Laskuharjoitus 1

18.01.2001

1. Kirjan tehtävä 1.13 korjauksella: Olkoon $\mathcal{G}$ graafi, jonka solmut ovat $\{0, 1, \ldots, 9\}$ ja kaaret

   $$\{01\}\{12\}\{23\}\{34\}\{04\}\{05\}\{16\}\{27\}\{38\}\{49\}\{57\}\{79\}\{69\}\{68\}\{58\}$$

   Muodosta tämän ns. Petersenin graafin insidenssi- ja naapuruusmatriisi.

2. Petersenin graafi on ns. $(3, 5)$-häkki (cage). Yleisesti (k, n)-häkki on solmujen määrältään pienin graafi, jonka kaikkien solmujen asteluku on k ja jonka lyhimmän syklin pituus on n. (a k-regular graph of girth n)

   Todista, että $(k, 5)$-häkissä on oltava vähintään $k^2 + 1$ noodia.

3. Olkoon $d \geq 2$ kokonaisluku ja $\mathcal{V} := \{0, 1\}^d$, ts. $\mathcal{V}$ koostuu $0 - 1$ jonoista joiden pituus on d. Määritellään suuntaamaton kaarijoukko $\mathcal{E}$ siten, että solmujen $x, y \in \mathcal{V}$ välillä on kaari jos x ja y poikkeavat toisistaan täsmälleen yhdessä koordinaatissa. Määritä graafin $(\mathcal{V}, \mathcal{E})$

   (a) solmujen asteluku,

   (b) lyhimmän syklin pituus (girth) ja

   (c) pisimmän syklin pituus (circumference).

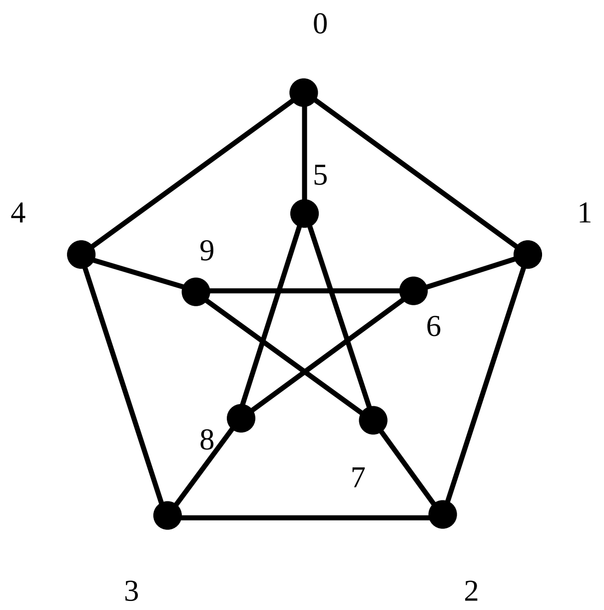
4. XY-yhtiöiden henkilöstöjohtajan tehtävänä on järjestää yrityksen virkistyspäivä. XY-yhtiöiden organisaatiohierarkia muodostaa puun, jonka solmuina on yrityksen henkilöstö ja juurena yrityksen toimitusjohtaja; esimiessuhteet esitetään kaarina. Jotta kaikki virkistyspäivän osallistujat kykenisivät rentoutumaan, vaaditaan, että kenenkään osallistujan välitön esimies ei ota osaa virkistyspäivään. Kuvaile dynaamiseen ohjelmointiin perustuva algoritmi, joka etsii virkistyspäivälle suurimmat kustannukset tuottavan osallistujalistan jos käytössä on lisäksi yhtiön henkilöstöosaston laatima arvio kunkin henkilön aiheuttamista kustannuksista.

Laskuharjoituksen 1 vastaukset

Määritellään ensin muutamia tarvittavia graafiteorian käsitteitä. Olkoon $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ mielivaltainen (suuntaamaton) graafi, ts. $\mathcal{V}$ on jokin äärellinen ei-tyhjä joukko ja $\mathcal{E} \subseteq \{\{x, y\} | x, y \in \mathcal{V}\}$.

