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).

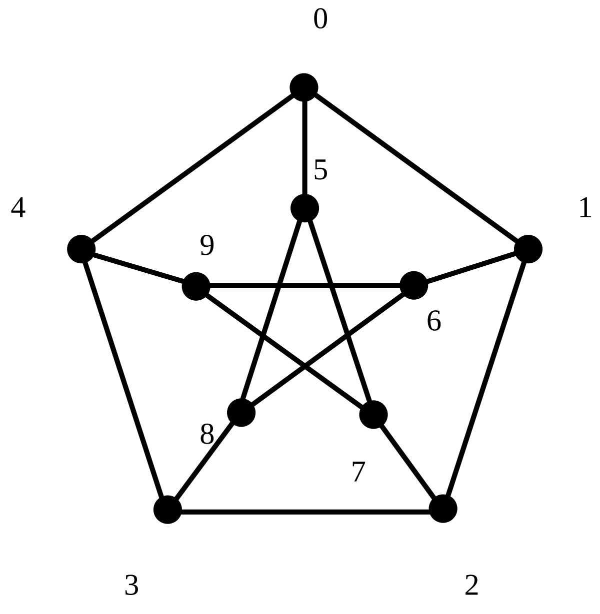
4. XY-yhtiöiden henkilöstöjohtajan tehtävänä on järjestää yrityksen virkistyspäivä. XY-yhtiöiden organisaatiohierarkia muodostaa puun, jonka solmuina on yrityksen henkilöstö ja juurena yrityksen toimitusjohtaja; esimiessuhteet esitetään kaarina. Jotta kaikki virkistyspäivän osallistujat kykenisivät rentoutumaan, 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 $x = (x_1, \ldots, x_d) \in \{0,1\}^d$ 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 $(0, 0, y_3, \ldots, y_d)$, $(0, 1, y_3, \ldots, y_d)$,

$(1, 1, y_3, \ldots, y_d)$, $(1, 0, y_3, \ldots, y_d)$ 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.

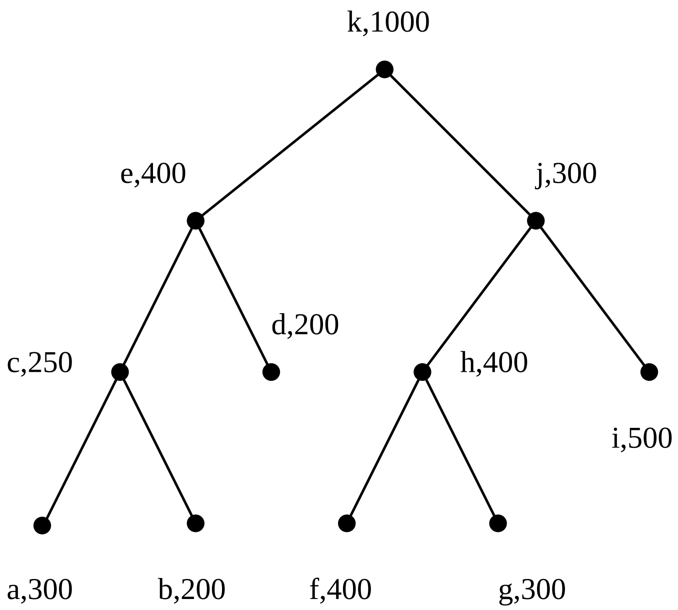
Olkoon n jokin puun $\mathcal{T}$ solmu. Merkitään $\mathcal{I}_n$:llä alipuun $\mathcal{T}[n]$ mahdollisten ratkaisujen joukkoa ja $\mathcal{I}_n^-$:lla niiden alipuun $\mathcal{T}[n]$ mahdollisten ratkaisujen joukkoa, jossa solmu n ei kuulu ratkaisuun. Merkitään lisäksi joukon $\mathcal{I}_n$ erästä (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. Taulukossa 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	$\{a,b,d,f,g,i,k\}$

Kotitehtävä 1

24.1.2001

Tehtävä: Kuinka monta erilaista Gray-koodia on olemassa 4-bittisten binäärivektoreiden joukossa? Oletetaan, että koodin koodisanat esitetään järjestettynä jonona siten, että ensimmäinen koodisana on aina 0000, ja peräkkäiset koodisanat poikkevat toisistaan täsmälleen yhdessä koordinaatissa. Koodit katsotaan erilaisiksi jos niitä esittävät jonot poikkeavat toisistaan.

Tehtävän voi ratkaista joko käsin tai tietokoneen avulla; jälkimmäisessä tapauksessa vaaditaan ohjelmalistauksen lisäksi lyhyt selostus ohjelman toiminnasta.

