

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: Egy nem teljes bináris fa bejárása során a következő sorozatokat kaptuk:

Inorder: 9; 6; 19; 10; 15; 14; 5; 7; 17; 12; 1; 3

Postorder: 9; 19; 15; 10; 6; 5; 14; 17; 1; 3; 12; 7

a) Rekonstruálja a binárisfát és rajzolja le a fát! (4 p)

b) Mi lesz a bináris fa **preorder** bejárása? (2 p)

2. feladat: Rendezzük növekedően az alábbi elemeket kupacrendezéssel!

90, 54, 22, 71, 85, 93, 48, 16, 39, 8

a) Rajzolja fel a kiindulási bináris fát! (1 p)

b) Jelölje a kupacépítés sorrendjét, majd adja meg a kezdeti kupacot gráfosan és tömbösen is! (3 p)

c) Ezután mutassa meg két elem helyrevitelét! (4 p)

3. feladat: Adott egy egész számokat tartalmazó **t** gyökerű láncoltan ábrázolt bináris fa. Írunk 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. (10 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!

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

c) Hajtsuk végre az **eredeti** kupakra az **x:=remMax()** utasítást! Adjuk meg x értékét és a kialakuló kupacot gráfosan! (2 pont)

Név:

Neptun kód:

6. feladat: (10 pont)

Adott egy 11 méretű nyílt címzéses hasító tábla. A kulcsütközések kezeléséhez **kettős hasítás** próbásorozatot használunk. A tábla az alábbi táblázat első sorában megadott kulcsokat már tartalmazza, a többi rés üres (kulcsa: E). A táblázat soraiban szemléltesse a megadott műveletek végrehajtását: adja meg a művelet elvégzéséhez használt hasító függvény értékeit, az aktuális próbásorozatot, és adja meg a művelet elvégzése után a tábla réseinek tartalmát.

$m=11$, $h_1(k) = k \bmod 11$ (elsődleges hasító függvény), $h_2(k) = (k \bmod 10) + 1$ (másodlagos hasító függvény),
 i – beszúrás, d – törlés, s - keresés