Operacijske Raziskave

Zbirka vaj - delovna verzija

Blaž JELENC

22. maj 2014

Kazalo

Poglavje 1

Uvod

Gre za zbirko nalog pri predmetu *Operacijske Raziskave*, ki je v nastajanju, zato po vsej verjetnosti vsebuje nekaj napak ter tudi kakšno nalogo, ki morda ne spada najbolj v okvir tega predmeta.

Poglavje 2

Časovna zahtevnost algoritmov

Razredi funkcij. Naj bosta $f, g : \mathbb{N} \to \mathbb{R}^+$. Velja

- $f = O(g)$, čee ostaja $C > 0$, da je $f(n) \leq C\, g(n)$ za vse dovolj velike $n \in \mathbb{N}$.
- $f = \Omega(g)$, čee $g = O(f)$.
- $f = \Theta(g)$, čee $f = O(g)$ in $f = \Omega(g)$.
- $f = o(g)$, čee $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = 0$.
- $f = \omega(g)$, čee $g = o(f)$.

Algoritmi. Dan je algoritem A in naj bo n velikost vhodnih podatkov. Označimo z

$T(n)-$ največje število elementarnih operacij, ki jih algoritem opravi za rešitev naloge

Čas izvajanja algoritma je potem $T(n) \cdot$ [čas elementarne operacije]. Zanima nas red velikosti

$$T = O(f)\,,\ T = \Theta(f)\,,\ldots$$

kjer je funkcija f tipično oblike $f(n) = n, n^2, n^k, n\log(n), 2^n, \ldots$

Naloge:

1. Pokaži, da je $f = \Theta(g)$ natanko tedaj, ko obstajata števili $0 < C_1 < C_2$, da za vse dovolj velike $n \in \mathbb{N}$ velja

$$C_1 g(n) \leq f(n) \leq C_2 g(n)\,.$$

2. Naj bosta $f, g : \mathbb{N} \to \mathbb{R}^+$. Pokaži, da velja

$$f = o(g) \Rightarrow f = O(g)\,.$$

Poišči primer funkcij $f, g : \mathbb{N} \to \mathbb{R}^+$, za kateri velja $f = O(g)$ in $f \neq o(g)$.

3. Naj bosta $f, g : \mathbb{N} \to \mathbb{R}^+$. Pokaži, da velja

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = c > 0 \Rightarrow f = \Theta(g)\,.$$

4. Dana naj bo funkcija $f(n) = a_k n^k + a_{k-1} n^{k-1} + \ldots + a_1 n + a_0$, kjer je $a_k > 0$. Pokaži, da je $f = O(n^k)$ in $f = \Theta(n^k)$.

5. Uredi funkcije

$$(\log n)^{100},\ n3^n \log n,\ \frac{n^2}{\log n},\ n2^n,\ 0.99^n,\ n^3,\ \sqrt{n}$$

po redu velikosti.

6. Poišči čim enostavnejšo funkcijo f, za katero je

$$\sum_{i=0}^{n} 2^i = \Theta(f)\,.$$

7. Dana naj bo funkcija $f(n) = n\log(n)$.

   (a) Poišči vsa števila $\alpha \in \mathbb{R}$, za katere je

   $$f = O(n^\alpha)\,.$$

   (b) Pokaži, da ne obstaja $\alpha \in \mathbb{R}$, za katerega bi bilo

   $$f = \Theta(n^\alpha)\,.$$

8. Dane naj bodo funkcije $f_1, f_2, \ldots, f_k : \mathbb{N} \to \mathbb{R}^+$, za katere velja $f_1 = O(f_2)$, $f_2 = O(f_3)$,$\ldots$,$f_{k-1} = O(f_k)$. Pokaži, da velja

$$(f_1 + f_2 + \ldots + f_k) = \Theta(f_k)\,.$$

9. Dokaži, oziroma ovrži trditev:

$$f = O(n) \Leftrightarrow \log \circ f = O(\log \circ g)\,.$$

10. Dokaži, oziroma ovrži trditve:

(a) $n^{\frac{1}{2}} = \Theta(n^{\frac{2}{3}})$

(b) $n! = o(n^n)$

(c) $n! = \omega(2^n)$

(d) $\sum_{i=1}^{n} i = \Omega(n)$

(e) $f + o(f) = \Theta(f)$

(f) $\lceil \log n \rceil ! = O(n^2)$

(g) $\log(n!) = \Theta(n \log n)$

11. Poišči najmanjši $\alpha > 0$ (če obstaja), za katerega velja

    (a) $\lceil \log n \rceil ! = O(n^\alpha)$

    (b) $\lceil \log(\log n) \rceil ! = O(n^\alpha)$.

12. Določi red časovne zahtevnosti algoritma za skalarno množenje vektorjev (algoritem zapiši v psevdokodi).

13. Določi red časovne zahtevnosti algoritma za množenje matrik (algoritem zapiši v psevdokodi).

14. Določi red časovne zahtevnosti algoritma za urejanje z vstavljanjem (algoritem zapiši v psevdokodi).

15. Določi red časovne zahtevnosti algoritma za urejanje z zlivanjem (algoritem zapiši v psevdokodi).

16. Oceni časovno zahtevnost algoritma, ki s pomočjo Gaussove eliminacije izračuna determinanto dane $n \times n$ matrike.