Erityisen ansiokkaista ratkaisuista voidaan antaa bonuspiste tenttiin. Bonuspisteen tavoittelemiseksi voit pohtia esimerkiksi joitakin seuraavista:

- Mikä on leksikografisessa järjestyksessä ensimmäinen 4-bittinen Gray-koodi?
- Miten ongelman symmetrioita voitaisiin käyttää hyväksi laskenta-ajan vähentämiseksi?
- Montako koodia saadaan, jos vaaditaan lisäksi, että ensimmäinen ja viimeinen koodisana eroavat täsmälleen yhdessä koordinaatissa?
- Entä, kun $n = 5$?

Palautus: Palautus assistentille (Petteri Kaski, Petteri.Kaski@hut.fi, TB358). Kotitehtävän voi palauttaa myös sähköpostitse esim. tekstinä tai hyvälaatuisena Postscriptinä. Viimeinen palautuspäivä on 15.02.2001.

Laskuharjoitus 2

25.01.2001

1. Kirjan tehtävä 2.4: Etsi vektorin 01010110 successor ja rank Gray-koodissa G^8
2. Mikä on 1000. kolmen alkion joukko colex-järjestyksessä? (vrt. kirjan tehtävä 2.5).
3. Olkoon $S = \{2, 3, 5, 7, 11, 13\}$. Mikä on osajoukon $\{3, 7, 13\}$ sijaluku joukon S kolmialkioisten osajoukkojen lex-järjestyksessä?
4. Kirjan tehtävä 2.9: Olkoon $1 \leq k \leq n$. Poistetaan peilatusta binääri-Gray-koodista (binary reflected Gray code) G^n kaikki koodisanat joiden Hamming-paino ei ole k. Osoita, että jäljelle jäävät koodisanat muodostavat n-joukon k-osajoukkojen minimimuutosjärjestyksen.

Laskuharjoituksen 2 vastaukset

1. Kirjoitetaan $r = (b_8 b_7 \dots b_0)_2$ ja $G_r^n = 01010110 = (a_7 a_6 \dots a_0)_2$. Kirjan lemman 2.3 mukaan

$$a_j \equiv (b_j + b_{j+1}) \bmod 2$$

ja

$$b_j \equiv \sum_{i=j}^{n-1} a_i \bmod 2.$$

Tästä laskemalla saadaan, että $\text{rank}\,(01010110) = 1100100_2 = 100$. Seuraajan rank on $1100101_2 = 101$, ja $\text{unrank}\,(1100101_2) = 01010111$.
(Ks. myös kirjan algoritmi 2.3.)

2. Ks. kirjan algoritmi 2.10. Jos ensimmäisen osajoukon rank on 0, on 1000. osajoukon rank 999. Asetetaan $r \leftarrow 999$. Haetaan ensin suurin x, jolle $\binom{x}{k} \leq r$, eli tässä tapauksessa 19. Kaikissa $\binom{19}{3} = 969$ ensimmäisessä osajoukossa suurin luku on korkeintaan 19. $\binom{20}{3} = 1140$, joten 20 on eräs haettavan kolmen alkion osajoukon alkio. Uusi arvo r:lle: $r \leftarrow 999 - \binom{19}{3} = 30$. Suurin x jolle $\binom{x}{2} \leq 30$ on 8, joten toinen osajoukon alkio on $8 + 1 = 9$. Uusi arvo r:lle: $r \leftarrow 30 - \binom{8}{2} = 2$, joten viimeinen osajoukon alkio on $2 + 1 = 3$. Kolmen alkion osajoukko, jonka rank on 999, on siis $\{3, 9, 20\}$.

3. Oletetaan joukon S alkiot järjestetyksi kuten tavallisesti. Tehtävä voidaan ratkaista muuntamalla ongelma joukon $\{1, 2, 3, 4, 5, 6\}$ vastaavaksi ongelmaksi ja käyttämällä tälle joukolle soveltuvaa kirjan algoritmia 2.8, tai kirjan lausetta 2.4 ja algoritmia 2.9. Muunnos voidaan tehdä kuvauksella
$f : S \to \{1, 2, 3, 4, 5, 6\}$, joka säilyttää järjestyksen, ts. kaikilla $x, y \in S$ pätee $x < y$ jos ja vain jos $f(x) < f(y)$; ainoa tällainen kuvaus saadaan asettamalla $f(2) = 1$, $f(3) = 2$, $f(5) = 3$, $f(7) = 4$, $f(11) = 5$, $f(13) = 6$. Kuvauksen f avulla joukko $\{3, 7, 13\} \subset S$ muuntuu joukoksi $f(\{3, 7, 13\}) = \{2, 4, 6\}$. (Järjestyksen säilyvyysominaisuuden avulla voidaan osoittaa, että joukon $\{2, 4, 6\}$ lex-rank joukon $\{1, 2, 3, 4, 5, 6\}$ 3-osajoukkojen joukossa vastaa joukon $\{3, 7, 13\}$ lex-rank'ia joukon S 3-osajoukkojen joukossa.) Määritetään joukon $\{2, 4, 6\}$ lex-rank kirjan lauseen 2.4 ja algoritmin 2.9 (colex-rank) avulla. Lauseen 2.4 mukaan joukon $\{1, \dots, n\}$ k-osajoukon T lex-rank voidaan määrittää k-osajoukon $T' = \{n + 1 - i : i \in T\}$ colex-rankin avulla seuraavasti:

