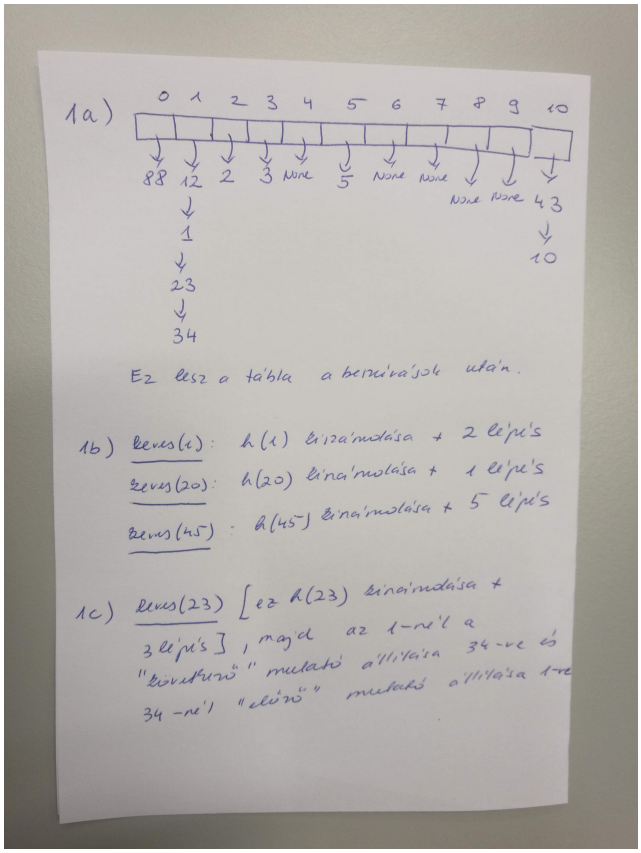
Algoritmusok és gráfok
HETEDIK GYAKORLAT, 2019. október 25.
Megoldások néhány feladathoz

1. Az ütközések feloldására egy, a nyílt címzéstől különböző stratégia a vödrös hash. Ennél a megoldásnál mindegyik cella egy listát tartalmaz, a listát azon elemek alkotják, amikre a hash függvény a cella indexét adja vissza. Beszúráskor a a K kulcsot a $h(K)$ indexű cella listájának a végére illesztjük, a K kulcs keresésekor a $h(K)$ indexű cella listáját járjuk végig, törléskor pedig a megtalált elemet a listáknál tanult módon (a mutatók állításával) töröljük a listából.

   Vödrös hash-t használva akarunk számokat tárolni, a $h(k) = k$ maradéka 11-gyel osztva hash függvényt használva.
   (a) Szúrjuk be a kezdetben üres, 11 méretű hash táblába a 2, 5, 12, 1, 3, 88, 23, 43, 10, 34 elemeket!
   (b) Hány lépésől állnak az a) pontban feltöltött táblában a következő keresések: keres(1), keres(20), keres(45)?
   (c) Hogyan zajlik az a) pont táblájában az 23 törlése?

   Megoldás:



2. A $h(k) = k$ maradéka 9-cel osztva hash függvényt használva vödrös hasheléssel akarunk számokat tárolni. Minden n-re adjon olyan n hosszú számsorozatot, hogy az n szám beszúrása után egy következő keresés rossz esetben n lépésig tartson.

   Megoldás

   Szúrjuk be a $0, 9, 18, 27, \ldots, 9(n-1)$ számokat. Ezek mind a 0-s indexű cellába mennek, ahol így egy n hosszú lista fog kialakulni, ha ebben keressük például a $9n$ értéket (ami nincs a táblában), akkor az egész listát be kell járnunk.

3. **(Vizsga 2018)** Az alábbi 11 méretű hash táblába nyílt címzéssel, lineáris próbával szúrtunk 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 $h(K) = \mathrm{K}$ maradéka 11-gyel osztva 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 beszúrjuk, 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.

Megoldás

A 11, 3, 15, 6 és 18 mindegyikére igaz, hogy a hash függvény által adott h(K) indexű cellában van, ha ezeket kitöröljük, akkor az ismételt beszúrásnál a h(K) indexű cella törölt lesz, ide fogjuk őket visszarakni. A maradék két lehetőség tehát a 22 és a 4. A 22-t kitörölve a 10-es cella törölt lesz, de a 22 beszúrásakor a 0-s cellában kezdünk, ami tele van, innen pedig a 10-be megyünk és ide szúrjuk be az elemet.