Poglavje 3

Celoštevilsko linearno programiranje

Teorija: Standardna oblika celoštevilskega linearnega programa je oblike

$$c^T x \to \max\,,$$

pri pogojih

$$\begin{aligned} Ax &\leq b \\ x &\geq 0 \\ x &\in \mathbb{Z}\,. \end{aligned}$$

Naloge:

1. Reši nalogo:

$$2x + y \to \max$$

pri pogojih

$$\begin{aligned} y &\leq 10 \\ 2x + 5y &\leq 60 \\ x + y &\leq 18 \\ 3x + y &\leq 44\,. \end{aligned}$$

in

$$x \geq 0,\ y \geq 0,\ x, y \in \mathbb{Z}\,.$$

2. The Apex Television Company has to decide on the number of 27- and 20-inch sets to be produced at one of its factories. Market research indicates that at most 40 of the 27-inch sets and 10 of the 20-inch sets can be sold per month. The maximum number of work-hours available

is 500 per month. A 27-inch set requires 20 work-hours and a 20-inch set requires 10 work-hours. Each 27-inch set sold produces a profit of 120$ and each 20-inch set produces a profit of 80$. A wholesaler has agreed to purchase all the television sets produced if the numbers do not exceed the maxima indicated by the market research.

(a) Formulate a linear programming model for this problem.

(b) Use the graphical method to solve this model.

3. Prevedi v standardno obliko celoštevilskega linearnega programa nalogo:
$$c^T x \to \max\,,$$
pri pogojih
$$\begin{aligned} Ax &\leq b \\ x &\in \mathbb{Z}\,. \end{aligned}$$

4. Prevedi v standardno obliko celoštevilskega linearnega programa nalogo:
$$c^T x \to \max\,,$$
pri pogojih
$$\begin{aligned} Ax &\leq b \\ x &\geq d \\ x &\in \mathbb{Z}\,. \end{aligned}$$

5. Prevedi v standardno obliko celoštevilskega linearnega programa nalogo:
$$c^T x \to \max\,,$$
pri pogojih
$$\begin{aligned} Ax &= b \\ x &\geq 0 \\ x &\in \mathbb{Z}\,. \end{aligned}$$

6. Prevedi v standardno obliko celoštevilskega linearnega programa nalogo:
$$c^T x \to \max\,,$$
pri pogojih
$$\begin{aligned} Ax &\leq b \\ |x| &\leq d \\ x &\in \mathbb{Z}\,. \end{aligned}$$

7. Prevedi v standardno obliko celoštevilskega linearnega programa nalogo:
$$c^T x \to \max\,,$$
pri pogojih
$$\begin{aligned} Ax &\leq b \\ x &\in \{0,1\}\,. \end{aligned}$$

8. Zapiši v standardni obliki celoštevilskega linearnega programa nalogo
$$\max_{1\leq i\leq m} |\sum_{j=1}^{n} a_{ij}x_j - b_i| \to \min$$

9. Zapiši v standardni obliki celoštevilskega linearnega programa nalogo:
$$c^T x \to \max\,,$$
pri pogojih
$$\begin{aligned} Ax &\leq b \\ x &\geq 0 \\ x_1 &\in \{0,1,4,6\}\,. \end{aligned}$$

10. Zapiši v standardni obliki celoštevilskega linearnega programa nalogo:

    V paketu je neznano število listov papirja debeline d. Z različnimi merilnimi inštrumenti izmerimo debelino paketa in dobimo meritve $\ell_1, \ell_2, \ldots, \ell_n$. Določi število listov papirja v paketu tako, da bo vsota absolutnih vrednosti merskih napak najmanjša.

11. (Kombinatorične dražbe) Dana naj bo množica objektov Z, $|Z| = n$ ter dane naj bodo ponudbe $(Z_1, b_1), \ldots, (Z_m, b_m)$, kjer je $Z_i \subseteq Z$. Naloga je izbrati množice ponudb, ki maksimizirajo dobiček (nalogo formuliraj v obliki celoštevilskega linearnega programa).

12. As the leader of an oil-exploration drilling venture, you must determine the least-cost selection of 5 out of 10 possible sites. Label the sites S_1, S_2,..., S_{10}, and the exploration costs associated with each as C_1, C_2,...,C_{10}. Regional development restrictions are such that:

    (a) Evaluating sites S_1 and S_7 will prevent you from exploring site S_8.
    (b) Evaluating site S_3 or S_4 prevents you from assessing site S_5.
    (c) Of the group S_5, S_6, S_7, S_8, only two sites may be assessed.

    Formulate an integer program to determine the minimum-cost exploration scheme that satisfies these restrictions.

Tipi problemov: Praktični problemi se pogosto prevedejo na eno od spodaj naštetih nestandardnih oblik celoštevilskega linearnega programa. Poglejmo si kako se vsaka od teh oblik preoblikuje v standardno obliko ILP.

1. Naloga:
$$c^T x \to \max\,,$$
pri pogojih
$$\begin{aligned} Ax &\leq b \\ x &\geq 0 \\ x &\in \mathbb{Z}\,, \end{aligned}$$
je standardna.