$$\text{rank}_{\text{lex}}(T) = \binom{n}{k} - 1 - \text{rank}_{\text{colex}}(T').$$

Nyt $n = 6$ ja $k = 3$, $T = \{2, 4, 6\}$, $T = \{6+1-2, 6+1-4, 6+1-6\} = \{5, 3, 1\}$. Kirjan algoritmin 2.9 avulla saadaan

$$\mathrm{rank}_{\mathrm{colex}}(\{5,3,1\}) = \binom{5-1}{3} + \binom{3-1}{2} + \binom{1-1}{1} = 4 + 1 + 0 = 5,$$

joten lauseen 2.4 perusteella

$$\mathrm{rank}_{\mathrm{lex}}(\{2,4,6\}) = \binom{6}{3} - 1 - 5 = 20 - 1 - 5 = 14.$$

4. Merkitään n-mittaisten peilatun binääri-Gray-koodin koodisanojen jonoa G^n:llä, ja vastaavaa käänteisessä järjestyksessä listattua jonoa $\overline{\mathsf{G}^n}$:llä. Nyt $(n+1)$-mittaisista koodisanoista koostuvaa peilattua binääri-Gray-koodia voidaan merkitä $\mathsf{G}^{n+1} = (0\mathsf{G}^n, 1\overline{\mathsf{G}^n})$.

   Koska G^n koostuu kaikista 2^n binäärikoodisanasta joiden pituus on n, on selvää, että G^n sisältää myös kaikki ne $\binom{n}{k}$ binäärikoodisanaa, joiden pituus on n ja joissa on täsmälleen k ykkösbittiä, ts., joiden Hamming-paino on k. Merkitään jonon G^n alijonoa, joka koostuu k-painoisista koodisanoista G^n_k:lla.

   Jono $\mathsf{G}^n_k = (x_1, \ldots, x_{\binom{n}{k}})$ on k-osajoukkojen minimimuutosjärjestys jos jonon peräkkäisten koodisanojen välinen Hamming-etäisyys (niiden koordinaattien lukumäärä, jossa koodisanat poikkeavat toisistaan) on aina 2, ts. $d_H(x_j, x_{j+1}) = 2$ kaikilla $j = 1, \ldots, \binom{n}{k} - 1$.

   Väite. Olkoon $n \geq 1$. Tällöin kaikilla $1 \leq k \leq n$ jono G^n_k on n-joukon k-osajoukkojen minimimuutosjärjestys.

   Todistus. Tapauksissa $k = n$ väite on selvästi tosi, koska jonossa G^n_n on vain yksi koodisana. Tapauksessa $k = 1$ jono G^n_1 koostuu koodisanoista $(10\cdots0, 010\cdots0, 0010\cdots0, \ldots, 0\cdots01)$ (ei tässä järjestyksessä), joita on n kpl. On selvää, että minkä tahansa kahden tätä muotoa olevan koodisanan Hamming-etäisyys on 2.

