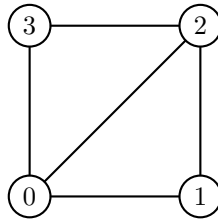


Príprava na skúšku

Úloha č. 5: Tetivy

Pripomeňme si, že *kružnicou* v neorientovanom grafe rozumieme uzavretý sled $v_0, v_1, \dots, v_n = v_0$ taký, že $n \geq 1$ a vrcholy $v_0, \dots, v_{n-1}$, ako aj neorientované hrany $\{v_0, v_1\}, \{v_1, v_2\}, \dots, \{v_{n-1}, v_n\}$, sú po dvoch rôzne. Ide teda o sled nenulovej dĺžky, v ktorom sa počiatočný a koncový vrchol rovnajú, ale inak sa v ňom neopakujú žiadne vrcholy ani neorientované hrany.

Tetivou kružnice v neorientovanom grafe bez slučiek rozumieme neorientovanú hranu spájajúcu dva rôzne vrcholy na tejto kružnici, ktorá ale nie je súčasťou kružnice samotnej. Napríklad v grafe na nasledujúcom obrázku je hrana $\{0, 2\}$ jedinou tetivou kružnice $0, 1, 2, 3, 0$.



Priložený ZIP archív obsahuje balík `graphs` a v ňom všetky triedy pre grafy z prednášky, ako aj kostru triedy `FixedChordCountCycles`. V tejto triede doprogramujte:

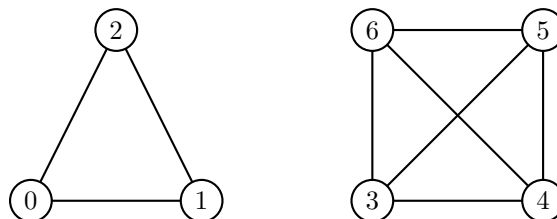
- Konštruktor

```
public FixedChordCountCycles(UndirectedGraph g, int throughVertex, int chordCount),
```

ktorý zistí, či v neorientovanom grafe bez slučiek `g` prechádza cez vrchol `throughVertex` aspoň jedna kružnica s práve `chordCount` rôznymi tetivami. Výslednú booleovskú hodnotu si pritom uloží do vhodnej premennej inštalácie.

- Metódu `public boolean existsCycle()`, ktorá vráti booleovskú hodnotu predvypočítanú v konštruktore.

Príklad. Uvažujme neorientovaný graf na nasledujúcom obrázku.



Cez vrchol 1 tohto grafu potom prechádza kružnica s nula tetivami, ale neprechádza tadiaľ žiadna kružnica s jednou alebo dvoma tetivami. Naopak cez vrchol 6 prechádza kružnica s nula tetivami aj kružnica s dvoma tetivami, ale neprechádza cezeň žiadna kružnica s práve jednou tetivou.

Na testovač odovzdávajte iba súbor `FixedChordCountCycles.java` obsahujúci zdrojový kód vašej triedy.