2. Naloga:
$$c^T x \to \max\,,$$
pri pogojih
$$\begin{aligned} Ax &\leq b \\ x &\in \{0,1\}\,, \end{aligned}$$
se da prevesti v standardno obliko (glej eno od nalog zgoraj, matriko A nadomestimo z razširjeno $[A\,I]^T$ in vektor b z vektorjem $[b\,1\,\ldots\,1]^T$).

3. Naloga:
$$c^T x \to \max\,,$$
pri pogojih
$$\begin{aligned} Ax &\leq b \\ x &\geq 0 \\ x &\in \mathbb{Z} \end{aligned}$$
in z dodatnimi pogoji oblike
$$f(x_1, \ldots, x_n) \in \{d_1, \ldots, d_m\}$$
se da prevesti v standardno obliko, če zadnji pogoj preoblikujemo v
$$f(x_1, \ldots, x_n) = d_1\,y_1 + \ldots + d_m\,y_m$$
kjer $y_i \in \{0,1\}$ in $\sum_{i=1}^{m} y_i = 1$.

4. Naloga: standardni ILP, kjer imamo še pogoje

$$
\begin{aligned}
f_1(x_1,\dots,x_n) &\le d_1 \\
&\dots \\
f_N(x_1,\dots,x_n) &\le d_N
\end{aligned}
$$

kjer zahtevamo, da velja le neka K-terica teh pogojev. Izberemo dovolj velik M in sistem prepišemo v

$$
\begin{aligned}
f_1(x_1,\dots,x_n) &\le d_1 + M\, y_1 \\
&\dots \\
f_N(x_1,\dots,x_n) &\le d_N + M\, y_N
\end{aligned}
$$

kjer so $y_i \in \{0,1\}$ in $\sum_{i=1}^{N} y_i = N - K$ (če je $y_i = 0$ pomeni, da zahtevamo, da i-ti pogoj velja).

5. Naloga: standardni ILP, kjer imamo še pogoje v obliki izjav, npr.

$$(f_1(x_1,\dots,x_n) \le d_1) \Rightarrow (f_2(x_1,\dots,x_n) \le d_2)\,.$$

Ta pogoj je ekvivalenten pogoju

$$(f_1(x_1,\dots,x_n) > d_1) \vee (f_2(x_1,\dots,x_n) \le d_2)\,.$$

$(A \Rightarrow B \Leftrightarrow \neg A \vee B)$

Dobimo torej dva pogoja

$$(f_1(x_1,\dots,x_n) > d_1) \text{ in } (f_2(x_1,\dots,x_n) \le d_2)\,,$$

od katerih mora biti vsaj en izpoljnjen, s čimer smo situacijo prevedli na prejšnji primer.

V splošnem bolj kompliciran pogoj lahko izrazimo v t.i. **disjunctive normal form**. Primer: če so A, B, C izjave, potem je npr. izjava $(A \vee B) \Rightarrow \neg C$ ekvivalentna izjavi

$$(\neg A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge \neg B \wedge C) \vee (\neg A \wedge B \wedge \neg C) \vee \dots$$

ter uporabimo zgoraj opisano metodo (vsaj eden od pogojev mora veljati).

6. Naloga: imamo ILP s samimi binarnimi spremenljivkami, $x_i \in \{0,1\}$, kjer dodamo še pogoje v obliki izjav. Potem lahko uvedemo nove binarne spremenljivke po spodnjem vzorcu

$$z = x \vee y \Rightarrow x \le z,\ y \le z,\ z \le x + y$$

$$z = x \wedge y \Rightarrow z \le x,\ z \le y,\ z \ge x + y - 1$$

$$z = \neg x \Rightarrow z = 1 - x$$

$$z = (x \Rightarrow y)\ \Rightarrow z \le 1 - x + y,\ z \ge 1 - x,\ z \ge y$$

$$z = (x \oplus y)(XOR)\ \Rightarrow z \le x + y,\ z \ge x - y,\ z \ge y - x,\ z \le 2 - x - y$$

7. (Fiksni stroški) Iščemo min oz. max funkcije $\sum_i f_i(x_i)$, kjer je

$$f_i(x_i) = \begin{cases} 0 & ; \quad x_i = 0 \\ a_i + b_i x_i & ; \quad x_i > 0 \end{cases}$$

Uvedemo dodatne spremenljivke $y_i \in \{0, 1\}$ ter iščemo min oz. max

$$\sum_i a_i y_i + b_i x_i$$

p.p.

$$x_i \leq M y_i$$

(za nek velik M).

Naloge:

1. (Set Cover) Dana naj bo množica X, $|X| = n$. Dane naj bodo podmnožice Y_i, cena katere je c_i. Določi najcenejše pokritje množice X.

2. Dano je omrežje cest (slika...). Na nekaterih križiščih hočemo postaviti hidrante tako, da bo vsaka cesta imela vsaj en hidrant. V standardni obliki zapiši linearni program, ki minimizira število hidrantov.

3. Reši nalogo:

Nal_12_6_.png

4. Na razpolago imame škatle volumna V in n objektov volumnov $a_1, \dots, a_n$. Poišči minimalno število škatel, tako da bodo vsi objekti zapakirani.

   Dodatno zapiši naslednje pogoje:

   (a) Objekta a_1 in a_7 ne smeta biti v isti škatli.

   (b) Če sta a_2 in a_4 v isti škatli, potem a_3 ne sme biti v tej škatli.