   Jatketaan induktiolla n:n suhteen. Tapaukset $n = 1, 2$ ovat tosia edellisten erikoistapauksien nojalla ja muodostavat siten induktion perustapauksen. Oletetaan, että väite pätee tapauksessa n ja kaikilla $1 \leq k \leq n$. Tarkastellaan jonoa G^{n+1} ja valitaan jokin k, $(2 \leq k \leq n)$. (Tapaukset $k = 1$ ja $k = n+1$ ovat tosia aikaisemman perusteella.) Jonon G^{n+1} määritelmän $\mathsf{G}^{n+1} = (0\mathsf{G}^n, 1\overline{\mathsf{G}^n})$ perusteella sen kaikkien k-painoisten koodisanojen alijono $\mathsf{G}^{n+1}_k = (x_1, \ldots, x_{\binom{n+1}{k}})$ voidaan jakaa peräkkäisiin alijonoihin $(x_1, \ldots, x_m)$ ja $(x_{m+1}, \ldots, x_{\binom{n+1}{k}})$ siten, että jonossa $(x_1, \ldots, x_m)$ ovat kaikki sellaiset koodisanat, joiden ensimmäinen koordinaatti on 0 ja jälkimmäisessä jonossa kaikki sellaiset koodisanat joiden ensimmäinen koordinaatti on 1. Jono $(x_1, \ldots, x_m)$ on siis jonon $0\mathsf{G}^n$ alijono, joka koostuu k-painoisista koodisanoista, ja $(x_{m+1}, \ldots, x_{\binom{n+1}{k}})$ jonon $1\overline{\mathsf{G}^n}$ alijono, joka koostuu koodisanoista, joiden n:n viimeisen koordinaatin paino on $k-1$. Näin ollen on oltava $(x_1, \ldots, x_m) = 0\mathsf{G}^n_k$ ja $(x_{m+1}, \ldots, x_{\binom{n+1}{k}}) = 1\overline{\mathsf{G}^n_{k-1}}$, missä $\overline{\mathsf{G}^n_{k-1}}$ tarkoittaa jonoa G^n_{k-1} käänteisessä järjestyksessä. Induktiooletuksen perusteella saadaan siis $d_H(x_j, x_{j+1}) = 2$ kaikilla $j = 1, \ldots, m-1$ ja toisaalta $d_H(x_{j+1}, x_j) = d_H(x_j, x_{j+1}) = 2$ kaikilla $j = m+1, \ldots, \binom{n+1}{k} - 1$.

Jäljelle jää tapaus $j = m$. Koska x_m on jonon $0\mathsf{G}^n_k$ viimeinen koodisana ja toisaalta x_{m+1} on jonon $1\mathsf{G}^n_{k-1}$ viimeinen koodisana, tapauksessa $j = m$ pätee $d_H(x_j, x_{j+1}) = 2$ seuraavan aputuloksen nojalla:

Lemma. Olkoon $n \geq 1$. Tällöin jonolle G^n pätee kaikilla $1 \leq k \leq n$:

(a) ensimmäisen k-painoisen koodisanan ja ensimmäisen $(k-1)$-painoisen koodisanan välinen Hamming-etäisyys on 1; ja

(b) viimeisen k-painoisen koodisanan ja viimeisen $(k-1)$-painoisen koodisanan välinen Hamming-etäisyys on 1.

Todistus. Tapauksessa $k = 1$ G^n sisältää vain yhden koodisanan, jonka paino on $k - 1 = 0$, ts. $00 \cdots 0$. On selvää, että kaikki 1-painoiset koodisanat ovat tästä koodisanasta Hamming-etäisyydellä 1. Tapaus $k = n$ on vastaavanlainen. Näin ollen jäljelle jäävät tapaukset $2 \leq k \leq n - 1$. Edetään induktiolla n:n suhteen kuten edellä; tapaukset $n = 1, 2$ ovat tosia em. erikoistapauksien perusteella. Oletetaan, että väite pätee tapauksessa n (ja kaikilla $1 \leq k \leq n$). Tarkastellaan jonoa G^{n+1} ja valitaan jokin k, $(2 \leq k \leq n)$. Jonon G^{n+1} määritelmän $\mathsf{G}^{n+1} = (0\mathsf{G}^n, 1\overline{\mathsf{G}^n})$ ja k:n valinnan perusteella huomataan, että ensimmäinen k-painoinen koodisana y_k ja $(k-1)$-painoinen koodisana y_{k-1} löytyvät osajonosta $0\mathsf{G}^n$. Koska koodisanojen y_k ja y_{k-1} ensimmäisen koordinaatin nolla ei vaikuta niiden painoon, on induktio-oletuksen perusteella $d_H(y_k, y_{k-1}) = 1$. Vastaavasti viimeinen k-painoinen koodisana z_k ja viimeinen $(k-1)$-painoinen koodisana z_{k-1} löytyvät osajonosta $1\overline{\mathsf{G}^n}$, ts., z_k vastaa jonon $1\mathsf{G}^n$ ensimmäistä k-painoista koodisanaa ja z_{k-1} ensimmäistä $(k-1)$-painoista koodisanaa. Nyt on oltava $z_k = 1z'_{k-1}$ ja $z_{k-1} = 1z'_{k-2}$, missä z'_{k-1} on jonon G^n ensimmäinen $(k-1)$-painoinen koodisana ja z'_{k-2} ensimmäinen $(k-2)$-painoinen koodisana. Jälleen voidaan käyttää induktio-oletusta (arvolla $k - 1 \geq 1$), jolloin saadaan $d_H(z'_k, z'_{k-1}) = d_H(z_k, z_{k-1}) = 1$. $\square$

Laskuharjoitus 3

01.02.2001

1. Määritä permutaation $[2, 1, 4, 6, 3, 5]$ 3 seuraajaa leksikografisessa järjestyksessä.

2. Määritä permutaation $[2, 1, 4, 6, 3, 5]$ 3 seuraajaa Trotter-Johnson minimimuutosjärjestyksessä.

3. Olkoon $n \geq 2$ kokonaisluku ja r kokonaisluku väliltä $0 \leq r \leq n! - 1$. Osoita, että r voidaan esittää yksikäsitteisesti muodossa

$$r = \sum_{i=1}^{n-1} d_i \cdot i!,$$

missä $0 \leq d_i \leq i$ kaikille $i = 1, \ldots, n - 1$.

