

Algoritmusok és adatszerkezetek I. - 2. minta ZH

Az eljárásokat és függvényeket a tanultak szerinti fejléccel rendelkező struktogrammal adjuk meg! A fejléc tartalmazza az algoritmus nevét, a paraméterek nevét, típusát, ha szükséges jelöljük a cím szerinti paraméterátvételt, függvények esetében adjuk meg a visszatérési típust is.

1. feladat: Rendezze a **tanult** gyorsrendezéssel (**quicksort**) a

$\langle 9, 7, 3, 5, 17, 13, 6, 8, 4, 12 \rangle$

egész számokat tartalmazó tömböt! Feltesszük, hogy a particionálás minden esetben az aktuális résztömb **első** elemét választja tengelynek (pivot elemnek).

Adja meg sorban a ***partition(A, p, r)*** segédfüggvény hívásai által kiszámolt (egynél hosszabb) résztömböket, az elemeik felsorolásával. Jelölje meg, hogy melyik elem volt a pivot (tengely) elem és mi volt az aktuális eljárás hívás! (Az első hívás: *partition(A, l, l0)*.) (10 p)

2. feladat:

a) A tanult algoritmussal építsen bináris keresőfát a következő adatokból: (6 p)

90, 160, 72, 83, 130, 185, 155, 122, 54, 188, 144!

b) Adja meg a kapott fa ***preorder*** és ***postorder*** bejárását! (2+2 p)

3. feladat: Adott egy egész számokat tartalmazó **t** gyökerű láncoltan ábrázolt bináris fa. Írjunk olyan **rekurzív** függvényt, amely megadja, hogy van-e **páros** szám a **nem levél** elemekben! (10 p)

4. feladat: Adott egy **t** gyökerű láncoltan ábrázolt bináris fa. Készítse el a **maxLevél(t)** iteratív függvényt, amely a fa **szintfolytonos bejárásán** alapul, és megadja, a **fa levelei** közül a **maximális kulcsú levél címét**. Ha üres a fa, akkor NULL-t kell visszaadnia. (14 p)

5. feladat: Egy maximum prioritásos sort egy tömbben ábrázolt kupac segítségével valósítunk meg.

A kupac aktuális tartalma a következő:

92, 78, 88, 71, 60, 25, 48, 17, 39, 28

a) Rajzoljuk fel a kiindulási bináris fát! (2 pont)

b) Hajtsuk végre az ***add(100)***, majd az ***add(90)*** műveleteket! Adjuk meg a kialakuló új kupacokat gráfosan és tömbösen is! (2+2 pont)

c) Hajtsuk végre az ***eredeti*** kupacra egymás után az ***x:=remMax()*** és az ***y:=remMax()*** utasításokat! Adjuk meg x, y értékét és a kialakuló kupacokat gráfosan és tömbösen is! (3+3 pont)

d) Építsen **minimum kupacot** az eredeti maximum kupacból, úgy ahogy a kupacrendezésnél építjük a kezdeti kupacot! Jelölje a kupacépítés sorrendjét, majd adja meg a kialakult kupacot tömbösen is! (4 pont)