   (c) Če je a_2 ali a_4 v škatli, potem a_3 ne sme biti v tej škatli.

5. Imamo n kvadratov s stranicami $a_1, \dots, a_n$. Zloži dane kvadrate znotraj čim manjšega kvadrata (je mešani problem, niso samo celoštevilske spremenljivke).

   (a) Isti problem za kocke ...

   (b*) (IKEA problem) Dani so kvadri s stranicami $(a_i, b_i, c_i)_{i=1,\dots,n}$. Poišči čim manjši kvader v katerega lahko zložimo vse dane kvadre (čim manjši v smislu, da je vsota stranic čim manjša).

6. Na začetku leta študent dobi seznam seminarskih nalog $(S_1, \ldots, S_n)$ skupaj z datumi oddaje teh nalog (r_i). Za vsako seminarsko nalogo naj bo znan čas izdelave te naloge (p_i). Določi vrstni red izdelave nalog, da bo skupni čas zamud oddaj čim manjši.

   Dodatno upoštevaj še pogoje:

   (a) S_7 mora biti oddana takoj za S_2.

   (b) Če je S_4 oddana pred S_3, potem mora biti S_9 oddana pred S_7.

   (c) Kaj pa, če hočemo minimizirati maksimalni čas zamude?

7. Naloga 3, str.366 (Taha: Operational Research)

8. Tovorno letalo ima v trupu 8 razdelkov za tovor (4×2). Danih imamo 8 tovorov, z masami m_i $(i = 1, 2, \ldots, 8)$. Formuliraj problem razporeditve tovora v letalu, da bo masno središče čim bliže centra letala (minimiziraj vsoto absolutnih vrednosti razlik med x-koord. težišča in centrom ter y-koord. težišča in centrom).

   (a) Dodatno še zahtevamo, da sta tovora m_1 in m_2 v sosednjih razdelkih (skupna stena).

9. Formuliraj problem maksimalne neodvisne množice na grafu $G(V, E)$ (glej npr. `http://en.wikipedia.org/wiki/Independent_set_(graph_theory)`).

   (a) Formuliraj še problem maksimalne 'klike'.

10. 3×3 mrežo zapolni s števili $1, 2, \ldots, 9$ tako, da bo vsota po vseh vrsticah, stolpcih in obeh diagonalah enaka 15.

11. Formuliraj 'sliding puzzle' oz. '15-game' kot celoštevilski linearni program (`http://www.neos-guide.org/content/15puzzle`).

12. Formuliraj 'Rubikovo kocko' kot celoštevilski linearni program (`http://www.m-hikari.com/imf-password2009/45-48-2009/aksopIMF45-48-2009-2.pdf`).

13. (Fiksni stroški) Naloga 4. str.362 ...

14. Naloga 2. str.361 ...

Poglavje 4

Teorija odločanja

Osnovna oblika problema teorije odločanja je naslednja. Dan naj bo sistem S in možna stanja sistema $s_1, \ldots, s_n$, pri čemer poznamo verjetnosti $p_1, \ldots, p_n$, da se sistem nahaja v danem stanju $s_1, \ldots, s_n$ (tu predpostavljamo, da nimamo vpliva na to, v katerem stanju je sistem). Imamo tudi množico odločitev $a_1, \ldots, a_m$. Dane imamo še podatke (profite)

$$v_{ij} = \{\text{profit v primeru, da izberemo } a_i \text{ in je sistem v stanju } s_j\}$$

Dobimo tabelo

$$\begin{array}{c|ccc}
\bullet & s_1 & \ldots & s_n \\
\hline
a_1 & & \vdots & \\
\ldots & \ldots & v_{ij} & \ldots \\
a_m & & \vdots &
\end{array}$$

Povprečen profit, ki ga prinese odločitev a_i je enak

$$\phi_i = \sum_{j=1}^{n} v_{ij} p_j \, .$$

Izberemo odločitev a_i, ki nam prinese največji povprečen profit.

Dodatna testiranja: Denimo, da imamo na izbiro dodatna testiranja, katerih strošek je T. Za izbrano metodo testiranja naj bodo znani podatki

$$\rho_{ij} = P(\text{test pokaže } s_i \mid \text{pravo stanje je } s_j)$$

(skratka podatki o zanesljivosti testa (verjetnosti za 'false positive, false negative,...' napake)).

Iz teh podatkov lahko s pomočjo Bayesove formule (in $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$) izračunamo naslednje verjetnosti

$$T_{s_i} = P(\text{test pokaže } s_i) = \sum_{j=1}^{n} \rho_{ij} p_j$$

$$P(\text{pravo stanje je } s_i \mid \text{test pokaže } s_j) = \frac{\rho_{ji} p_i}{\sum_{i=1}^{n} \rho_{ji} p_i}$$

Sedaj lahko za vsak izid testa sestavimo tabelo • (kjer upoštevamo stroške testa) in izberemo optimalno odločitev in ustrezen pričakovani profit. Ker pa poznamo tudi verjetnosti za posamezne rezultate testa T_{s_i}, lahko sedaj še enkrat izračunamo pričakovan profit čez vse možne izide testa. To zadnje sedaj primerjamo s profitom, ki ga imamo, če testov ne delamo in na podlagi tega se odločimo, če bomo teste opravljali.

