

## Úkoly obecně

Úkoly odevzdávejte emailem vašemu cvičícímu a předmět emailu začněte OPT: HW1. Vypracovaný úkol se skládá z generátoru LP/IP a dokumentace. Na adrese <http://iuuk.mff.cuni.cz/~husek/opt-ukol1.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` a (pravděpodobně) problematické vstupy pro první úlohu označené `-c`.

Generátor LP/IP je program v rozumném programovacím jazyce, který transformuje vstup (formát vstupu viz jednotlivé úlohy) na LP/IP program v jazyce GNU MathProg pro řešič `glpsol`. Jednoduchý manuál ke GNU MathProg naleznete na [http://iuuk.mff.cuni.cz/~bohmf/texts/mathprog\\_intro\\_cz.html](http://iuuk.mff.cuni.cz/~bohmf/texts/mathprog_intro_cz.html), oficiální dokumentace je součástí distribuce `glpk`, případně rozumně aktuální verze je online na <https://www3.nd.edu/~jeff/mathprog/glpk-4.47/doc/gmpl.pdf>.

Vstup můžete načítat ze standardního vstupu nebo ze souboru, jehož jméno bude předáno jako argument programu, obdobně pro výstup. Můžete předpokládat, že vstup je vždy ve validním formátu.

Formát výstupu vygenerovaného LP/IP (tj. co vypíše `glpsol -m vygenerovane_lp.mod`) je popsán u jednotlivých příkladů, v dokumentaci popište, co znamená, pokud vaše LP/IP nemá řešení (v tomto případě LP/IP nemusí vypisovat povinnou část výstupu).

Odevzdávejte zdrojový kód, ne přeložený program. Za rozumné určitě považujeme jazyky C, C++, Java, C#, Python, Perl, Bash. (Pokud toužíte po exotičtějším jazyce, zeptejte se svého cvičícího.) Odevzdaný zdrojový kód generátoru by měl být čitelný, formátovaný a v rozumné míře okomentovaný. Na druhou stranu, vygenerovaný lineární program může být nečitelný a dlouhý, jak jen potřebujete.

Druhou povolenou možností, jak úkoly vyřešit, je negenerovat výstup v MathProgu, ale užít knihovnu `glpk` a s její pomocí problém přímo vyřešit (`glpsol` je vlastně jen jednoduchý obal knihovny `glpk`). Tento způsob bude nejspíše pracnější.

Nedílnou součástí řešení je dokumentace, která musí obsahovat:

1. popis, jak program sestavit,
2. informace, jak program ovládat,
3. stručný popis, jak bude vypadat vaše LP/IP, a
4. co znamená, když vygenerovaný LP/IP nemá řešení.

Dokumentaci odevzdejte ve formátu pdf nebo plain text (Markdown je povolený). Dokumentace nemusí být dlouhá; měla by se vejít na 1 nebo 2 stránky.

**Upozorňujeme**, že řešení, která (kombinatoricky) vyřeší zadanou úlohu, a pak na ní pustí triviální LP, nebudou uznávána. Cílem tohoto domácího úkolu je seznámit se s tvorbou lineárních programů. Slibujeme, že druhý domácí úkol bude větší výzva, která jen tak snadno kombinatoricky nepůjde.

Je-li cokoli nejasné, zeptejte se svého cvičícího.

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

# Úloha 1 – Topologické uspořádání

[10 bodů]

Na vstupu máte orientovaný graf. Vaším úkolem je napsat LP, který nalezne (částečné) uspořádání vrcholů takové, že pro každou hranu  $uv$  platí, že vrchol  $u$  je v daném uspořádání menší než  $v$ . (Uspořádání modelujte pomocí nezáporných celých čísel.) Očekávaná doba běhu na připravených vstupech je cca 10 sekund.

## Formát vstupu

Soubor s orientovaným grafem má následující formát: První řádek začíná slovem DIGRAPH a za ním následuje počet vrcholů a počet hran, obě odděleno mezerami. Vrcholy jsou číslovány od nuly. Další řádky mají tvar  $i-->j$  a určují jednotlivé hrany. Příklad  $K_{1,2}$ :

```
DIGRAPH 3 2
0 --> 1
0 --> 2
```

## Formát výstupu

Program může vypisovat jakékoli informace uznáte za vhodné, ale výstup vždy musí obsahovat následující povinnou část: Povinná část je ohraničena řádky #OUTPUT: a #OUTPUT END. Mezi nimi je výpis pořadí vrcholů ve tvaru  $v_i: x$ , kde  $i$  je číslo vrcholu a  $x$  jeho pořadí. Pořadí musí být nezáporné celé číslo. Příklad pro orientaci  $K_{1,2}$  uvedenou výše:

```
#OUTPUT:
v_0: 0
v_1: 1
v_2: 1
#OUTPUT END
```

## Úloha 2 – Graf bez krátkých cyklů

[15 bodů]

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 délky 4 či kratší. Graf na vstupu neobsahuje smyčky. Očekávaná doba běhu na připravených vstupech je 5 až 120 sekund dle vstupu.

### 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. 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

Program může vypisovat jakékoli informace uznáte za vhodné, ale výstup 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
```