

- Vödrös hash-t használva akarunk számokat tárolni, a $h(x) = x \pmod{11}$ hash függvényt használva.
 - Szűrjük be a kezdetben üres, 11 méretű hash táblába a 2, 5, 12, 1, 3, 88, 23, 43, 10, 34 elemeket!
 - Hogyan zajlanak a feltöltött táblában a következő keresések: keres(1), keres(20), keres(45)?
 - Hogyan zajlik az a) pont táblájában az 23 törlése?
- Nyitott címzéssel hash-elünk egy kezdetben üres $M = 11$ méretű táblába a $h(x) = x \pmod{11}$ hash-függvénnyel. Mi lesz a tábla állapota, ha a 4, 5, 14, 15, 16, 26, 3 kulcsokat ebben a sorrendben beszűrjük és az ütközések feloldására (a) lineáris próbát használunk? (b) kvadratikus maradék próbát használunk? (c) kettős hash-elést használunk, amikor $h'(x) = (7x \pmod{10}) + 1$ a második hash-függvény? Hány ütközés történt az egyes esetekben?
- Egy B_{20} -fának (huszadrendű B-fának) 10^9 levele van. Mekkora a fa szintjeinek minimális, illetve maximális száma?

- Előfordulhat-e nyitott címzéses hash-elés esetén, hogy az $n > 3$ méretű táblában csak 3 elem van, de a keresés lépésszáma n ?
- Jó választás-e $M = 7$ méretű táblánál a $h(x) = x^2 \pmod{7}$ hash-függvény?
- Az alábbi 11 méretű hash táblába nyílt címzéssel, lineáris próbával szűrünk be néhány egész számot majd egyet közülük kitöröltünk, így az alábbi állapotot kaptuk (* jelöli a törölt cellát, a kitöltetlen cellák mindvégig üresek voltak). A használt hash függvény a $h(x) = x \pmod{11}$ függvény volt.

0	1	2	3	4	5	6	7	8	9	10
11	4	*	3	15		6	18			22

Ebben a táblázatban egyetlen olyan szám van, amire igaz, hogy ha ezt a számot kitöröljük, majd újra beszűrjük, akkor a szám egy másik helyre kerül vissza. Melyik ez a szám? Válaszának indoklására mutassa be, hogy hogyan zajlik ennek a számnak a törlése és beszúrása.

- Egy B_{11} fába az x elem beszúrásakor 2 csúcsvágást kellett végezni. Legalább hány elemet tárolt a beszúrás előtt az adatstruktúra?
 - Tervezzünk olyan adatszerkezetet, ami egy rendezett halmaz elemeinek tárolására szolgál. A megvalósítandó műveletek: *Felépít*(n): n elemből felépíti a struktúrát; *Mintör*, *Maxtör*: a min. illetve max. elem törlése; *Beszúr*(x): az x elemet a struktúrába illeszti. Az egyes műveletek uniform költsége ne legyen több, mint *Felépít*: $O(n)$; *Mintör*, *Maxtör*, *Beszúr*: $O(\log n)$, ahol n a tárolt elemek száma.
-
- A $T[0 : M]$ táblában $2n$ elemet ($n < M/3$) helyeztünk el valamilyen hash-függvény segítségével, amikor azt tapasztaltuk, hogy az elemek mindegyike az első $3n$ hely egyikére került. Ha nem volt közben törlés és a végén a táblában minden $3i$ indexű hely üres maradt ($0 \leq i < n$), akkor legfeljebb hány ütközés lehetett, ha (a) lineáris próbát használtunk? (b) kvadratikus maradék próbát használtunk?
 - A 0 és 70 közötti páratlan számokat szűrjük be valamilyen sorrendben, nyílt címzéssel, lineáris próbával egy kezdetben üres 47 méretű hash táblába, a $h(x) = x \pmod{47}$ hash függvényt használva. Mutassa meg, hogy mindegy, hogy a számok milyen sorrendben érkeznek, az ütközések száma minden sorrend esetén ugyanannyi.
 - Tervezzon adatstruktúrát a következő feltételekkel. Természetes számokat kell tárolni, egy szám többször is szerepelhet. A szükséges műveletek:
 BESZÚR(i): i egy újabb példányát tároljuk
 TÖRÖL(i): i egy példányát töröljük
 MINDTÖRÖL(i): i összes példányát töröljük
 DARAB(i): visszaadja, hogy hány példány van i -ből
 ELEM(K): megmondja, a nagyság szerinti rendezésben a K -adik elem értékét.
 Az adatstruktúra legyen olyan, hogy ha m -féle elemet tárolunk, akkor mindegyik művelet lépésgénye $O(\log m)$.
 (Például ha a tárolt elemek 1,1,3,3,3,8, akkor DARAB(1)=2, ELEM(4)=3 és $m = 3$.)