# Računalništvo 3 + Kombinatorična optimizacija: 3. izpit

14. september 2009

Čas reševanja: 100 minut

## Naloga 1 [25 točk]

Dana je enoparametrična družina linearnih programov $LP(a)$, kjer je $a$ poljubno realno število:

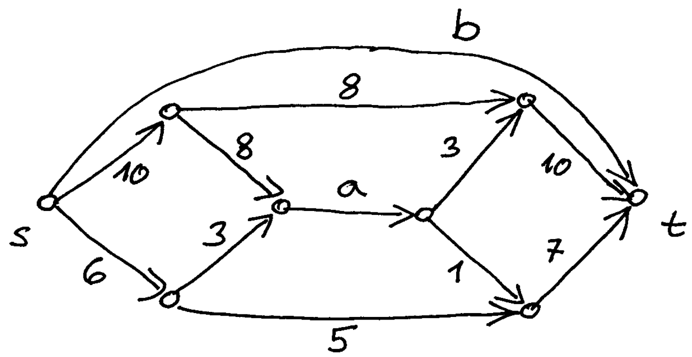
$$\begin{aligned} \max \quad & 2x_1 + 3x_2 + x_3 \\ & x_1 - x_2 + 2x_3 \leq 1 \\ & 4x_1 + 2x_2 - x_3 \leq a \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

(i) Z dualnim dopolnjevanjem dokaži, da je vektor $y^* = [\frac{5}{3}, \frac{7}{3}]^T$ optimalna rešitev duala linearnega programa $LP(2)$.

(ii) Naj bo $z(a)$ optimalna vrednost linearnega programa $LP(a)$. (Če $LP(a)$ nima optimalne rešitve, potem vrednost $z(a)$ ni definirana.) Ugotovi, za katere $a \in \mathbb{R}$ vrednost $z(a)$ obstaja, in jo izračunaj.

## Naloga 2 [25 točk]

Poišči maksimalni $(s,t)$-pretok in minimalni $(s,t)$-prerez v omrežju s spodnje slike. Odgovor bo seveda odvisen od vrednosti parametrov $a \geq 0$ in $b \geq 0$.




## Naloga 3 [25 točk]

Odločitveni problem PETINA kot vhod dobi zaporedje naravnih števil $a_1, a_2, \ldots, a_n$, zanima pa nas, ali obstaja taka množica $I \subseteq \{1, 2, \ldots, n\}$, da velja

$$4 \cdot \sum_{i \in I} a_i = \sum_{i \notin I} a_i \,.$$

(i) Dokaži, da je problem PETINA NP-poln.

(ii) Recimo, da imamo podprogram, ki reši odločitveni problem PETINA. Opiši polinomski algoritem, ki kot vhod dobi zaporedje naravnih števil $a_1, a_2, \ldots, a_n$ in v polinomskem času s polinomsko mnogo klici prej omenjenega podprograma poišče množico indeksov $I \subseteq \{1, 2, \ldots, n\}$, ki reši problem PETINA (oziroma ugotovi, da problem nima rešitve).

## Naloga 4 [25 točk]

Optimizacijski problem UTEŽENO POKRITJE kot vhod dobi neusmerjen graf $G$ in utežitev točk $w\colon V(G) \to \mathbb{N}$, iščemo pa tako pokritje $C \subseteq V(G)$, za katero je vrednost $\sum_{v \in C} w(v)$ čim manjša.

Problem UTEŽENO POKRITJE rešujemo z naslednjim postopkom:

```
C = {}
∀v ∈ V(G) : t(v) = w(v)
∀e ∈ E(G) : c(e) = 0
while C ni pokritje do
    Izberi nepokrito povezavo uv.
    m = min{t(u), t(v)}
    t(u) = t(u) − m
    if t(u) = 0 then C = C ∪ {u}
    t(v) = t(v) − m
    if t(v) = 0 then C = C ∪ {v}
    c(uv) = m
endwhile
return C
```

(i) Dokaži, da ima zgornji algoritem kot algoritem za reševanje problema UTEŽENO POKRITJE faktor aproksimacije 2. (Nasvet: Vrednosti $c(e)$ v algoritmu v resnici ne potrebujemo, pomagajo pa nam pri dokazovanju. Poskusite vsoto končnih vrednosti $c(e)$ povezati z vrednostjo optimalne rešitve in vrednostjo rešitve, ki jo vrne algoritem.)

(ii) Ali je število 2 najboljša možna ocena faktorja aproksimacije za zgornji algoritem?