4. Joukon $\{1, \ldots, n\}$ permutaatio on parillinen (pariton) jos se voidaan esittää parillisen (parittoman) määrän transpositioita tulona. Osoita, että parillinen ja pariton permutaatio ovat hyvin määriteltyjä käsitteitä, ts. ei ole olemassa permutaatiota, joka olisi parillinen ja pariton.

Laskuharjoituksen 3 vastaukset

1. Käytetään kirjan algoritmia 2.14 permutaatioon $\pi = [2, 1, 4, 6, 3, 5]$; tällöin $n = 6$. Olkoon $\pi[1] = 2$, $\pi[2] = 1$, $\pi[3] = 4$, jne.

   (a) Etsitään ensin i, jolle pätee
   $$\pi[i] < \pi[i+1] > \pi[i+2] > \cdots > \pi[n].$$
   Saadaan $i = 5$.

   (b) Etsitään seuraavaksi se listan $\pi[i+1], \ldots, \pi[n]$ osa, jonka alkiot ovat pienempiä kuin $\pi[i]$, ts. sellainen j, jolle $\pi[j] > \pi[i]$ ja kaikille k välillä $j < k \leq n$ pätee $\pi[k] < \pi[i]$. Saadaan $j = 6$.

   (c) Vaihdetaan alkiot $\pi[i]$ ja $\pi[j]$ keskenään; tuloksena saadaan listaesitys $[2, 1, 4, 6, 5, 3]$.

   (d) Käännetään alilistan $[\pi[i+1], \ldots, \pi[n]]$ alkioiden järjestys. Tuloksena saadaan seuraajapermutaation listaesitys $[2, 1, 4, 6, 5, 3]$.

   Permutaation $[2, 1, 4, 6, 5, 3]$ seuraaja saadaan vastaavasti: tällöin $i = 3$ ja $j = 5$, ja vaiheet (c) ja (d) tuottavat listat $[2, 1, 5, 6, 4, 3]$ ja $[2, 1, 5, 3, 4, 6]$.

   Permutaatiolle $[2, 1, 5, 3, 4, 6]$ saadaan $i = 5$, $j = 6$, ja vaiheet (c) ja (d) ovat $[2, 1, 5, 3, 6, 4]$ ja $[2, 1, 5, 3, 6, 4]$.

2. Trotter-Johnson minimimuutosjärjestyksessä permutaation $\pi = [2, 1, 4, 6, 3, 5]$ seuraaja saadaan määrittämällä permutaation $\pi' = [2, 1, 4, 3, 5]$ pariteetti. Tehdään tämä permutaation π' sykliesityksen $(1\ 2)(3\ 4)(5)$ perusteella (ks. tehtävä 4). Huomataan, että permutaatio π' koostuu kahdesta transpositiosta (ja yhdestä triviaalisyklistä) ja on siten parillinen, koska sen muodostamiseen tarvitaan parillinen määrä transpositioita. Parillisen permutaation tapauksessa alkio 6 \`\`siirtyy vasemmalle" seuraajassa (mikäli mahdollista), ts. π:n seuraajat ovat $[2, 1, 6, 4, 3, 5]$, $[2, 6, 1, 4, 3, 5]$ ja $[6, 2, 1, 4, 3, 5]$.

   Vaikka tehtävässä ei tätä vaadittu, määritetään vielä esimerkin vuoksi permutaation $[6, 2, 1, 4, 3, 5]$ seuraaja. Tässä tapauksessa (6 on vasemmassa laidassa ja π' on parillinen) seuraaja muodostetaan muuttamalla permutaatio π' seuraajakseen. Määritetään siis permutaation $\pi'' = [2, 1, 4, 3]$ pariteetti, jotta saataisiin permutaation π' seuraaja määritetyksi. Permutaation sykliesitys on $(1\ 2)(3\ 4)$, joten se on parillinen. Parillisuuden perusteella permutaation π' seuraajassa alkio 5 siirtyy vasemmalle, ja seuraaja on siten $[2, 1, 4, 5, 3]$. Näin ollen permutaation $[6, 2, 1, 4, 3, 5]$ seuraaja on $[6, 2, 1, 4, 5, 3]$.

3. (Alkuperäisessä tehtävänannossa oli pieni lipsahdus; tapaus $n = 1$ ei ole mielekäs.)

