

Náhradný test, úloha č. 3

Biklikou v neorientovanom grafe G s množinou vrcholov V nazveme dvojicu disjunktných podmnožín V_1, V_2 množiny V takú, že každý vrchol z množiny V_1 je spojený hranou s každým vrcholom z množiny V_2 . Bikliky v grafe G teda tiež možno opísať ako úplné bipartitné podgrafy grafu G . Pre dané prirodzené čísla s, t budeme hovoriť, že biklika je veľkosti (s, t) , ak je množina V_1 veľkosti s a množina V_2 veľkosti t (alebo naopak; dvojice biklík, ktoré vzniknú zámenou množín V_1 a V_2 budeme v tejto úlohe stotožňovať).

Priložený archív obsahuje balík `graphs` s triedami z prednášky a kostru triedy `FixedSizeBiclique`. Doprogramujte túto triedu tak, aby jej inštancie realizovali hľadanie niektorej spomedzi biklík danej veľkosti v grafe z argumentu konštruktora.

Trieda `FixedSizeBiclique` by mala poskytovať:

- Konštruktor

```
public FixedSizeBiclique(UndirectedGraph g, int s, int t),
```

ktorý ako vstupy dostane neorientovaný graf g a prirodzené čísla s, t . Pre tieto vstupy konštruktor nájde ľubovoľnú spomedzi biklík v neorientovanom grafe g veľkosti (s, t) .

Môžete predpokladať korektnosť vstupov konštruktora.

- Metódu

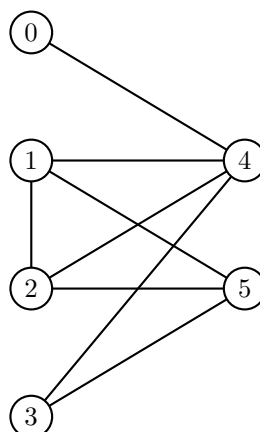
```
public Set<Set<Integer>> getBiclique(),
```

ktorá na výstupe vráti bikliku nájdenú konštruktorom v podobe *nemodifikovateľnej* množiny pozostávajúcej z dvoch *nemodifikovateľných* množín vrcholov. V prípade, že v grafe žiadna biklika požadovanej veľkosti neexistuje – a teda konštruktor žiadnu bikliku nenašiel – vráti táto metóda na výstupe referenciu `null`.

V prípade potreby môžete do tejto triedy pridať aj ďalšie pomocné metódy.

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

Príklad 1. Uvažujme neorientovaný graf na nasledujúcom obrázku, $s = 3$ a $t = 2$.



Biklika veľkosti (s, t) v tomto grafe potom môže byť reprezentovaná množinou $[[1, 2, 3], [4, 5]]$.