Algoritmusok és gráfok - Zárthelyi
2019. november 21.
Megoldások

A válaszokat indokolni kell. Hivatkozni az előadáson tanultakra lehet.

1. Az alábbi pszeudokód inputja egy $n \geq 2$ szám, a pszeudokódban lépésnek egy darab * kiírása és az értékadás számít. Mutassa meg, hogy a pszeudokód által megadott algoritmus lépésszáma $O(n^2)$.

```
ciklus j = 0-tól (n-1)-ig:
    k  := 0
    ciklus amíg k < j:
        kiírunk egy  darab *-ot
        k := k + 1
    ciklus vége
ciklus vége
```

**Megoldás**

A belső ciklus (az amíg-ciklus) magja $j$-szer fut le, mert minden lefutáskor nő eggyel a $k$ értéke, ami 0-tól indul és $j-1$-re fut le utoljára. Mivel $j \leq n$, ezért a belső ciklus magja legfeljebb $n$-szer fut le.
A belső ciklus magja 2 lépés, azaz konstans, így a belső ciklus egésze $O(n)$-es.

A külső ciklus ciklusmagja egy utasításból, majd az előbb látott belső ciklusból áll, vagyis a külső ciklus ciklusmagja $1 + O(n)$, azaz $O(n)$ lépés.
A kód maga ez a külső ciklus, aminek $O(n)$-es magja $n$-szer fut le, így a teljes lépésszám $O(n^2)$.

2. A $12, x, 2, 9, 10, 5$ tömböt kiválasztásos rendezéssel és buborékrendezéssel is rendeztük. Adja meg $x$ összes lehetséges értékét, ha tudjuk, hogy $x$ a tömbben máshol elő nem forduló egész szám és a kiválasztásos rendezés során volt $2, 5, 12, 9, 10, x$ átmeneti állapot, a buborékrendezés során pedig $2, x, 9, 5, 10, 12$ állapot.

**Megoldás**

A kiválasztásos rendezés először a tömb legkisebb elemét keresi meg és egy cserével a tömb elejére mozgatja, így kerül a 2 az első helyre és a $2, x, 12, 9, 10, 5$ tömböt kapjuk (eközben kiderült, hogy $x > 2$, mert különben $x$ került volna az első helyre és nem a 2). Ezután a maradék elemek közüli legkisebbet cserével a tömb második cellájába visszük, így kerül oda az 5, a kapott tömb $2, 5, 12, 9, 10, x$. Ekkor kiderül, hogy $x > 5$, mert különben az x került volna a második helyre és nem az 5. Több infót ebből a rendezésből nem tudunk meg, mert elértük a közbülső állapotot, tehát eddig annyit tudunk, hogy $x > 5$.

A buborékrendezés futása során először a 12 a tömb végére kerül, eközben a 12-t meg kellett cserélnünk minden őt követő elemmel, vagyis az első kör után a tömb állapota biztosan $x, 2, 9, 10, 5, 12$. Ezután a 10 a tömb hátulról második helyére kerül, eközben a 2-t és az $x$-et megcseréltük (hiszen $x > 5$ a kiválasztásos miatt), de $x$ már nem cserélődött fel 9-cel, vagyis $x < 9$. Mivel itt is elértük a közbülső állapotot több infó nem derül ki, csak az, hogy $x < 9$

A lehetséges értékek tehát $x = 6, 7, 8$.

3. Egy kezdetben üres, 11 méretű hash táblába nyílt címzéssel, lineáris próbával szúrtunk be kilenc egész számot. A használt hash függvény értéke a $K$ kulcs esetén $K$ maradéka 11-gyel osztva. A beszúrások után egyetlen számot kitöröltünk a táblából, de a törölt cellát véletlenül nem állítottuk töröltre, üresnek látszik. Az alábbi táblát kaptuk, amiben tehát a törlés helye nem látszik. Melyik cellában történt a törlés?

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| | 13 | 2 | 28 | 15 | | 7 | 18 | | 21 | 10 |

**Megoldás**

A törlés az 5-ös indexű cellában történt, mert a 28-as számra a hash függvény értéke 6, azaz a 28 a 6-os indexű cellába került volna és csak úgy kerülhetett a 3-as indexűbe, ha a beszúráskor mind a 6-os, mind az 5-ös és a 4-es indexű cella is foglalt volt.

4. Egy öt csúcsú AVL-fát preorder bejárással bejárva a csúcsokat $9, 5, x, 8, 12$ sorrendben látogatjuk meg. Adja meg, hogy hogyan néz ki ez a fa és adja meg $x$ összes lehetséges értékét, ha tudjuk, hogy $x$ egy olyan pozitív egész szám, ami máshol nem fordul elő a fában.