- Solmu $x \in \mathcal{V}$ on solmun $y \in \mathcal{V}$ *naapuri* jos $\{x, y\} \in \mathcal{E}$, ts. solmujen x ja y välillä on kaari.
- Solmun $x \in \mathcal{V}$ *ympäristö* $N_{\mathcal{G}}(x) = \{y \in \mathcal{V} | \{x, y\} \in \mathcal{E}\}$ koostuu kaikista sen naapureista. Solmun *asteluku* on sen naapurien lukumäärä.
- Graafi $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ on graafin $\mathcal{G}$ *aligraafi* jos $\mathcal{V}' \subseteq \mathcal{V}$ ja $\mathcal{E}' \subseteq \mathcal{E}$.
- Solmut $x_1, x_m \in \mathcal{V}$ yhdistävä *polku* on graafin $\mathcal{G}$ muotoa $\mathcal{V}' = \{x_1, x_2, x_3, \ldots, x_m\}$, $\mathcal{E}' = \{\{x_1, x_2\}, \{x_2, x_3\}, \ldots, \{x_{m-1}, x_m\}\}$ oleva aligraafi. Polun *pituus* on sen kaarien lukumäärä. Yleensä polku esitetään luettelemalla sen solmut järjestyksessä siten, että peräkkäisten solmujen välillä on kaari, ts. $x_1 x_2 \ldots x_m$ tai $x_m x_{m-1} \ldots x_1$.
- Graafi $\mathcal{G}$ on *yhtenäinen* jos kaikille solmupareille on olemassa niitä yhdistävä polku.
- Solmut $x, y \in \mathcal{V}$ yhdistävä polku, jonka pituus on vähintään 2, muodostaa *syklin* jos siihen lisätään kaari $\{x, y\} \in \mathcal{E}$. Syklin *pituus* on sen kaarien lukumäärä (=solmujen lukumäärä). Yleensä sykli esitetään luettelemalla sen solmut siten, että peräkkäisten solmujen välillä on kaari.

1. Petersenin graafi on esitetty kuvassa 1. Graafin insidenssimatriisissa on ri-



Kuva 1: Petersenin graafi.

veillä solmut ja sarakkeissa kaaret. Rivin ja sarakkeen risteyskohdassa on 1, jos ko. solmu on ko. kaaren päätepiste.

Naapuruusmatriisissa sekä riveillä että sarakkeissa on solmu. Risteyskohdassa on 1, jos ko. solmut ovat toistensa naapureita, ts. solmuja yhdistää kaari.

	01	04	05	12	16	23	27	34	38	49	57	58	68	69	79
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0
3	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0
4	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0
5	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0
6	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0
7	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1
8	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0
9	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1

Taulukko 1: Kuvan 1 graafin insidenssimatriisi.

	0	1	2	3	4	5	6	7	8	9
0	0	1	0	0	1	1	0	0	0	0
1	1	0	1	0	0	0	1	0	0	0
2	0	1	0	1	0	0	0	1	0	0
3	0	0	1	0	1	0	0	0	1	0
4	1	0	0	1	0	0	0	0	0	1
5	1	0	0	0	0	0	0	1	1	0
6	0	1	0	0	0	0	0	0	1	1
7	0	0	1	0	0	1	0	0	0	1
8	0	0	0	1	0	1	1	0	0	0
9	0	0	0	0	1	0	1	1	0	0

Taulukko 2: Kuvan 1 graafin naapuruusmatriisi.

