

Név: Neptun kód:

Algoritmusok és adatszerkezetek I. vizsga, 2016.01.19.

1. Egy bináris fa mikor *szigorúan bináris*? Mikor *teljes*? Mikor *majdnem teljes*? Ez utóbbi mikor *balra tömörített*, és mikor *kupac*? Szemléltessük az alábbi vektorban ábrázolt kupacra a 7, majd az **eredmény kupacra** a 8 beszúrásának műveletét! $\langle 7; 5; 7; 5; 3; 4; 6; 2; 1; 2 \rangle$. Szemléltessük az **eredeti kupacra** a maxKivesz eljárás végrehajtását! Mindkét beszúrás és a maxKivesz végrehajtása után is rajzoljuk újra a fát! (20p)

2. A bináris keresőfákat ismertnek feltételezve, mondjuk ki az AVL fa meghatározásához szükséges definíciókat! Rajzoljuk le a következő AVL fát a csúcsok egyensúlyaival együtt! $\{ [(\{1\} \ 2 \ \{3\}) \ 5 \ (6)] \ 7 \ [\ 8 \ (9)] \}$ – Szemléltessük a 4 beszúrását és a 7 törlését, **mindkét esetben az eredeti fára!** Jelöljük, ha ki kell egyensúlyozni, a kiegyensúlyozás helyét, és a kiegyensúlyozás után is rajzoljuk újra fát! A rajzokon jelöljük a csúcsok egyensúlyait is, a szokásos módon! Rajzoljuk le a hat általános kiegyensúlyozási séma közül azokat, amiket alkalmaztunk! (20p)

3. Egy láncolt lista mikor *egyszerű*? Az L pointer egy nemüres, egyszerű lista első elemére mutat. Írjuk meg a **szétoszt**(L, L_1, L_2) eljárást, ami L első elemét a helyén hagyja, míg – a megfelelő listaelemek átfűzésével – a többi, az elsőnél kisebb-egyenlő kulcsú elemből az L_1 , a nagyobb kulcsú elemekből pedig az L_2 listát építi fel. Az L lista végül egyelemű lesz. Az L_1 és L_2 egyszerű listákban az elemek sorrendje tetszőleges, akár az eredeti sorrendjük fordítottja is lehet (a program így a legegyszerűbb). $T(n) \in \Theta(n)$, ahol n az L lista eredeti hossza. A program az L listát csak egyszer dolgozza fel! (20p)

4. A bináris fa fogalmát ismertnek feltételezve, mit értünk a bináris keresőfa alatt? Adott a **gykTöröl**(f), $T(h(f)) \in O(h(f))$ hatékonyságú eljárás, ami az f nemüres bináris keresőfából törli a gyökércsúcsban található kulcsot a megfelelő csúccsal együtt. (Ezt nem kell megírni.) Ennek segítségével írjuk meg a **töröl**(t, k, s) – ciklust nem tartalmazó – szintén $T(h(t)) \in O(h(t))$ hatékonyságú rekurzív eljárást, ami az s , logikai típusú paraméterben visszaadja, hogy talált-e a t bináris keresőfában k kulcsú csúcsot, és ha talált, a kulcsot és a megfelelő csúcsot törli a fából. A fát kizárólag a **gykTöröl**(f) eljárás segítségével szabad módosítani. A fa csúcsai Csúcs típusúak, azaz szülő pointert nem tartalmaznak. (20p)

5. Adjuk meg a **kupacrendezés**($A[1..n]$) és segédeljárásai struktogramjait! Igaz-e, hogy $MT(n) \in \Theta(n \lg n)$? Miért? Igaz-e, hogy $mT(n) \in \Theta(n)$? Miért? (20p)