

### Úkoly obecně

Úkoly odevzdávejte emailem vašemu cvičícímu a předmět emailu začněte OPT: HW2. Vypracovaný úkol se skládá z generátoru LP/IP a dokumentace. Na adrese <http://iuuk.mff.cuni.cz/~husek/opt-ukol2.zip> naleznete archiv s připravenými vstupy, na nichž máte váš program testovat.<sup>1</sup> Kromě základních vstupů označených `vstupi-x.txt`, kde *i* je číslo úlohy a *x* je číslo vstupu, (a k těmto vstupům se vztahuje doba běhu uvedená níže), archiv dále obsahuje malé vstupy označené `-s`.

Pro první úlohu platí stejná pravidla jako v prvním úkolu – jejím řešením je generátor celočíselného nebo lineárního programu, který nám ve zdrojové podobě zašlete. Generátor samotný ale lineární program neřeší. Samozřejmě můžete použít i `glpk` a pak vaším výstupem bude rovnou vyřešená úloha.

Pro druhou úlohu toto pravidlo neplatí – generátor sám může volat příkaz `glpsol -m program.mod` či `glpk` a číst jeho výstup, dokonce může generátor vytvořit a spustit několik lineárních / celočíselných programů. Výstup pro druhou úlohu bude tedy vypsán samotným generátorem. Stále ale platí požadavek, že váš generátor musí nějakým netriviálním způsobem využít lineární / celočíselné programování.

Obě úlohy by měly obsahovat dokumentaci, podobně jako tomu bylo v prvním úkolu.

Je-li cokoli nejasné, zeptejte se svého cvičícího.

---

<sup>1</sup> Váš generátor samozřejmě musí fungovat i pro jiné podobné vstupy.

# Úloha 1 – Nezávislost v grafu známostí ve škole

[12 bodů]

Na vstupu máte velký graf, který popisuje *známost ve škole*. Můžete si představit velkou školu se třídami, kde hrany modelují relaci „znají se“, která je tentokrát symetrická.

Grafy známostí ve škole nejsou rovnoměrně náhodné, ale vždy splňují pravidlo, že v nich jsou třídy. Ve třídě se všichni žáci znají, ale mezi třídami navzájem se znají většinou málo, a tedy takových hran bude typicky málo.

Vášim úkolem bude najít největší nezávislou množinu, tedy skupinu žáků, kteří se navzájem vůbec neznají.

Důležité je, že vaše řešení by mělo být relativně rychlé. Na běžných počítačích by na ukázkových vstupech mělo sběhnout v řádu *sekund*, ne minut. K tomuto budeme přihlížet při vyhodnocování řešení.

## Formát vstupu

Soubor s neorientovaným grafem má následující formát: První řádek začíná slovem `GRAPH` a za ním následuje počet vrcholů a počet hran, obě odděleno mezerami, a na konci prvního řádku je dvojtečka. Vrcholy jsou číslovány od nuly. Další řádky mají tvar `i--j` a určují jednotlivé hrany. Příklad  $K_{1,2}$ :

```
GRAPH 3 2:  
0 -- 1  
0 -- 2
```

## Formát výstupu

Váš vygenerovaný lineární/celočíselný program v `MathProgu` může vypisovat jakékoli informace uznáte za vhodné, ale výstup vždy musí obsahovat následující povinnou část: Povinná část je ohraničena řádky `#OUTPUT: k` a `#OUTPUT END`, kde `k` je velikost nalezené nezávislé množiny. Mezi nimi je výpis pořadí vrcholů ve tvaru `v_i`, kde `i` jsou čísla vrcholů tvořících největší nezávislou množinu. Příklad pro  $K_{1,2}$  uvedenou výše:

```
#OUTPUT: 2  
v_1  
v_2  
#OUTPUT END
```

## Úloha 2 – Graf bez krátkých cyklů se vrací!

[13 bodů]

Stejně jako ve druhé úloze 1. úkolu se budete snažit odebrat všechny orientované cykly z grafu, tentokrát ale bez omezení na jejich délku.

Na vstupu máte orientovaný graf s váhovou funkcí na hranách. Vaším úkolem je napsat IP, který najde nejmenší váženou podmnožinu hran takovou, že po jejím odebrání graf nebude obsahovat žádnou **orientovanou** kružnici. Graf na vstupu neobsahuje smyčky ani cykly délky 2.

V této úloze, na rozdíl od všech předešlých, nemá být výstupem vašeho programu zadání pro `glpsol`, ale máte si `glpsol` volat sami. Díky tomu **můžete** při řešení této úlohy **volat `glpsol` opakovaně**, což se vám pravděpodobně bude hodit.

Stejně jako v první úloze by vaše řešení mělo být co nejrychlejší. Očekávaný čas běhu na většině ukázkových vstupů je pod minutu (a na spoustě z nich v několika sekundách, velmi záleží na konkrétním vstupu) ne desítky minut. K tomuto budeme přihlížet při vyhodnocování řešení.

### Formát vstupu

Soubor s orientovaným grafem má následující formát: První řádek začíná slovem `WEIGHTED DIGRAPH` a za ním následuje počet vrcholů a počet hran, obě odděleno mezerami, a na konci prvního řádku je dvojtečka. Vrcholy jsou číslovány od nuly. Další řádky mají tvar  $i \rightarrow j (w)$  a určují jednotlivé hrany,  $w$  je nezáporná celočíselná váha hrany. Příklad  $K_4$ :

```
WEIGHTED DIGRAPH 4 6:  
0 --> 1 (4)  
0 --> 2 (3)  
0 --> 3 (1)  
1 --> 2 (4)  
2 --> 3 (2)  
3 --> 1 (5)
```

### Formát výstupu

Generátor může vypisovat jakékoli informace uznáte za vhodné, ale jeho výstup (ať už standardní výstup nebo výstupní soubor) vždy musí obsahovat následující povinnou část: Povinná část je ohraničena řádky `#OUTPUT: W` a `#OUTPUT END`,  $W$  je celková váha odebraných hran. Mezi nimi je výpis odebraných hran ve tvaru  $i \rightarrow j$ . Příklad pro vstup výše:

```
#OUTPUT: 2  
2 --> 3  
#OUTPUT END
```