2. Valitaan graafin solmuista jokin, v_0. Koska graafin kaikkien solmujen asteluku on k, on polkuja, joiden pituus on 1 ja toinen pää v_0, kaikkiaan k kappaletta. Polkuja, joiden pituus on 2 ja toinen pää v_0, on $k(k-1)$ kappaletta: ensimmäisen kaaren jälkeen voidaan valita toinen kaari $k-1$ kaaresta, jotka eivät johda takaisin v_0:aan. Siten polkuja, joiden toinen päätepiste on v_0 ja pituus 1 tai 2, on kaikkiaan k^2. Jos kahdella näistä olisi yhteinen toinen päätepiste v_0:n lisäksi, syntyisi sykli, jonka pituus on korkeintaan 4. Kaikkien k^2 polun tulee siis johtaa eri solmuihin. Lisäksi graafissa on solmu v_0, joten graafissa on oltava vähintään k^2+1 solmua.

3. Graafi (V, E) on ns. d-ulotteinen (hyper)kuutio. Olkoon $x = (x_1, \dots, x_d) \in \{0,1\}^d$ jokin graafin solmu. Solmulla on d naapurisolmua, koska solmuja joiden koordinaatit poikkeavat x:n koordinaateista täsmälleen yhdessä koordinaatissa on selvästi d kappaletta.
Lyhimmän syklin pituus on 4: solmut $(0, 0, y_3, \dots, y_d)$, $(0, 1, y_3, \dots, y_d)$,

$(1, 1, y_3, \ldots, y_d)$, $(1, 0, y_3, \ldots, y_d)$ muodostavat syklin jonka pituus on 4, ja toisaalta graafi ei voi sisältää kolmiota, koska tällöin jonkin solmun y naapureihin sisältyisi kaksi solmua, jotka poikkeavat toisistaan vain yhdessä koordinaatissa. Tämä on mahdotonta, koska tällöin solmun y tulisi olla sama kuin jompi kumpi em. naapureista.
Pisimmän syklin pituus on 2^d, ts. sykli koostuu kaikista graafin solmuista. Tapauksessa $d = 2$ sykli on ilmeinen $00, 01, 11, 10$. Oletetaan, että jokin tapauksen d sykli on $\nu_1, \nu_2, \nu_3, \ldots, \nu_{2^d}$. Sykli voidaan jatkaa $(d + 1)$-ulotteisen kuution sykliksi $0\nu_1, 0\nu_2, \ldots, 0\nu_{2^d}, 1\nu_{2^d}, 1\nu_{2^d-1}, 1\nu_{2^d-2}, \ldots, 1\nu_1$. Näin esimerkiksi tapauksen $d = 3$ sykliksi tulisi em. sykliä $00, 01, 11, 10$ jatkamalla
$000, 001, 011, 010, 110, 111, 101, 100$.

4. *Puu* on yhtenäinen graafi $\mathcal{T} = (\mathcal{V}, \mathcal{E})$, jossa ei ole syklejä. Puussa jokaista solmuparia $y_1, y_2 \in \mathcal{V}$ yhdistää yksikäsitteinen polku $x_1 \ldots x_m$, jolle $y_1 = x_1$ ja $y_2 = x_m$. Asetetaan jokin solmu $r \in V$ puun juureksi, ja olkoon $n \in V$ jokin puun solmu. Tällöin voidaan määritellä r-juurellisen puun $\mathcal{T}$ *n-juurellinen alipuu* $\mathcal{T}[n]$ siten, että se koostuu kaikista niistä solmuista $x \in \mathcal{V}$, joilla solmu n kuuluu solmuja x ja r yhdistävään polkuun, ja kaikista ko. solmujen x välisistä puun $\mathcal{T}$ kaarista.

   Olkoon nyt puu $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ XY-yhtiöiden organisaatiohierarkia, ja $r \in \mathcal{V}$ yhtiön toimitusjohtaja. Funktio $\mu : \mathcal{V} \to \mathbb{R}$ määrää kunkin henkilön $x \in \mathcal{V}$ aiheuttamat virkistyspäiväkustannukset $\mu(x)$. Virkistyspäiväongelman mahdollisten ratkaisujen joukko $\mathcal{I}$ koostuu sellaisista solmujoukoista $I \subseteq \mathcal{V}$, joille minkään kahden henkilön $x, y \in I$, $x \neq y$ välillä ei ole esimiessuhdetta, ts. $\{x, y\} \notin \mathcal{E}$. Ongelman optimaalinen ratkaisu on sellainen ratkaisu $I \in \mathcal{I}$, joka maksimoi kustannukset $\sum_{x \in I} \mu(x)$.