A 4-es elemnél viszont a törlés után a beszúrást a 4-es cellában kezdjük, ami tele van, utána a 3-as cellát nézzük, ez is tele van, utána a 2-be megyünk, ami törölt és ide rakjuk be a 4-t, azaz máshova kerül, mint ahonnan töröltük.

4. **(Mintazh 2018)** A 0 és 70 közötti páratlan számokat szúrjuk be valamilyen sorrendben, nyílt címzéssel, lineáris próbával egy kezdetben üres 47 méretű hash táblába, a $h(K) = K$ maradéka 47-tel osztva hash függvényt használva. Mutassa meg, hogy mindegy, hogy a számok milyen sorrendben érkeznek, az ütközések száma minden sorrend esetén ugyanannyi.

   Megoldás

   Az $1, 3, \ldots, 45$ (tehát a 47-nél kisebb páratlan számok) mind a helyükre kerülnek, köztük nincs ütközés, bármilyen sorrendben is jönnek. A 47-től 70-ig levő páratlan számok (azaz $47, 49, \ldots, 69$) a $0, 2, 4, \ldots, 22$ cellákba kerülnek, ezért ők egymással sem ütköznek, bármilyen sorrendben is jönnek és a 47-nél kisebb számokkal sem ütköznek, mert azok páratlan indexű cellákba akarnak kerülni.
   Tehát teljesen mindegy, hogy mi a sorrend, mert soha nem lesz ütközés, nincs két olyan szám, aminek a hash értéke ugyanaz lenne.

5. Egy m méretű hash-táblában már van néhány elem. Adjon $O(m)$ lépésszámú algoritmust, amely meghatározza, hogy egy újabb elem lineáris próbával történő beszúrásakor maximum hány ütközés történhet.

   Megoldás

   Lineáris próba esetén a beszúrás lépésszáma a leghosszabb, a táblában levő egybefüggő foglalt rész hosszával egyezik meg, ahol ez az egybefüggő rész úgy is előállhat, hogy a tábla elején néhány cella és a tábla végén néhány cella alkotja (a keresés során visszakanyarodtunk a 0-s cellából az $(m-1)$-es cellába). Arra kell tehát algoritmus, hogy ezt meghatározzuk, ennek pszeudokódja a következő lehet:

   Algo

```
max_eddig: = 0     // az eddigi legnagyobb hossz
számláló := 0         // az aktuális blokk hossza eddig

ciklus i = 0-tól (m-1)-ig:  // a nem visszakanyarodó eset
    ha az i indexu cella foglalt:
        számláló +=1
    egyébként:                                  // amikor vége van a blokknak
        ha számláló > max_eddig:
            max_eddig : = számláló
            számláló: = 0
    elágazás vége
ciklus vége

ha számáló == m: // a tábla tele van
    return m
```

```
ha számláló > 0 és számláló < m:   // a végén van néhány tele cella
           ciklus i =0-tól (m-1)-ig:
               ha az i indexu cella foglalt:    // tovább számolunk
                   számláló += 1:
               egyébként:
                   visszakanyarodó_hossz: = számláló
                   kiszállunk a ciklusból
               elágazás vége
           ciklus vége

return max (max_eddig, visszakanyarodó_hossz)
```

Az algoritmus magyarázata a következő: először megkeressük a leghosszabb egybefüggő, nem visszakanyarodó blokk hosszát, utána pedig, ha ez m volt, akkor a tábla tele van, ha pedig nem m volt, de a tábla végén a számláló értéke nem nulla (vagyis az utolsó cella nem üres), akkor újra elindulunk az elejéről és addig növeljük még a számlálót, amíg vannak elemek a tábla elején.

Jóság:
Az első ciklus megtalálja a leghosszabb egybefüggő blokk hosszát, ami nem a tábla végén van vagy nem visszakanyarodik. Az ezutáni elágazások kezelik a tele táblát és azt, amikor a tömb végére esik a leghosszabb blokk vagy visszakanyarodik.

Lépésszám: Az első ciklus magja legfeljebb m-szer fut le, minden lefutás konstans sok lépés, ez $O(m)$. Az elágazásban levő második ciklus is legfeljebb m-szer fut le, minden lefutás konstans lépés, ez is $O(m)$, ezen felül pedig csak konstans lépés van, vagyis az egész is $O(m)$.