

## Cvičenia č. 8, úloha č. 3

Sled  $v_0, v_1, \dots, v_n$  v grafe – či už orientovanom alebo neorientovanom – nazveme *uzavretým*, ak sa vrcholy  $v_0$  a  $v_n$  rovnajú. V orientovanom grafe navyše nazveme *cyklom* – alebo *orientovanou kružnicou* – uzavretý sled  $v_0, v_1, \dots, v_n$  taký, že  $n \geq 1$  a vrcholy  $v_0, \dots, v_{n-1}$  sú po dvoch rôzne. Cyklus je teda sled nenulovej dĺžky, v ktorom sa počiatočný a koncový vrchol rovnajú a ide pritom o jedinú dvojicu opakujúcich sa vrcholov sledu. Každá slučka je tak súčasne aj cyklom.

Priložený ZIP archív obsahuje triedy pre grafy z prednášky a tiež kostru triedy `DirectedCycles`. Vašou úlohou je doplniť do tejto triedy implementáciu statickej metódy `belongsToCycle`, ktorá vezme ako argumenty orientovaný<sup>1</sup> graf `g` a jeho vrchol `vertex`; na výstupe táto metóda vráti booleovskú hodnotu podľa toho, či je vrchol `vertex` súčasťou nejakého cyklu v grafe `g`. Výstupom tejto metódy teda má byť `true` práve vtedy, keď v grafe `g` existuje cyklus  $v_0, v_1, \dots, v_n$  taký, že niektorý z vrcholov  $v_0, v_1, \dots, v_n$  je rovný vrcholu `vertex`. Trieda `DirectedCycles` už obsahuje hotovú implementáciu metódy `main`, ktorá načíta zo vstupu orientovaný graf nasledovaný nejakým jeho vrcholom a pre túto dvojicu vstupov zavolá metódu `belongsToCycle`. Pri implementácii metódy `belongsToCycle` môžete použiť vhodne upravené prehľadávanie do hĺbky alebo do šírky z prednášky. Okrem samotnej tejto metódy môžete v prípade potreby do triedy `DirectedCycles` doplniť aj ďalšie metódy; implementáciu metódy `main` ale nemeňte.

Na testovač odovzdávajte iba súbor `DirectedCycles.java` obsahujúci zdrojový kód vami upravenej triedy `DirectedCycles` v balíku `graphs`. Zvyšné triedy balíka `graphs` k nej budú na testovači priložené.

### Príklad vstupu č. 1:

```
4 4
0 1
1 2
2 3
3 2
1
```

### Príklad výstupu č. 1:

```
Vrchol 1 nelezi na orientovanej kruznici.
```

### Príklad vstupu č. 2:

```
4 4
0 1
1 2
2 3
3 2
2
```

### Príklad výstupu č. 2:

```
Vrchol 2 lezi na orientovanej kruznici.
```

---

<sup>1</sup>V prípade, že sa ako argument použije neorientovaný graf, interpretuje sa tento ako orientovaný graf v rovnakom zmysle ako na prednáške.