   Väite. Olkoon $n \geq 2$ kokonaisluku ja r kokonaisluku väliltä $0 \leq r \leq n! - 1$. Tällöin r voidaan esittää yksikäsitteisesti muodossa
   $$r = \sum_{i=1}^{n-1} d_i \cdot i!,$$
   missä $0 \leq d_i \leq i$ kaikille $i = 1, \ldots, n-1$.

Todistus. (yksikäsitteisyys) Oletetaan, että luvulla r on kaksi ehdot täyttävää esitystä $(d_1, \ldots, d_{n-1})$ ja $(d'_1, \ldots, d'_{n-1})$. Tällöin on selvästi

$$r - r = \sum_{i=1}^{n-1} d_i \cdot i! - \sum_{i=1}^{n-1} d'_i \cdot i! = \sum_{i=1}^{n-1} (d_i - d'_i) i! = 0.$$

Oletetaan nyt, että esitykset poikkeavat toisistaan. Tällöin joukko $\{i \in \{1, \ldots n-1\} : d_i \neq d'_i\}$ ei ole tyhjä, ja siten sillä on suurin alkio k. Nyt nähdään suoraan, että ei voi olla $k = 1$, koska tällöin olisi $r - r = d_1 - d'_1 \neq 0$, mikä on ristiriita. Oletetaan siis $k > 1$. Lisäksi yleisyyttä rajoittamatta voidaan olettaa $d_k > d'_k$. Nyt k:n määritelmän perusteella saadaan

$$r - r = \sum_{i=1}^{k} (d_i - d'_i) i! = (d_k - d'_k) k! + \sum_{i=1}^{k-1} (d_i - d'_i) i!.$$

Lukujen d_i ja d'_i rajoituksien perusteella on oltava $-i \leq d_i - d'_i \leq i$, ts. $|d_i - d'_i| \leq i$. Kolmioepäyhtälön $|x + y| \leq |x| + |y|$ ja edellisen avulla saadaan

$$|\sum_{i=1}^{k-1} (d_i - d'_i) i!| \leq \sum_{i=1}^{k-1} |d_i - d'_i| i! \leq \sum_{i=1}^{k-1} i \cdot i!.$$

Epäyhtälön oikealle puolelle voidaan todistaa induktiolla (ei kuitenkaan tässä; todistus on suoraviivainen), että kaikilla $k \geq 2$ pätee $\sum_{i=1}^{k-1} i \cdot i! = k! - 1$. Näin ollen saadaan kolmioepäyhtälön $|x + y| \geq ||x| - |y||$ avulla

$$|(d_k - d'_k) k! + \sum_{i=1}^{k-1} (d_i - d'_i) i!| \geq |k! - |\sum_{i=1}^{k-1} (d_i - d'_i) i!|| \geq |k! - (k! - 1)| > 0,$$

mikä on ristiriita.

(olemassaolo) Tapauksessa $n = 2$ olemassaolo on ilmeinen. Oletetaan siis, että tapauksessa $n - 1 \geq 1$ kaikille $0 \leq r \leq (n-1)! - 1$ on olemassa jokin esitys, ja valitaan mielivaltainen kokonaisluku r väliltä $0 \leq r \leq n! - 1$. Jos $r \leq (n-1)! - 1$, voidaan selvästi valita $d_{n-1} = 0$ ja loput esityksestä saadaan oletuksen perusteella. Toisaalta jos $(n-1)! \leq r \leq n! - 1$, valitaan d_{n-1} siten, että se on joukon $\{s \in \{1, \ldots, n-1\} : r - (n-1)! s \geq 0\}$ suurin alkio (joukko on selvästi ei-tyhjä, koska $r \geq (n-1)!$). Nyt on oltava $0 \leq r - d_{n-1}(n-1)! \leq (n-1)! - 1$, sillä jos olisi $r - d_{n-1}(n-1)! \geq (n-1)!$, ei d_{n-1} olisi suurin alkio em. joukossa, koska tapauksessa $d_{n-1} = n - 1$ edellinen epäyhtälö ei ole mahdollinen ($r - (n-1)(n-1)! < n! - (n-1)(n-1)! = (n-1)!$). Näin ollen oletuksen perusteella luvulle $r - d_{n-1}(n-1)!$ on olemassa esitys, ts. $r - d_{n-1}(n-1)! = \sum_{i=1}^{n-2} d_i i!$, josta selvästi saadaan luvulle r vaaditut ehdot täyttävä esitys. □

