

Test č. 3, úloha č. 2

Pripomeňme si, že sled v_0, v_1, \dots, v_n v orientovanom grafe nazývame *uzavretým*, ak je v_0 rovné v_n . *Orientovanou kružnicou* alebo *cyklom* v orientovanom grafe rozumieme 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. Každá slučka je tak súčasne aj orientovanou kružnicou.

Dĺžkou orientovanej kružnice v_0, \dots, v_n rozumieme dĺžku tohto sledu – čiže číslo n . Hovoríme, že orientovaná kružnica v_0, v_1, \dots, v_n *prechádza cez vrchol* v , ak pre nejaké $k \in \{0, \dots, n\}$ je v_k rovné v .

Priložený ZIP archív obsahuje balík graphs s triedami pre grafy z prednášky; okrem toho obsahuje aj kostru triedy ShortestCycleThroughVertex, ktorá má realizovať hľadanie niektorej spomedzi *najkratších* orientovaných kružníc v danom grafe prechádzajúcich cez daný vrchol. V triede ShortestCycleThroughVertex doprogramujte:

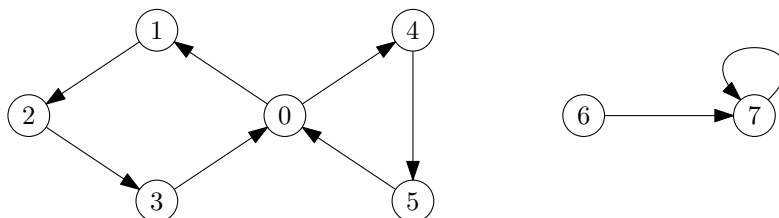
- Konštruktor `public ShortestCycleThroughVertex(DirectedGraph g, int vertex)`, ktorý nájde niektorú spomedzi najkratších orientovaných kružníc v orientovanom grafe g prechádzajúcich cez vrchol $vertex$ (ak nejaká takáto kružnica existuje). Môžete predpokladať korektnosť obidvoch argumentov konštruktora.
- Metódu `public int getShortestCycleLength()`, ktorá vráti dĺžku najkratšej kružnice nájdenej konštruktorom. V prípade, že hľadaná orientovaná kružnica neexistuje, vráti táto metóda na výstupe číslo -1 .
- Metódu `public List<Integer> getShortestCycle()`, ktorá vráti najkratšiu kružnicu nájdenú konštruktorom v podobe zoznamu obsahujúceho postupne všetky vrcholy tvoriace túto kružnicu. V prípade, že hľadaná orientovaná kružnica neexistuje, vráti táto metóda na výstupe referenciu `null`.

V prípade potreby môžete v triede ShortestCycleThroughVertex definovať aj ďalšie pomocné metódy a premenné inštancie.

Úlohu riešte pomocou prehľadávania do šírky alebo nejakou podobne efektívnou metódou – špeciálne úlohu *neriešte* prehľadávaním s návratom, ktoré je v tomto prípade zbytočne neefektívne a na testovači by s najväčšou pravdepodobnosťou narazilo na časový limit.

Na testovač odovzdávajte iba súbor ShortestCycleThroughVertex.java obsahujúci zdrojový kód vami doplnenej triedy. Pri tvorbe kódu rešpektujte konvencie jazyka Java a elementárne zásady objektovo orientovaného programovania.

Príklad. Uvažujme orientovaný graf na nasledujúcom obrázku a predpokladajme, že sme vytvorili inštanciu triedy ShortestCycleThroughVertex pre tento graf a niektorý z jeho vrcholov.



- Pre vrchol 0 uvažovaného grafu je výstupom metódy `getShortestCycleLength` číslo 3 a výstupom metódy `getShortestCycle` môže byť napríklad zoznam `[0, 4, 5, 0]`.
- Pre vrchol 1 uvažovaného grafu je výstupom metódy `getShortestCycleLength` číslo 4 a výstupom metódy `getShortestCycle` môže byť napríklad zoznam `[1, 2, 3, 0, 1]`.
- Pre vrchol 6 uvažovaného grafu je výstupom metódy `getShortestCycleLength` číslo -1 a výstupom metódy `getShortestCycle` je referencia `null`.
- Pre vrchol 7 uvažovaného grafu je výstupom metódy `getShortestCycleLength` číslo 1 a výstupom metódy `getShortestCycle` je zoznam `[7, 7]`.