

INFORMACE K PRAKTICKÝM DOMÁCÍM ÚKOLŮM

Optimalizační metody LS 2017/18

Úkoly odevzdávejte emailem vašemu cvičícímu. Vypracovaný úkol se skládá z generátoru LP/CLP a stručné dokumentace. Ke každému úkolu naleznete archiv s připravenými vstupy, na nichž máte váš program testovat.¹

Generátor LP/CLP (lineárního či celočíselného programu) je program v rozumném programovacím jazyce, který transformuje vstup (formát vstupu viz jednotlivé úlohy) na LP/CLP program v jazyce GNU MathProg pro řešič glpsol. Jednoduchý manuál ke GNU MathProg naleznete na http://iuk.mff.cuni.cz/~bohm/texts/mathprog_intro_cz.html, oficiální dokumentace je součástí distribuce glpk, případně rozumně aktuální verze je online na <http://kam.mff.cuni.cz/~mitchondra/let1718/optimalizace/gmpl.pdf>.

Vstup načítejte ze standardního vstupu a výstup vypisujte na standardní výstup. Můžete předpokládat, že vstup je vždy ve validním formátu.

Formát výstupu vygenerovaného LP/CLP (tj. co vypíše `glpsol -m vygenerovane_lp.mod`) je popsán u jednotlivých příkladů, v dokumentaci popište, co znamená, pokud vaše LP/CLP nemá řešení.

Odevzdávejte zdrojový kód, ne přeložený program. Zdrojový kód musí být možné spustit či zkompileovat v počítačové laboratoři Rotunda na (libovolném) počítači s Linuxem. (Počítače by měly být vybaveny stejným softwarem.) Za rozumné určitě považujeme jazyky Python, Perl, Bash, C, C++, Java, C#. Můžete používat standardní knihovny příslušného jazyka (jsou-li k dispozici v labu). Pokud si nejste jistí, jestli se daná knihovna dá považovat za standardní, zeptejte se cvičího.

(Pokud si s Linuxem opravdu nerozumíte či v laboratoři nemají kompilátor vašeho oblíbeného jazyka, ozvěte se a pokusíme se to nějak vyřešit. Podobně, pokud toužíte po exotičtějším jazyce nebo knihovně, která není v labu k dispozici.)

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, ale je vhodný, pokud si chcete vstup netriviálním způsobem ztransformovat a po vyřešení LP/CLP bude třeba výsledek „ztransformovat zpět“.

Nedílnou součástí řešení je dokumentace, která musí obsahovat:

1. popis, jak program sestavit (na počítači v laboratoři Rotunda),
2. informace, jak program ovládat,
3. matematický zápis úlohy LP/CLP, na kterou pouštíte řešit GLPK.
4. zdůvodnění správnosti všech postupů, které zrychlují výpočet, například snížení počtu proměnných či podmínek, rozdělení zadaného problému na menší podúlohy LP/CLP, předzpracování zadaných dat, ...

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

Veškerý odevzdaný kód musí být vaším dílem, bez jakékoliv externí „inspirace“! Speciálně tedy nesdílejte svůj kód s kýmkoliv jiným (vyjma vašeho cvičího), aby dotyčný nebyl omylem „inspirován“.

Je-li cokoli nejasné, zeptejte se svého cvičícího.

¹ Váš generátor samozřejmě musí fungovat i pro jiné podobně velké vstupy.