4. Jokainen joukon $\{1, \ldots, n\}$ permutaatio π voidaan esittää järjestystä vaille yksikäsitteisesti pistevieraiden (eri sykleillä ei esiinny samaa pistettä) syklien tulona

$$\pi = (a_{1,1}\ a_{1,2}\ a_{1,3}\ \cdots\ a_{1,m_1}) \cdots (a_{k,1}\ a_{k,2}\ a_{k,3}\ \cdots\ a_{k,m_k}),$$

missä $k \geq 1$ ja $m_j \geq 1$ kaikilla $j = 1, \ldots, k$. Mielivaltainen ei-triviaali ($m_j \geq 2$) sykli $(a_{j,1} \ \cdots \ a_{j,m_j})$ voidaan selvästi esittää $m_j - 1$:n transposition avulla:

$$(a_{j,1} \ \cdots \ a_{j,m_j}) = (a_{j,1}, a_{j,2})(a_{j,2}, a_{j,3}) \cdots (a_{j,m_j-2}, a_{j,m_j-1})(a_{j,m_j-1}, a_{j,m_j}).$$

Nyt koko permutaatio π voidaan esittää $\sum_{j=1}^{k}(m_j - 1)$ transposition tulona, koska jokainen sykli voidaan esittää transpositioina kuten edellä, ja nämä esitykset yhdistää: syklit eivät vaikuta toisiinsa, koska ne ovat pistevieraita.

Määritellään nyt mielivaltaiselle joukon $\{1, \ldots, n\}$ permutaatiolle π

$$\lambda(\pi) = \sum_{j=1}^{k}(m_j - 1),$$

missä k on permutaation π pistevieraiden syklien lukumäärä, ja $m_1, \ldots, m_k$ näiden syklien pituudet. Kuvaus λ on hyvin määritelty, koska arvo $\lambda(\pi)$ ei riipu syklien esitysjärjestyksestä. Tarkastellaan seuraavaksi miten mielivaltaisen transposition (x, y), ($x, y \in \{1, \ldots, n\}$, $x \neq y$) lisääminen permutaatioon π vaikuttaa funktion λ arvoon. Tarkastelu jakaantuu kahteen tapaukseen:

(a) x ja y kuuluvat samaan permutaation π sykliin $(a_{j,1} \ \cdots \ a_{j,m_j})$, ts. $x = a_{j,p}$ ja $y = a_{j,q}$ joillakin $p < q$. (Oletetaan syklin alkiot näin numeroiduiksi.) Tällöin sykli $(a_{j,1} \ \cdots \ a_{j,m_j})$ jakaantuu permutaatiossa $(x, y)\pi$ kahdeksi pistevieraaksi sykliksi $(a_{j,1} \ \cdots \ a_{j,p-1} \ a_{j,q} \ \cdots \ a_{j,m_j})$ ja $(a_{j,p} \ \cdots \ a_{j,q-1})$; muut π:n syklit säilyvät muuttumattomina. Näin ollen $\lambda((x, y)\pi) = \lambda(\pi) - (m_j - 1) + m_j - (q - p) - 1 + q - p - 1 = \lambda(\pi) - 1$, koska ensimmäisen uuden syklin pituus on $m_j - (q - p)$ ja toisen $q - p$.

(b) x ja y kuuluvat eri sykleihin $(a_{j,1} \ \cdots \ a_{j,m_j})$ ja $(a_{l,1} \ \cdots \ a_{l,m_l})$. Oletetaan syklien alkiot numeroiduksi siten, että $x = a_{j,1}$ ja $y = a_{l,1}$. Tällöin em. syklien tilalla on permutaatiossa $(x, y)\pi$ sykli $(a_{j,1} \ \cdots \ a_{j,m_j} \ a_{l,1} \ \cdots \ a_{l,m_l})$, ja muut syklit säilyvät muuttumattomina, joten $\lambda((x, y)\pi) = \lambda(\pi) + 1$.

Yleisessä tapauksessa saadaan siis $\lambda((x, y)\pi) \equiv \lambda(\pi) + 1 \pmod 2$ riippumatta permutaatiosta π ja transpositiosta (x, y). Näin ollen jos mielivaltainen joukon $\{1, \ldots, n\}$ permutaatio τ esitetään kahdella tavalla transpositioiden $\tau = (x_1, y_1) \cdots (x_m, y_m)$ ja $\tau = (x'_1, y'_1) \cdots (x'_{m'}, y'_{m'})$ tulona, on oltava

$$m \equiv m - 1 + \lambda((x_m, y_m)) \equiv \cdots \equiv \lambda(\tau) \equiv \cdots \equiv m' \pmod 2,$$

joten molemmat m ja m' ovat joko parittomia tai parillisia.

Laskuharjoitus 4

08.02.2000

1. Määritä osituksen $1+3+3+3+7$ rsf-lex rank.

2. Kuinka monella eri tavalla voidaan 30 samanlaista palloa asettaa kuuteen samanlaiseen laatikkoon, jos vaaditaan, että missään kahdessa laatikossa ei ole samaa määrää palloja.

