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	2/1.5	3/2	4/2	5/1	6/2	7/1.5	8/6	9/4	∑/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). Táto halda obsahuje hodnoty 1, 2, ..., 11 (každú práve raz). Označme si H najväčšiu hodnotu uloženú v listoch haldy.

   - (1b) Krížikom označte tie pozície v poli, ktoré zodpovedajú listom haldy.

   - (0.5b) Akú najväčšiu hodnotu H je možné dosiahnuť? H:____________________
   - (1b) Nájdite takú haldu, v ktorej je hodnota H najväčšia možná a zaznačte jej hodnoty v poli:

2. (1.5b) Takmer haldou nazveme takmer úplný binárny strom, ktorý vieme vytvoriť z nejakej haldy zmenou hodnoty v jej koreni. Uvažujme metódu `void uhalduj(int idxKorena, int[] p)`, ktorá takmer haldu s koreňom na indexe `idxKorena` prerobí na haldu. S využitím metódy `uhalduj` napíšte kód, ktorý takmer úplný binárny strom reprezentovaný poľom `p` transformuje na haldu.

3. (2b) Uvažujme rekurzívnu metódu zahada:

```
void zahada(int index, int[] pole) {
  if ((index < 0) || (index >= pole.length)) {
    return;
  }

  zahada(pole[index] + index, pole);
  pole[index] = 0;
}
```

Doplňte chýbajúce hodnoty do poľa `pole` tak, aby po volaní `zahada(0, pole)` obsahovalo toto pole len samé hodnoty 0.

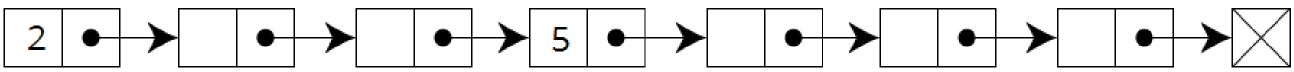
5	7			-2		0	89	

4. (1+1b) Ktorá z nižšie uvedených zložitostí najtesnejšie určuje asymptotickú časovú zložitosť metódy `zahada` vzhľadom na n - veľkosť poľa `p`. Svoju odpoveď zdôvodnite.

$O(1)$ $O(\log n)$ $O(n)$ $O(n.\log n)$ $O(n^2)$ $O(n^2.\log n)$ $O(2^n)$

```
void zahada(int[] p) {
      int idx = 0;
      int pocet = p.length * p.length;
      for (int j = 0; j < pocet; j++) {
            p[idx] += j;
            idx = (idx + 1) % p.length;
            pocet = pocet / 2;
      }
}
```

5. (1b) Uvažujme spájaný zoznam nižšie a metódu triedy `SpajanyZoznam`.




```
void zahada() {
  Uzol aktualny = prvy;
  while (aktualny != null) {
    if ((aktualny.dalsi != null) &&
          (aktualny.dalsi.hodnota % 2 == 0)) {
      aktualny.dalsi = aktualny.dalsi.dalsi;
    } else {
      aktualny = aktualny.dalsi;
    }
  }
}
```

Doplňte do spájaného zoznamu chýbajúce hodnoty tak, aby po aplikovaní metódy zahada mal zoznam dĺžku práve 4.

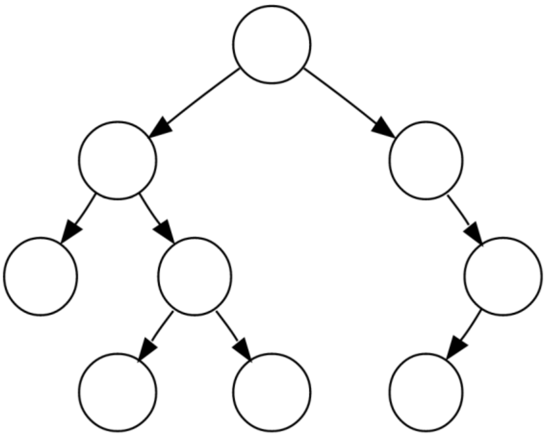
6. (2b) Uvažujme algoritmus bublinkového triedenia, kde volanie *vymen*(p, i, i + 1) vymení v poli p obsah políčok na indexoch i a i+1:

```java
void bubbleSort(int[] p) {
  boolean bolaVymena;
  do {
    bolaVymena = false;
    for (int i = 0; i < p.length - 1; i++) {
      if (p[i] > p[i + 1]) {
        vymen(p, i, i + 1);
        bolaVymena = true;
      }
    }
    System.out.println(Arrays.toString(p));
  } while (bolaVymena);
}
```

Čo vypíše metóda bubbleSort, ak pole p má na začiatku obsah [6, 4, 5, 1, 9]?

1.____
2.____
3.____
4.____
5.____
6.____

7. (1.5b) Do stromu vpravo vpíšte navzájom <u>rôzne</u> prirodzené čísla tak, aby jeho inorder postupnosť bola usporiadaná.



8. (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: Postupnosť hodnôt $5, 6, 2, 6, 7, 9, 10, 6, 10, 11$ môže byť výsledkom pivotizácie (metóda $rozdel$ z QuickSort-u) s pivotom 7.

Áno Nie

C: Inorder postupnosť binárneho vyhľadávacieho stromu je usporiadaná.

Áno Nie

D: V algoritme binárneho vyhľadávania volíme ako stredový index $X + (Y - X)/2$, pričom X je začiatočný a Y je koncový index aktuálne uvažovaného podpoľa. Ak stredový index budeme voliť ako $X + (Y - X)/3$, časová zložitosť modifikovaného algoritmu bude taktiež $O(\log n)$.

Áno Nie

E: V binárnej halde s maximom v koreni vieme nájsť maximum v čase $O(1)$. Minimálnu hodnotu v halde vieme nájsť v čase $O(\log n)$.

Áno Nie

F: Máme 2 binárne vyhľadávacie stromy, pričom každý obsahuje n hodnôt. V čase $O(n)$ vieme zistiť, či tieto binárne vyhľadávacie stromy obsahujú rovnaké prvky.

Áno Nie

9. (4b) Zdôvodnite svoju odpoveď k ľubovoľnému tvrdeniu A-C (1b) a k ľubovoľnému tvrdeniu D-F (3b) z predošlej úlohy - zakrúžkujte písmená zdôvodňovaných tvrdení.