



Polsemestrálny test teoretická časť



Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

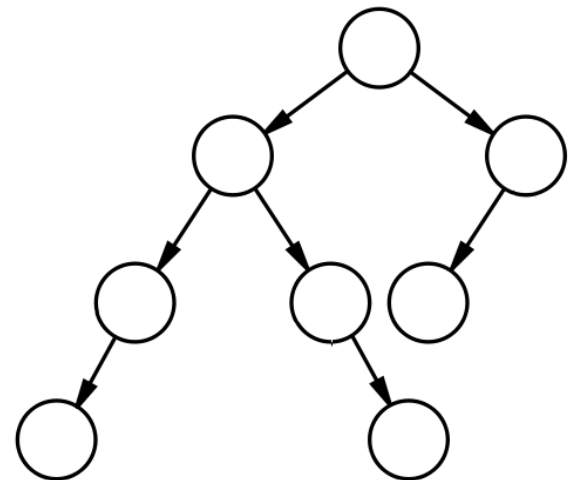
Píšte prosím čitateľne!

Hodnotenie, vyplní opravujúci:

Meno a priezvisko:	Skupina PAZ:	
--------------------	--------------	--

1/0.5	2/2	3/2	4/2	5/1	6/2	7/6	8/2	9/3	10/2	Σ/22.5

1. (0.5b) Definujte, kedy nazveme nejaký binárny strom binárnym vyhľadávacím stromom:



2. (2b) Hodnoty 2, 3, 5, 8, 13, 21, 34, 55 vložte do binárneho stromu tak, aby strom spĺňal vlastnosti binárneho vyhľadávacieho stromu.

3. (2b) Nech *z* je referencia na objekt realizujúci zásobník celých čísel. Uvažujme kód programu taký, ako je uvedený vpravo. Zistite, čo sa nachádzalo v zásobníku pred cyklom, keď sme dostali takýto výstup:
5, 7, 4, 3, 5, 6, 4, 6, 5, 5,
Pôvodný obsah zásobníka uveďte v poradí od najspodnejšieho prvku až po vrch zásobníka.

```
Stack<Integer> z = ...;
while (!z.isEmpty()) {
    int i = z.pop();
    if ((i % 5 > 2) || (z.isEmpty())) {
        System.out.print(i + ", ");
    } else {
        int j = z.pop();
        System.out.print(j + ", ");
        System.out.print(i + ", ");
    }
}
```

4. (2b) Akú hodnotu vráti metóda *dž* pri volaní *dž(pp)* ak pole *pp* bolo vytvorené takto:
`int[] pp={-5, -8, -2, -7, 9, 6, 10};`

```
public static int dž(int[] p) {
    int z = 0;
    int k = p.length - 1;
    if (p[k] < 0)
        return k+1;

    while (z < k) {
        int c = (z + k) / 2;
        if (p[c] < 0)
            z = c+1;
        else
            k = c;
    }

    return z;
}
```

5. (1b) Ak zoberieme počet prvkov poľa *p* ako parameter *n*, ktorá z týchto tried funkcií najlepšie vystihuje časovú zložitost' metódy *dž*?

$O(n)$, $O(1)$, $O(n \cdot \log(n))$, $O(\log(n))$, $O(n^2)$

8. (0.5b za správnu odpoveď, -0.25b za nesprávnu odpoveď) Uvažujme 7-prvkovú haldu. Rozhodnite o pravdivosti tvrdení.

a) $A > B > C$

Áno Nie

b) Každý z uzlov A, B, C musí byť aspoň 10

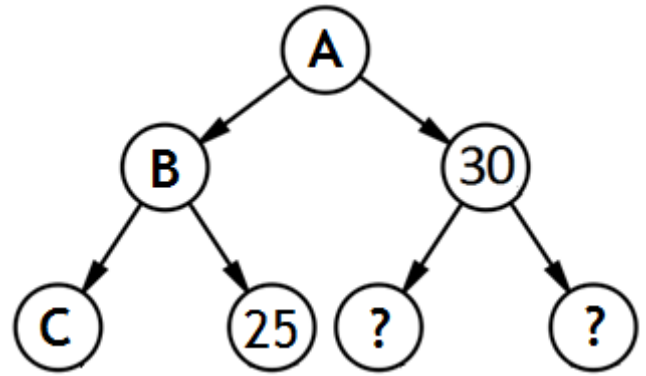
Áno Nie

c) $A > 30$ a $B > 30$

Áno Nie

d) Ak $A=40$ a $C=20$, potom $25 < B < 40$

Áno Nie



9. (3b) Máme dve verzie rekurzívnej metódy, ktoré počítajú presne to isté.

```

public static String funkcia1(int n) {
    if (n <= 0)
        return "*";
    else
        return funkcia1(n-1) + funkcia1(n-1);
}

public static String funkcia2(int n) {
    if (n <= 0)
        return "*";
    else {
        String pom = funkcia2(n-1);
        return pom + pom;
    }
}
  
```

Trvanie *funkcia1*(8):

Trvanie *funkcia2*(8):

Dĺžka reťazca
vráteného metódou
funkcia1(8):

Predpokladajte, že mechanizmus jedného volania metódy *funkcia1* alebo *funkcia2*, a tiež operácia zretazenia dvoch reťazcov "stojí počítač" presne 0.001 sekundy a všetky ostatné operácie a príkazy zanedbáme. Zistite, koľko bude trvať výpočet *funkcia1*(8) a *funkcia2*(8) a tiež, aký dlhý znakový reťazec sa pri tom vygeneruje.

Poznámka: netreba presnú hodnotu, stačí výraz (napr. $3 \cdot 8^{300} - \log 1000$)

10. (2b) Môže existovať aspoň 6-prvkový binárny strom, ktorého žiadne 2 uzly neobsahujú rovnakú hodnotu a ktorého postupnosti inorder a postorder prechodu sú rovnaké? Ak áno, nakreslite, ak nie, zdôvodnite.