   Dynaamiseen ohjelmointiin perustuva ratkaisualgoritmi voidaan johtaa seuraavasta havainnosta: Olkoon $I \in \mathcal{I}$ suurimmat kustannukset tuottava osallistujajoukko. Tällöin joko

   (a) *toimitusjohtaja ei osallistu virkistyspäivään*, ts. $r \notin I$: Tällöin ratkaisun I tulee tuottaa suurimmat kustannukset myös toimitusjohtajan välittömien alaisten $n \in N_{\mathcal{T}}(r)$ virittämissä n-juurellisissa alipuissa $\mathcal{T}[n]$, sillä jos näin ei ole, voidaan korkeammat kustannukset jollekin n-juurellisille alipuulle tuottava ratkaisu ottaa osaksi koko ratkaisua I siten, että kokonaiskustannukset kasvavat, jolloin I ei olisikaan optimaalinen.

   (b) *toimitusjohtaja osallistuu virkistyspäivään*, ts. $r \in I$: Tällöin ratkaisun I tulee tuottaa suurimmat kustannukset toimitusjohtajan välittömien alaisten $n \in N_{\mathcal{T}}(r)$ virittämissä n-juurellisissa alipuissa $\mathcal{T}[n]$ jos asetetaan aliratkaisulle lisäehto, että henkilö n ei saa osallistua juhliin. Jos ratkaisu I ei ole tässä ratkaisujoukossa optimaalinen, voidaan kalliimpi aliratkaisu ottaa osaksi ratkaisua I siten, että kokonaiskustannukset kasvavat ja esimiesehtoa ei rikota.

   Havaitaan siis, että optimaalisen ratkaisun tulee olla optimaalinen myös toimitusjohtajan alaisten virittämille alipuille joko (a) kaikkien aliratkaisujen joukossa, tai (b) sellaisten aliratkaisujen joukossa, jossa toimitusjohtajan välitön alainen ei osallistu virkistyspäivään. Lisäksi vastaava päättely voidaan selvästi tehdä toimitusjohtajan lisäksi tämän välittömille alaisille, näiden välittömille alaisille jne.

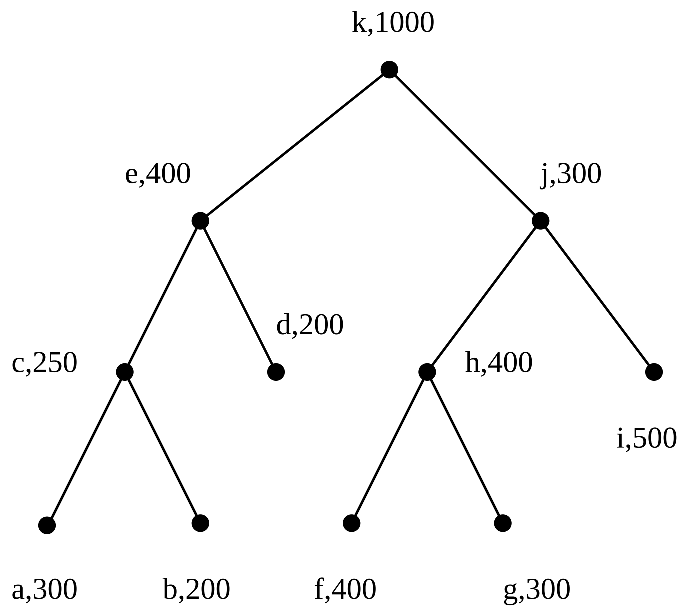
Olkoon n jokin puun $\mathcal{T}$ solmu. Merkitään $\mathcal{I}_n$:llä alipuun $\mathcal{T}[n]$ mahdollisten ratkaisujen joukkoa ja $\mathcal{I}_n^-$:lla niiden alipuun $\mathcal{T}[n]$ mahdollisten ratkaisujen joukkoa, jossa solmu n ei kuulu ratkaisuun. Merkitään lisäksi joukon $\mathcal{I}_n$ erästä (ei välttämättä yksikäsitteinen) optimaalista ratkaisua O_n:lla, ja vastaavia kustannuksia $\mu(O_n)$:lla. Vastaavasti joukolle $\mathcal{I}_n^-$ merkitään erästä optimaalista ratkaisua O_n^-:lla ja tämän kustannuksia $\mu(O_n^-)$:lla. Tällöin aikaisemman päättelyn perusteella voidaan asettaa joko $O_p := I_p^+$tai $O_p := I_p^-$, missä