Odločitveno drevo: Določene naloge se bolj splača predstaviti v obliki odločitvenega drevesa, na liste • zabeležimo profite, posamezna razvejišča pa označimo bodisi z □ ali z ◯ s čimer nakažemo ali gre za našo odločitev ali za različna stanja sistema (v tem primeru na vejo zapišemo še ustrezno verjetnost, da se stanje pojavi (. . . pogojne verjetnosti. . .))

1. V trgovini prodajajo žemlje po $40c$. Vsakodnevno povpraševanje (po številu žemelj) je dano z naslednjimi podatki:

Št. prodanih žemelj	50	60	70	80	90	100
Verjetnost	0.1	0.15	0.3	0.2	0.15	0.1

   Pekarna trgovini proda (vnaprej) žemlje po $20c$, v primeru, da trgovini zmanjka žemelj ji pekarna dostavi le te po $30c$, presežek pa trgovina proda nazaj pekarni po $15c$. Koliko žemelj naj trgovina naroči na začetku dneva?

2. Naloga o zaposlitvi. . . zapiski.

3. Na letalu je 100 sedežev. Letalska družba se lahko odloči, da proda $100, 101, 102$ ali 103 karte, pri čemer so znane verjetnosti p_0, p_1, p_2 in p_3 koliko potnikov tik pred zdajci odpove polet. Vsak nezaseden sedež pomeni c_1 stroškov, medtem ko vsak nezadovoljen potnik pomeni c_2 stroškov.

   (a) Koliko kart naj proda letalska družba, da bodo pričakovani stroški čim manjši?

   (b) Svetovalno podjetje je pripravljeno za ceno T analizirati vse mogoče podatke (vremenske, prometne,. . .) in letalski družbi posredovati oceno za število odpovedanih letov (poznana naj bo zanesljivost testa - 10% verjetnost da se zmoti za največ enega potnika).

4. Preprodajalec slik . . .

5. Detektiv preverja resničnost informacije, katera je z verjetnostjo 0.3 resnična in z verjetnostjo 0.7 napačna. Detektiv se mora odločiti ali bo na podlagi te informacije izvedel aretacijo. Spodnja tabela podaja profite oz. sankcije glede na odločitev detektiva:

	Info. je TRUE	Info. je FALSE
Izvede aretacijo	1000	-500
Ne izvede aretacije	-700	400

   (a) Ali naj detektiv izvede aretacijo?

   (b) Ali naj za ceno 5 predhodno o tem posvetuje s svojim ovaduhom, ki ima 80% zanesljivost?

6. Določi optimalno izbiro odločitev za drevo

`Drevo11.png`

7. Imamo dva kovanca, prvi je 'pošten', drugi pa ima na obeh straneh grb.

Na slepo izberemo enega od kovancev. Potrebno je ugotoviti katerega od kovancev smo izbrali. Če uganemo dobimo 5$, sicer 5$ izgubimo.

(a) Opiši optimalno izbiro odločitev.

(b) Denimo, da lahko vržemo zastonj poskusni met. Na podlagi tega poskusnega meta je opiši strategijo, ki maksimizira dobiček.

(c) Določi maksimalno ceno, ki bi jo bil še pripravljen plačati za poskusni met.

(d) Določi maksimalni pričakovani dobiček, če imamo na voljo še en poskusni met (smiselno ga je uporabiti, če po prvem poskusnem metu pade grb).

Poglavje 5

Dinamično programiranje

1. Dana naj bo $m \times n$ matrika A. Določi pot $A_{i\varphi(i)}$, $i = 1, \ldots, m$, kjer je $\varphi(i) = \varphi(i-1) + \{-1, 0, 1\}$, tako da bo vsota $\sum_{i=1}^{m} A_{i\varphi(i)}$ maksimalna.

2. Dana naj bo $m \times n$ matrika A. Določi strnjeno podmatriko matrika A, ki ima lastnost, da je vsota njenih elementov maksimalna možna.

3. Obravnavaj primer 'Levenshtein distance'.

4. (Box stacking) Dana je množica škatel z dimenzijami

   h_i-višina

   w_i-širina

   d_i-dolžina.

   Izračunaj maksimalno višino stolpa, ki ga lahko sestaviš iz škatel, pri čemer morata biti širina in dolžina zgornje škatle manjši od širine in dolžine spodnje škatle.

5. Obravnavaj problem nahrbtnika. Maksimalna dovoljena masa je W, predmeti imajo maso w_i in vrednost r_i (izbereš lahko po več enakih predmetov).

6. Bankovci pridejo v vrednostih $c_1, \ldots, c_n$. Dano količino denarja izplačaj v minimalnem številu bankovcev.

7. Za dani niz črk poišči maksimalno dolžino palindroma (besede, ki se enako prebere v obe smeri).

8. Za dani niz črk poišči minimalno število potrebnih dodanih črk, ki dani niz spremeni v palindrom.

9. Za dani niz črk poišči minimalno število strnjenih palindromov, ki sestavljajo to besedo (Primer: $AXUBUC \to A, X, UBU, C$).

10. Dana je množica daljic, ki imajo eno krajišče na premici $y = 0$, drugo pa na premici $y = 1$. Določi maksimalno število paroma nesekajočih daljic (krajišča naj bodo med seboj različna).

