Cvičenia č. 8, úloha č. 2

Nech $k \geq 1$ je prirodzené číslo. Pod (*vrcholovým*) *k-farbením* neorientovaného grafu rozumieme zobrazenie, ktoré každému vrcholu grafu priradí číslo – takzvanú *farbu* – z množiny $\{0, \ldots, k-1\}$ tak, aby všetky hrany v grafe viedli medzi vrcholmi s rôznymi farbami. Pre graf s množinou vrcholov V teda ide o zobrazenie $f\colon V \to \{0, \ldots, k-1\}$ také, že $f(u) \neq f(v)$ kedykoľvek v grafe vedie hrana z vrcholu u do vrcholu v. Pre grafy so slučkami teda špeciálne neexistuje žiadne k-farbenie bez ohľadu na k.

Priložený ZIP archív obsahuje triedy pre grafy z prednášky a tiež kostru triedy `Colourings` (všetko v balíku `graphs`). V triede `Colourings` je už hotový konštruktor, ktorý ako argument dostane neorientovaný graf `g` a celé číslo `colours` reprezentujúce hodnotu k z definície k-farbenia. Následne konštruktor zavolá rekurzívnu metódu `search`, ktorá pomocou prehľadávania s návratom nájde všetky `colours`-farbenia grafu `g` a uloží ich do zoznamu `colourings`. Každé farbenie grafu o n vrcholoch $0, \ldots, n-1$ je pritom reprezentované ako zoznam dĺžky n, kde pre $v = 0, \ldots, n-1$ je na v-tej pozícii zoznamu uložená farba v tomto farbení priradená vrcholu v.

Takisto je v triede `Colourings` hotová metóda `getColourings`, ktorá na výstupe vráti zoznam všetkých nájdených farbení.

Vašou úlohou bude dokončiť implementáciu metódy `search` realizujúcej samotné prehľadávanie s návratom:

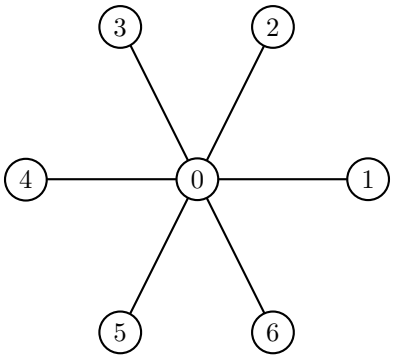
- Jediným argumentom tejto metódy je celé číslo `vertex`.
- Ak je hodnota argumentu `vertex` rovná n, kde n je počet vrcholov grafu `g`, našlo sa ďalšie farbenie, ktoré sa pridá do zoznamu `colourings`. Ošetrenie tohto prípadu je už v metóde `search` implementované.
- Ak je hodnotou argumentu `vertex` číslo z množiny $\{0, \ldots, n-1\}$, mala by metóda `search` vyskúšať všetky prípustné farby pre vrchol `vertex` a pre každú z nich rekurzívne zavolať samú seba s argumentom `vertex + 1`. *Ošetrenie tohto prípadu je potrebné doimplementovať.*

Kód píšte výhradne do bloku `else` metódy `search`. Ostatné časti triedy `Colourings` nemodifikujte.

Prehľadávanie s návratom napíšte tak, aby sa neskúšali úplne všetky možnosti priradenia farieb vrcholom,[1] ale aby sa farba danému vrcholu priradila iba v prípade, že žiaden z doposiaľ ofarbených vrcholov, ktorý je s týmto vrcholom spojený hranou, nemá priradenú rovnakú farbu. Farbenia môžete generovať v ľubovoľnom poradí, ale každé `colours`-farbenie grafu `g` by malo byť do zoznamu `colourings` pridané práve raz.

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

Príklad. Uvažujme graf na nasledujúcom obrázku.



Existujú práve dve 2-farbenia tohto grafu, ktoré môžu byť v zmysle vysvetlenom vyššie reprezentované pomocou zoznamov `[0, 1, 1, 1, 1, 1, 1]` a `[1, 0, 0, 0, 0, 0, 0]`.

[1] Takéto riešenia môžu na testovači naraziť na časový limit.