$$I_p^+ = \{p\} \cup \bigcup_{n \in N_{\mathcal{T}[p]}(p)} O_n^-, \qquad I_p^- = \bigcup_{n \in N_{\mathcal{T}[p]}(p)} O_n,$$

riippuen siitä, kumpi kustannuksista $\mu(I_p^+)$, $\mu(I_p^-)$ on suurempi (yhtäsuuruuden tapauksessa valinta on vapaa). Lisäksi aina voidaan asettaa $O_p^- := I_p^-$.

Dynaamiseen ohjelmointiin perustuva ratkaisualgoritmi voi määrittää esimerkiksi syvyyshakuperiaatteella ensin osaratkaisut O_n ja O_n^- kaikille henkilön p välittömille alaisille (kaikille $n \in N_{\mathcal{T}[p]}(p)$) ja tämän jälkeen yhdistää ne joukoiksi I_p^+ ja I_p^-, joista voidaan määrittää osaratkaisut O_p ja O_p^- kuten edellä.

Kuvassa on esitetty esimerkkiyrityksen organisaatio ja virkistyskustannukset kullekin sen jäsenelle. Taulukkossa 3 on esitetty algoritmin toiminta esimerkkitapauksessa. (Esimerkkipuun solmut on nimetty syvyyshaun etenemisjärjestyksessä.) Ratkaisuksi saadaan $O_k = \{a, b, d, f, g, i, k\}$, jolle $\mu(O_k) = 2900$.



Kuva 2: Esimerkkiorganisaatio ja sen virkistyskustannukset.

Taulukko 3: Algoritmin toiminta kuvan esimerkkitapauksessa.

p	$N_{\mathcal{T}[p]}(p)$	$\mu(p)$	I_p^+	$I_p^- (= O_p^-)$	$\mu(I_p^+)$	$\mu(I_p^-)$	O_p
a	$\emptyset$	300	$\{a\}$	$\emptyset$	300	0	$\{a\}$
b	$\emptyset$	200	$\{b\}$	$\emptyset$	200	0	$\{b\}$
c	$\{a, b\}$	250	$\{c\}$	$\{a, b\}$	250	500	$\{a, b\}$
d	$\emptyset$	200	$\{d\}$	$\emptyset$	200	0	$\{d\}$
e	$\{c, d\}$	400	$\{a, b, e\}$	$\{a, b, d\}$	900	700	$\{a, b, e\}$
f	$\emptyset$	400	$\{f\}$	$\emptyset$	400	0	$\{f\}$
g	$\emptyset$	300	$\{g\}$	$\emptyset$	300	0	$\{g\}$
h	$\{f, g\}$	400	$\{h\}$	$\{f, g\}$	400	700	$\{f, g\}$
i	$\emptyset$	500	$\{i\}$	$\emptyset$	500	0	$\{i\}$
j	$\{h, i\}$	300	$\{f, g, j\}$	$\{f, g, i\}$	1000	1200	$\{f, g, i\}$
k	$\{e, j\}$	1000	$\{a, b, d, f, g, i, k\}$	$\{a, b, e, f, g, i\}$	2900	2100	$\{a, b, d, f, g, i, k\}$