11. Zgornja naloga, le da so krajišča daljic na enotski krožnici.

12. Dana naj bo $m \times n$ matrika A, kjer je $a_{ij} \in \{0, 1\}$. Poišči največjo (po številu enk) podmatriko samih enk.

13. Dani naj bosta besedi X in Y. Beseda Z je **zmes** besed X in Y, če je dobljena iz črk besed X in Y, pri čemer ohranimo vrstni red črk iz X, Y. (primer *uabvcw* je zmes *abc* in *uvw*, saj ${}_u{}^{ab}{}_v{}^{c}{}_w$).

14. (Razporejanje dela) Projekt traja n tednov, na i-ti teden potrebujemo najmanj b_i delavcev. Denimo, da na i-ti teden dela x_i delavcev. Potem nam $\phi_1(x_i)$ pove strošek povezan z delavci, ki presegajo potrebnih b_i, strošek $\phi_2(x_i)$ pa predstavlja najem dodatnih delavcev (torej je neničelen kadar je $x_i > x_{i-1}$ in ga prvi teden ni). Poišči minimalni strošek?

15. Reši:
$$y_1 y_2 \ldots y_n \to max$$
pri pogojih $y_i \geq 0$ in $y_1 + \ldots + y_n = c$.

16. Reši:
$$y_1^2 + y_2^2 + \ldots + y_n^2 \to min$$
pri pogojih $y_i > 0$ in $y_1 y_2 \ldots y_n = c$.

17. Dani so intervali $[a_i, b_i]$ z vrednostmi v_i. Izbereš lahko intervale, ki se med seboj ne sekajo. Določi maksimalno skupno vrednost, ki jo lahko dosežeš pri izbiranju intervalov.

18. Ob cesti so podane lokacije za reklamne table $\{x_1, \ldots, x_n\}$ s pripadajočimi profiti $\{r_1, \ldots, r_n\}$. Določi maksimalni skupni profit, če mora biti razdalja med reklamami vsaj M.

19. Danih je N kovancev, med katerimi je natanko eden težji od drugih (ostali so enako težki). Na voljo imaš tehtnico s katero lahko primerjaš dve masi (ne dobiš mase v absolutnem smislu, samo informacijo o tem katere a je težja). Določi minimalno število tehtanj s katerimi nedvoumno identificiraš težji kovanec.

20. Naprava je sestavljena iz 5 zaporedno vezanih sklopov. V vsak sklop lahko vežemo vzporedno poljubno število elementov istega tipa. Dane so mase m_i razpoložljivih elementov in verjetnosti p_i, da posamezen element ne deluje. Določi maksimalno zanesljivost naprave, če njena masa ne sme presegati M. (i-ti sklop je sestavljen iz elementov i-tega tipa)

21. Dano je zaporedje števil $a_1, \ldots, a_n$. Določi maksimalno dolžino CIK-CAK podzaporedja (t.j. zaporedje kjer iz $a_{i-1} \leq a_i$ sledi $a_{i+1} \leq a_i$ oz. iz $a_{i-1} \geq a_i$ sledi $a_{i+1} \geq a_i$).

22. Dane so koordinate hotelov $0 < a_1 < \ldots < a_n$. Začnemo v izhodišču. Na dan lahko naredimo pot dolžine L, če naredimo pot dolžine x plačamo $(L - x)^2$ denarja. Ustavimo se lahko samo na lokacijah kjer so hoteli. Načrtuj pot, pri kateri so stroški minimalni.

23. Dan je kos blaga dolžine L, z napakami na koordinatah $y_1, \ldots, y_{N-1}$ (naraščujoče zaporedje). Dana naj bo funkcija

$$P(x, m) = \text{ vrednost blaga dolžine x z m napakami}$$

Določi maksimalno vrednost danega kosa blaga, če predpostavimo, da ga lahko razrežemo (z rezom v napaki, se te znebimo).

24. Dano je zaporedje ničel in enk $0100100111\ldots001001$. Pregib tega zaporedja je zaporedje zavojev (slika...). Določi pregib, ki maksimizira število horizontalno sosednjih parov enic (ki niso zaporedne).

25. Dano je zaporedje črk in funkcija *dict*, ki za dano zaporedje črk w preveri ali tvorijo veljavno besedo, torej $dict(w) = 1$, če je w veljavna beseda in 0 sicer. Določi dinamični program, ki preveri ali je dano zaporedje črk sestavljeno iz veljavnih besed.

26.

Kolokvij 1
Operacijske raziskave, 17.4.2013

1. Dana je množica naravnih števil S moči $|S| = n$ in naravno število $k \in \{1, 2, \ldots, n\}$. Dan je algoritem **Alg** zapisan v psevdokodi:

   Alg(S, k)
   if $|S| = 1$ **then**
   | return edini element v S;
   else
   | $x \in S$ poljuben
   | $S^+ := \{y \in S \mid y \geq x\}$
   | $S^- := \{y \in S \mid y < x\}$
   | **if** $|S^-| < k$ **then**
   | | return **Alg**$(S^+, k - |S^-|)$;
   | **else**
   | | return **Alg**(S^-, k);
   | **end**
   end

   Ugotovi, kaj je rezultat algoritma pri danih vhodnih podatkih S in k, in oceni časovno zahtevnost algoritma (v najslabšjem primeru).

