



Polsemestrálny test praktická časť



Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Doplňujúce zdrojové kódy sú na stránke predmetu PAZ1b.

1. (5b) Uvažujme triedu SpajanyZoznam z prednášky o spájaných zoznamoch. Do triedy SpajanyZoznam pridajte metódu odstranCyklickyPrefix, ktorá ako parameter dostane nenullovú referenciu na pole čísel (potenciálne aj prázdne). Toto pole definuje nekonečnú cyklickú postupnosť čísel. Konkrétne, ak obsah poľa je [3, 8, 3], tak táto postupnosť je [3, 8, 3, 3, 8, 3, 3, 8, 3, 3, 8, 3, 3, 8, ...]. Po zavolaní metódy nech sa zo začiatku spájaného zoznamu odstráni čo najviac hodnôt tak, aby platilo, že odstraňovaná postupnosť je prefixom (=má rovnaký začiatok) uvedenej nekonečnej cyklickej postupnosti čísel.

```
public void odstranCyklickyPrefix(int[] prefix)
```

Metóda nech pracuje v lineárnom čase vzhľadom k dĺžke zoznamu, t.j. $O(n)$, kde n je dĺžka zoznamu, a s pamäťou $O(1)$. Za opakovaný prechod spájaným zoznamom je bodový zisk znížený o 1 bod.

Uvažujme zoznam [3, 4, 5, 3, 4, 3, 4, 5, 2]. Metóda zmení obsah tohto zoznamu takto:

- odstranCyklickyPrefix([3, 4, 8]) → [5, 3, 4, 3, 4, 5, 2]
[3, 4, 5, 3, 4, 3, 4, 5, 2]
- odstranCyklickyPrefix([3, 4, 5]) → [3, 4, 5, 2]
[3, 4, 5, 3, 4, 3, 4, 5, 2]
- odstranCyklickyPrefix([3, 4, 5, 3, 4, 3, 4, 5, 2, 2]) → []
[3, 4, 5, 3, 4, 3, 4, 5, 2]
- odstranCyklickyPrefix([]) → [3, 4, 5, 3, 4, 3, 4, 5, 2]

Uvažujme zoznam [3, 8, 3, 3, 8, 3, 3, 8, 8, 3, 2]. Metóda zmení obsah tohto zoznamu takto:

- odstranCyklickyPrefix([3, 8, 3]) → [8, 3, 2]
[3, 8, 3, 3, 8, 3, 3, 8, 8, 3, 2]
- odstranCyklickyPrefix(3, 8, 3, 3, 8) → [3, 8, 8, 3, 2]
[3, 8, 3, 3, 8, 3, 3, 8, 8, 3, 2]

2. (5b) Do triedy Osoba z prednášky o stromoch pridajte metódu, ktorá vráti, či všetky osoby v strome potomkov, ktoré nie sú listami (t.j. majú aspoň jedno dieťa), majú rovnaký počet detí. Strom potomkov zahŕňa aj osobu v koreni. Inými slovami zaujíma nás, či pre každý uzol v strome platí, že má 0 detí alebo taký počet detí ako koreň.

```
public boolean rovnakyPocetDeti()
```

3. (5b) Každý objekt v Jave má metódu `hashCode`, ktorá vráti číselný odtlačok obsahu objektu. Platí, že objekty s rovnakým obsahom majú rovnaký hash kód.

Z reťazca sa vytratil medzery, no našťastie máme uložený jeho hash kód. Vaša úloha je nasledovná: Do zadaného reťazca bez medzier doplňte medzery tak, aby výsledný reťazec mal rovnaký hash kód ako pôvodný reťazec s medzerami. Medzi každú dvojicu susedných znakov môžete doplniť nanajvýš jednu medzeru (inými slovami v pôvodnom slove medzery neboli na začiatku ani na konci a žiadne 2 medzery neboli susedné).

Vytvorte triedu `NajdiSlovo` a v nej metódu `najdiRiesenie`, ktorá pre zadaný reťazec bez medzier a hash kód (parametre konštruktora) nájde taký reťazec, ktorý vznikne doplnením medzier do zadaného reťazca a jeho hash kód bude rovnaký, ako bol zadaný. Ak taký reťazec neexistuje, metóda nech vráti `null`.

```
public class NajdiSlovo {  
  
    public NajdiSlovo(String slovo, int hashCode) {  
    }  
  
    public String najdiRiesenie() {  
    }  
}
```

Poznámka: Svoju implementáciu môžete otestovať aj takto:

```
System.out.println(new NajdiSlovo("jetook", "je to ok".hashCode()))
```