnénk eltárolni: 56, 91, 27, 19, 36, 52, 79.
  - Feltételezve, hogy egy rekord egy blokknyi méretű, mennyi az átlagos rekordelérési idő?
  - Tetszőleges másik hash függvényeket választva mennyi az elméletileg elérhető legjobb és legrosszabb véletlenszerű rekordelérési idő ugyanakkora elemszámnál?
- Vödörös hash szervezéssel tárolunk egy állományt, amelyben a rekordok száma 15000. Egy rekord hossza 120 bájt, egy blokkba 4000 bájt fér el, egy kulcs hossza 25 bájt, egy mutatóé 8 bájt. A szervezést 10 vödörrel oldjuk meg. (Feltételezhetjük, hogy a hash függvény egyenletesen osztja el a kulcsokat.)
  - Mekkora az átlagos vödörméret?
  - Mekkora lemezterület szükséges a teljes struktúra tárolásához (valódi méret, illetve felhasznált tárterület)?
  - Mennyi az átlagos rekordelérési idő, ha a blokkelérési idő 5 ms? (A keresés során a vödör-katalógust a memóriában tároljuk.)
  - Mekkora legyen a vödörök minimális száma, ha a keresés során átlagosan 5 blokkelérési idő alatt akarjuk megtalálni a keresett rekordot?
- Egy állományt kétféle szervezéssel tudunk tárolni: sűrű index, majd erre épített egyszintes ritka index vagy pedig hash algoritmussal. Az állományon néha intervallumkeresést is meg kell valósítani. Melyik szervezési módszert válasszuk? Adjon értelmes alsó becslést a szükséges blokkok számára az alábbi feltételek mellett:
  - az állomány 3 000 000 rekordból áll
  - egy rekord hossza 300 bájt
  - egy blokk mérete 4000 bájt
  - a kulcshossz 45 bájt
  - egy mutató hossza 5 bájt
- Egy 10 000 000 rekordból álló állományt szeretnénk B\*-fa szervezéssel tárolni. A rekordhossz 850 bájt, egy blokk kapacitása (a fejrészt nem számítva) 4000 bájt. A kulcs 50 bájtos, egy mutató tárolásához 18 bájt kell. Legalább hány blokkra van szükség? Mennyi az átlagos rekordelérési idő, ha a memóriában egy blokk fér el? (Egy blokk elérésének ideje 5 ms.)
- Gondolkodtató kérdések
  - Elképzelhető-e sűrű index felhasználása hash szervezés esetén?
  - Mik a hash szervezés előnyei, hátrányai a B\*-fával szemben?
  - Milyen adatszerkezetet tudsz elképzelni egy térképszoftver adatbázisának?
  - Milyen sorrendben kell beszúrunk a rekordokat egy B\*-fába ahhoz, hogy a legtöbb helyet pazaroljuk?
  - Legfeljebb hány ritka index építhető közvetlenül egy heap szervezésű állományra?
  - Milyen plusz feladataink vannak beszúrásakor, illetve törléskor, ha sűrű indexek segítségével több B\*-fát építünk az adatbázisunkra?
  - Lehet értelme egy kulcs szerint indexelt (B\*) adatbázis esetén is használni sűrű indexet? Mit nyerünk vele, és mennyit? Mítől függ, hogy mennyit nyerünk?
  - Miért építünk B\*-fát, ha ritka indexszel is logaritmikus a keresés?
  - Milyen nehézségeink adódnak, ha a töredékblokkokat is fel szeretnénk használni a merevlemezen?
  - Miben különbözik egy kicsi és egy nagy blokkméretű lemezen tárolt adatbázis?
  - Miért nem beszéltünk arról, hogy blokkon belül hogyan tároljuk az adatokat?
  - Helyezd el a következő kifejezéseket a táblázatban: „blokknyi” „egyetlen”

	hány rekordot jelöl egy bejegyzése?	hány rekordot jelöl ki egy mutató érték?
Sűrű index		
Ritka index		

- Hogyan változnának meg az adatbázisok, ha a jövőben a fizikai memóriában (az operatív tárban) helyezkedne el az adatbázisunk?