2. Danih je n praznih škatel z volumni $v_1, \ldots, v_n$. Dana so naslednja pravila pospravljanja škatel:

   - več manjših škatel lahko zložimo v večjo, dokler ne presežemo volumna večje škatle, in

   - škatle, ki ima znotraj sebe že kakšno škatlo, ne smemo več postaviti znotraj večje škatle.

   Zanima nas takšno pospravljanje škatel, da bo skupen volumen škatel, ki ostanejo, čim manjši. Formuliraj nalogo kot **celoštevilski linearni program**.

   Na primer, če imamo škatle z volumni $2, 3, 6, 8, 9$, potem je ena možna rešitev (ni nujno optimalna!), da zložimo škatli z volumnoma 2 in 3 znotraj škatle z volumnom 6 ter škatlo z volumnom 8 znotraj največje škatle. Ta razporeditev ima skupni volumen $6 + 9 = 15$.

3. Na letalu je 100 sedežev. Dane so naslednje verjetnosti za število potnikov, ki ne pridejo na vkrcanje, pri čemer je $P(i)$ verjetnost, da točno i potnikov ne pride na vkrcanje:

   $$P(0) = 0.25,\ P(1) = 0.5,\ P(2) = 0.25\,.$$

   (a) Koliko letalskih kart naj proda letalska družba, če vsak prazen sedež na letalu prinese 100\$ izgube, vsak nezadovoljen potnik, ki ne dobi sedeža, pa 200\$ izgube.

(b) Za 200$ lahko izvedemo predhodno analizo, ki nam napove število potnikov, ki jih ne bo na vkrcanju. Zanesljivost analize je podana z naslednjimi verjetnostmi,

$$[p_{ij}]_{i,j\in\{0,1,2\}} = \begin{bmatrix} 0.7 & 0.1 & 0 \\ 0.2 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.9 \end{bmatrix},$$

kjer je p_{ij} verjetnost, da analiza napove odpoved i potnikov, pri pogoju, da imamo j odpovedi. Ali naj letalska družba izvede analizo pred prodajo kart?

4. V okviru dinamičnega programiranja formuliraj naslednji problem (t.i. pravična delitev).

   Dana naj bo množica naravnih števil $\{a_1, \ldots, a_n\}$. Naš cilj je razdeliti to množico na dva dela S_1 in S_2 tako, da bo absolutna vrednost razlike med vsoto elementov v S_1 in S_2 čim manjša. Zapiši dinamični program, ki izračuna minimalno vrednost te razlike.

5.1 Rešitve:

1. Op: očitna varianta požrešne metode, kjer bi začeli z največjo škatlo in vanjo vstavili naslednjo manjšo,..., ne deluje (npr. $3, 4, 6, 8, 10$).

   Označimo z y_i, $i = 1, 2, \ldots, n$ binarne spremenljivke

   $$y_i = \begin{cases} 1 & ; \quad \text{če i-ta škatla ni znotraj nobene druge} \\ 0 & ; \quad \text{sicer} \end{cases}$$

   in $(i \neq j)$

   $$x_{ij} = \begin{cases} 1 & ; \quad \text{če je i-ta škatla znotraj j-te škatle} \\ 0 & ; \quad \text{sicer} \end{cases}$$

   Linearni program se sedaj glasi:

   $$\sum_i v_i y_i \to min$$

   pri pogojih:

   (1) j-ta škatla ima volumen v_j

   $$\forall j : \sum_{i=1}^{n} x_{ij} v_i \leq v_j y_j$$

   (2) Če $x_{ij} = 1$, potem $y_j = 1$, oz. če ima j-ta škatla znotraj sebe kakšno drugo škatlo, ostane zunaj (je že zajeto v (1))

   $$x_{ij} \leq y_j \qquad \forall i, j$$

   (3) Če $y_j = 1$, potem $x_{ij} = 0$, oz. če je j-ta škatla zunaj ne sme biti znotraj kakšne druge škatle

   $$x_{ij} \leq 1 - y_i \qquad \forall i, j$$

   (4) Vsaka škatla je največ znotraj ene škatle

   $$\forall i : \sum_{j=1}^{n} x_{ij} \leq 1 \,.$$

2. ...

3. Označimo z $S = \sum_i a_i$. Potem lahko problem formuliramo kot $0 - 1$ nahrbtnik; $x_i = 1$, če izberemo $i-$to število in 0 sicer,

   $$\sum a_i x_i \to max$$

pri pogoju $\sum a_i x_i \leq \lfloor \frac{S}{2} \rfloor$, nato se pa le prevedemo v dinamični program, kjer def.

$v(i,j)$ - max vrednost, ki jo lahko dosežemo, če izbiramo med $\{a_1, \ldots, a_i\}$ in je skupna vrednost navzgor omejena z j

pri čemer je rekurzija dana z

$$v(i,j) = max(a_i + v(i-1, j-a_i), v(i-1, j))\,.$$

Potem je $|S - 2v(n, \lfloor \frac{S}{2} \rfloor)|$ rešitev naloge.

Poglavje 6

Algoritmi na grafih

1. Za dane utežene grafe z izhodišči poišči drevesa minimalnih poti. Poišči še negativne cikle, če ti obstajajo.

   (a) ...

2. Dan naj bo utežen usmerjen graf G in izbrani naj bosta (neki dve) povezavi e in f. Določi najkrajšo pot med danima dvema točkama, pri pogoju, da če pot gre skozi povezavo e, mora nujno iti tudi skozi f.