**Megoldás**

A preorder bejárás a fa gyökerét látogatja meg először, vagyis a gyökér a 9. A bal fába kerülnek a 9-nél kisebb értékek a bináris keresőfa tulajdonság miatt, vagyis 5 és 8 biztosan, de ha 8 balra kerül, akkor $x$ is, mert a preorder bejárásnál a teljes bal fát előbb látjuk, mint a jobb fát. Vagyis a jobb fában csak a 12 van, a balban pedig $5, x, 8$.

A jobb részfa magassága tehát 1 és mivel ez egy AVL-fa, ezért a bal fa magassága legfeljebb 2 lehet, vagyis az $5, x, 8$ csúcsok úgy helyezkednek el, hogy egyikük a gyökér, a másik kettő meg ennek a jobb és bal gyereke.

Mivel a preorder bejárás az 5-öt látja meg az $5, x, 8$ közül elsőnek, ezért ez van középen és akkor $x$ ennek a bal, 8 pedig a jobb gyereke.

A fából a bináris keresőfa tulajdonság miatt látjuk, hogy $x$ olyan elem, ami 5-nél kisebb pozitív egész vagyis $x = 1, 2, 3, 4$ lehet.

5. Adott két $n$ elemű tömb, mindegyikben $n$ darab különböző egész számot tárolunk. Adjon $O(n \log n)$ lépésszámú algoritmust a két tömb legkisebb közös elemének megtalálására, vagy ha nincs ilyen elem, akkor ennek jelzésére.

**Egy lehetséges megoldás**

Algo:

(a) Rendezzük mindkét tömböt összefésüléses rendezéssel.

(b) Menjünk végig az első tömbön és a tömb minden elemére csináljunk egy bináris keresést a második tömbben (nézzük meg, hogy az első tömb aktuális eleme benne van-e a második tömbben). Amelyik értéknél ez először előfordul, az a legkisebb közös elem, ha egyszer sem fordul elő, akkor nincs közö elemük.

Jóság: A rendezés után a bináris keresés használható, hogy az első tömb elemeiről eldöntsük, hogy benne vannak-e a másodikban, mert a második tömb rendezett. Mivel minden elemet megnézek az első tömbből, ezért biztos észreveszem, ha van egyezés és mivel az első tömb elemeit nagyság szerinti sorban nézzük, ezért a legkisebb közöset találjuk meg először.

Lépésszám: A két rendezés $O(n \log n)$ és $O(n \log n)$, ez együtt is $O(n \log n)$, ezután pedig $n$ darab bináris keresést csinálok egy $n$ méretű tömbben, azaz $n$-szer futtatok egy $O(\log n)$-es eljárást, ez is $O(n \log n)$, együtt is $O(n \log n)$.

6. Egy szomszédossági mátrixával adott $n$ csúcsú, irányított $G$ gráfban kettő csúcs, $u_1$ és $u_2$ kékre van színezve, a többi csúcs színtelen. Adott továbbá két kijelölt színtelen csúcs, $s$ és $t$ és $s$-ből szeretnénk $t$-be eljutni a gráf éleit használva úgy, hogy közben mindkettő kék csúcson áthaladunk. Adjon $O(n^2)$ lépésszámú algoritmust, ami meghatározza, hogy ehhez legkevesebb hány élen kell áthaladnunk (vagy ha az eljutás nem lehetséges, akkor ez derüljön ki).

**Megoldás**

Ötlet: Két lehetőség van arra, hogy $s$-ből $t$-be eljussunk a két kék csúcsot érintve: $s$-ből elmegyek a $u_1$-be, onnan $u_2$-be, onnan $t$-be vagy $s$-ből $u_2$-be megyek, onnan $u_1$-be, onnan $t$-be (vagyis az a két lehetőség, hogy a két kék csúcsot milyen sorrendben látogatom meg). Az itt felmerülő $s - u_1, s - u_2, u_1 - u_2, u_1 - t, u_2 - u_1, u_2 - t$ távolságokat (a legkevesebb élből álló utak hosszát, azaz élszámát) kell tehát meghatározni és így megkapom a két lehetőséghez tartozó hosszakat, ebből a kisebb lesz a keresett érték.

Algo: Szélességi bejárást futtatok $s$-ből (így meglesz a távolság $s$-ből $u_1$-be és $u_2$-be), $u_1$-ből (így meglesz a távolság $u_1$-ből $u_2$-be és $t$-be) és $u_2$-ből (így meglesz a távolság $u_2$-ből $u_1$-be és $t$-be). Ebből a hat értékből kiszámolom a fenti két lehetőséghez tartozó legrövidebb séta hosszát és a kisebbik lesz a keresett szám.

Jóság: Csak ez a két lehetőség van, tehát elég ezt a hat rész-távolságot meghatározni, ezeket pedig a szélességi bejárás megadja (tanultuk).

Lépésszám: Egy bejárás $O(n^2)$, ezt kell lefuttatni háromszor, az $3 \cdot O(n^2)$, vagyis $O(n^2)$, utána már konstans sok munka a két összeg meghatározása.