Polsemestrálny test
teoretická časť

Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Píšte prosím čitateľne!

Hodnotenie, vyplní opravujúci:

Meno a priezvisko:	Skupina PAZ:	

1/2.5	2/2	3/1.5	4/1	5/1	6/1.5	7/2	8/3	9/8	∑/22.5

Zdôvodnenia majú byť stručné (1-3 vety) a zachycujúce podstatné argumenty.

1. Uvažujme binárnu haldu s maximom v koreni uloženú v poli, pričom koreň sa nachádza na indexe 0 (úplne vľavo). Zapíšte hodnoty 1, 2, ..., 12 do haldy tak, aby:
   - (1b) hodnota 10 sa nachádzala na čo najmenšom indexe poľa

   - (1.5b) hodnota 10 sa nachádzala na čo najväčšom indexe poľa

2. (2b) n-té Katalánovo číslo je definované rekurzívnym vzťahom:

$$C_0 = 1 \quad \text{and} \quad C_{n+1} = \frac{2(2n+1)}{n+2} C_n,$$

Napíšte rekurzívnu metódu (bez použitia cyklov) na výpočet Katalánovho čísla:

```
public static int katalan(int n) {



}
```

Poznámka: $2.(2n + 1).C_n$ je stále deliteľné číslom $n + 2$

3. (0.5+1b) Rozhodnite, či nasledujúca postupnosť hodnôt v poli môže byť výsledkom pivotizácie (metóda *rozdel* z QuickSort-u) s pivotom 7: $5, 6, 2, 6, 7, 9, 10, 8, 10, 11$. Svoju odpoveď zdôvodnite.

4. (1b) Doplňte definíciu:

`Binárny strom nazveme binárnym vyhľadávacím stromom práve vtedy, ak ...`

5. (1b) Aký bude obsah poľa p = $[11, 3, 21, 9, 13, 8]$ po vykonaní metódy `pisomka(p)`?

```
public void pisomka(int[] p) {
      Stack<Integer> z = new Stack<Integer>();
      for (int i = 0; i < p.length; i++)
            z.push(p[i]);

      for (int i = 0; i < p.length; i++)
            p[i] = z.pop();
}
```

---

6. (1.5b) Doplňte do poľa hodnoty tak, aby metóda zahada s parametrom referencujúcim toto pole vrátila hodnotu **true**.

9		82	13		3	12	26	5	

```
boolean zahada(int[] pole) {
      int[] p = pole.clone();
      Arrays.sort(p);
      for (int i = 1; i < p.length - 1; i++) {
            if ((p[i - 1] == p[i]) && (p[i] == p[i + 1])) {
                  return true;
            }
      }
      return false;
}
```

7. (1+1b) Čo najtesnejšie určte asymptotickú časovú zložitosť metódy zahada vzhľadom na n - veľkosť poľa p. Svoju odpoveď zdôvodnite.

```
int zahada(int[] p) {
      int vysledok = 0;
      int pocet = p.length;

      for (int i = 0; i < p.length; i++) {
            for (int j = 0; j < pocet; j++)
                  vysledok = vysledok + p[j];
            pocet = pocet / 2;
      }

      return vysledok;
}
```

8. (3b) Implementuje triedu `MnozinaCisel`, ktorej objekty uchovávajú nejakú dynamickú (modifikovateľnú) podmnožinu množiny $\{0, 1, 2, \ldots, n\}$, pričom n je určené parametrom konštruktora. Implementujte túto triedu s pamäťovou zložitosťou $O(n)$ tak, aby každá z metód `pridaj` (pridá prvok do množiny), `odober` (odstráni prvok z množiny), `obsahuje` (vráti, či sa hodnota nachádza v množine) mala časovú zložitosť $O(1)$.

Ak neviete napísať Java kód, pokúste sa napísať aspoň hlavné myšlienky implementácie jednotlivých metód.

```java
public class MnozinaCisel {

    public MnozinaCisel(int n) {

    }

    public void pridaj(int h) {

    }

    public void odober(int h) {

    }

    public boolean obsahuje(int h) {

    }
}
```

9. (1b za správnu, -0.5b za nesprávnu a 0b za žiadnu odpoveď) **Rozhodnite** o pravdivosti tvrdení vyznačením (napr. zakrúžkovaním) možnosti Áno alebo Nie:

A: Ak $f(n) = O(g(n))$, potom $n.f(n) = O(g(n))$.

Áno Nie

B: Nech $n > 8$. Každá n-prvková halda má aspoň $n/2$ listov.

Áno Nie

C: Nech $n > 8$. Každý binárny vyhľadávací strom uchovávajúci n hodnôt má aspoň $n/2$ listov.

Áno Nie

D: Algoritmus *QuickSort* vykoná $O(n.logn)$ porovnaní na každom vstupe veľkosti n.

Áno Nie

E: Existuje taký binárny vyhľadávací strom s aspoň 5 hodnotami, v ktorom je maximálna hodnota uložená v koreni.

Áno Nie

F: Máme 2 usporiadané (neklesajúce) postupnosti (polia) A veľkosti n a B veľkosti m. Potom v čase $O(n + m)$ možno zistiť, či existuje hodnota, ktorá sa nachádza v oboch postupnostiach, t.j. aj v A aj v B.

Áno Nie

Zdôvodnite svoju odpoveď k ľubovoľnému tvrdeniu B-F (2b, zakrúžkujte písmeno zdôvodňovaného tvrdenia):