3. Dano je cestno omrežje, kjer je za vsak odsek ceste podana verjetnost policijske kontrole. Med danima dvema točkama poišči pot na kateri se najverjetneje izogneš policijski kontroli.

4. Dan je utežen usmerjen graf G. Uteži povezav naj predstavljajo čas potovanja po posamezni povezavi. Za vsako vozlišče naj bo dan čas čakanja v tem vozlišču. Med danima vozliščema določi pot, ki minimizira celoten čas potovanja.

5. Dan je sistem neenačb $Ax \leq w$, kjer v vsaki neenačbi nastopa le razlika dveh spremenljivk, npr. $x_j - x_i \leq w_{ij}$. Sistemu priredimo graf, tako, da vsaki neenačbi $x_j - x_i \leq w_{ij}$ priredimo usmerjeno povezavo od i-tega do j-tega vozlišča z utežjo w_{ij}. Pokaži, da velja:

   Sistem $Ax \leq w$ je rešljiv natanko tedaj, ko pripadajoči graf nima negativnih ciklov.

6. Za dane utežene grafe poišči minimalna vpeta drevesa.

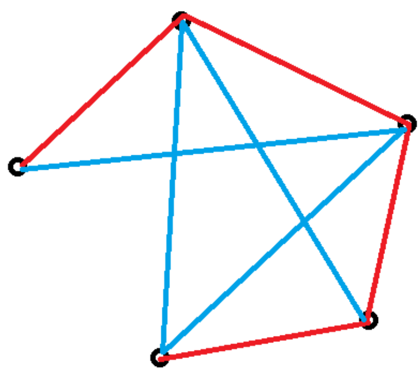
   (a) ...

7. Poišči primer pozitivno uteženega neusmerjenega grafa z lastnostjo, da se minimalno vpeto drevo razlikuje od vsakega drevesa minimalnih poti.

8. Dan naj bo strogo pozitivno utežen neusmerjen graf z minimalnim vpetim drevesom T in poljubnim drevesom najkrajših poti $\tilde{T}$. Pokaži, da imata drevesi najmanj eno skupno povezavo.

   Rešitev: Naj bo T minimalno vpeto drevo in $\tilde{T}$ drevo minimalnih poti z izhodiščem v vozlišču s. Denimo, da nimata drevesi nobene skupne povezave. Potem obstaja vozlišče q, ki je z neko povezavo $e \in T$ in $e \notin \tilde{T}$, uteži w, povezano s točko s. Označimo z γ najkrajšo pot od s do q. Pot γ je sestavljena iz vsaj dveh povezav, ker predpostavljamo da so grafi enostavni. To pa pomeni, da je $(T \cup \gamma) - e$ povezan graf z skupno utežjo manjšo ali enako uteži drevesa T. Graf $(T \cup \gamma) - e$ pa ima vsaj $|V|$ povezav, tako da moramo najmanj eno povezavo še izločiti, da dobimo drevo. Dobljeno drevo pa ima skupno utež strogo manjšo od T, ker so vse uteži strogo pozitivne.

   Opomba: Trditev ne velja več, če dopuščamo ničelne uteži. Recimo, da imamo poln graf na petih točkah, kjer so vse uteži 0. Spodnja slika prikazuje vpeti drevesi brez skupne povezave:



9. Dane naj bodo točke v ravnini. Določi minimalno vpeto drevo polnega grafa na teh točkah, če za uteži vzameš evklidske razdalje med točkami. Pokaži, da so stopnje točk v tem minimalnem vpetem drevesu ≤ 6.

10. (Bottleneck distance) V danem uteženem grafu med danima točkama poišči takšno pot, da bo vrednost minimalne uporabljene uteži čim večja. (Primerno modificiraj Dijkstrov alg.)

11. Dan naj bo utežen neusmerjen graf in povezavi e in f. Poišči minimalno vpeto drevo grafa, pri pogoju, da če je v drevesu povezava e mora biti tudi f.

12. Poišči maksimalno vpeto drevo danega grafa.

13. Zapiši celoštevilski linearni program, ki za dani graf izračuna minimalno vpeto drevo.

14. Dan naj bo utežen graf. Določi množico povezav z minimalno skupno vsoto uteži, tako da vsako vozlišče leži na vsaj eni povezavi.

15. (Clustering) Dan naj bo utežen neusmerjen graf. Vozlišča razdelimo v k disjunktnih skupin $S_1, \ldots, S_k$. Razdalja med skupinama S_i in S_j je dana z

$$d_{ij} = min(\text{dolžina najkrajše poti od } v_k \text{ do } v_k | v_k \in S_i\,,\ v_l \in S_j)\,.$$

Definiramo $d = min_{i \neq j}(d_{ij})$. Za dani graf določi razdelitev vozlišč v k disjunktnih skupin, da bo dobljeno število d maksimalno možno.

- Pokaži, da je rešitev dana z izbiro minimalnega vpetega drevesa, v katerem nato izbrišemo $(k-1)$ najdražjih povezav, kar nam da k iskanih skupin. Minimalna razdalja med skupinami pa je enaki ceni $(k-1)^{st}$ povezave (t.j. zadnje povezave, ki jo izbrišemo iz drevesa).