3. Minkä graafin Prüferin listaesitys on $(3,1,4,1,5)$?

4. Neljän termin tulo $abcd$ voidaan ryhmitellä kahden termin tuloiksi viidellä erilaisella tavalla:

$$((ab)c)d, \qquad (a(bc))d, \qquad a((bc)d), \qquad a(b(cd)), \qquad (ab)(cd).$$

   Yleisessä tapauksessa tulossa on $n+1$ termiä. Esitä bijektio erilaisten kahden termin tuloiksi ryhmittelyjen ja Catalanin perheen C_n välille. (Vihje: jokainen ryhmittely voidaan esittää binääripuuna, jonka lehdet koostuvat tulon termeistä.)

Laskuharjoituksen 4 vastaukset

1. Ositus $1 + 3 + 3 + 3 + 7$ on käänteisessä standardimuodossa (reverse standard form); standardimuoto on $7 + 3 + 3 + 3 + 1$. Rsf-lex rank on leksikografinen rank osituksien listaesityksille käänteisessä standardimuodossa. Osituksen rsf-rank voidaan laskea seuraavan rekursion (kirjan s. 76) avulla: Oletetaan, että rankattava ositus on standardimuodossa $[a_1, \ldots, a_n]$, missä $a_1 \geq \cdots \geq a_n$. Rekursion perustapauksessa $n = 1$ on aina $\mathsf{rank}([a_1]) = 0$. Tapauksessa $n > 1$ rekursioaskel on

$$\mathsf{rank}([a_1, \ldots, a_n]) = \begin{cases} \mathsf{rank}([a_1, \ldots, a_{n-1}]) & \text{jos } a_n = 1 \\ \mathsf{rank}([a'_1, \ldots, a'_n]) + P(m-1, n-1) & \text{jos } a_n > 1, \end{cases}$$

missä $a'_j = a_j - 1$ kaikilla $1 \leq j \leq n$ ja $m = \sum_{i=1}^{n} a_n$. Soveltamalla rekursioyhtälöä saadaan

$$\mathsf{rank}([7,3,3,3,1]) = \mathsf{rank}([7,3,3,3]) = \mathsf{rank}([6,2,2,2]) + P(16-1, 4-1),$$

josta edelleen

$$\mathsf{rank}([6,2,2,2]) = \mathsf{rank}([5,1,1,1]) + P(12-1, 4-1).$$

Nyt $\mathsf{rank}([5,1,1,1]) = \cdots = \mathsf{rank}([5]) = 0$, joten yhdistämällä saadaan

$$\mathsf{rank}([7,3,3,3,1]) = P(15,3) + P(11,3) = 19 + 10 = 29.$$

2. Oletetaan aluksi, että pallot saa asetella laatikoihin mielivaltaisesti. Jos kuudesta laatikosta $0 \leq k \leq 5$ jätetään tyhjäksi, pallot asetellaan jäljellä oleviin $6 - k$ laatikkoon. Numeroidaan ei-tyhjät laatikot $1, \ldots, 6-k$, ja merkitään a_j:llä laatikossa j olevien pallojen lukumäärää. Nyt on selvästi $a_j \geq 1$ kaikilla $j = 1, \ldots, 6-k$, ja toisaalta $\sum_{j=1}^{6-k} a_j = 30$. Näin ollen joukko $\{(a_1, \ldots, a_{6-k}) : a_j \geq 1 \text{ ja } \sum_{j=1}^{6-k} a_j = 30\}$ käsittää kaikki mahdolliset tavat asetella 30 samanlaista palloa *numeroituihin* laatikoihin. Koska laatikot ovat samanlaisia, lasketaan tällöin kuitenkin tietty osuus asetelmia useampaan kertaan; esimerkiksi asetelmat $(1,1,1,1,1,25)$ ja $(25,1,1,1,1,1)$ tapauksessa $k = 0$ ovat samanlaisia. Samanlaisten asetelmien kertautuminen voidaan välttää vaatimalla lisäksi $a_1 \leq \cdots \leq a_{6-k}$. Tällöin huomataan, että joukon

$$\mathcal{P}(30, 6-k) := \{(a_1, \ldots, a_{6-k}) \ : \ 1 \leq a_1 \leq \cdots \leq a_{6-k} \text{ ja } \sum_{j=1}^{6-k} a_j = 30\}$$

alkiot vastaavat itse asiassa luvun 30 erilaisia osituksia $6 - k$ osaan, joten joukossa on $P(30, 6-k)$ erilaista asetelmaa. Näin ollen erilaisia asetelmia on kaiken kaikkiaan

$$\sum_{k=0}^{5} P(30, 6-k) = P(30,6) + P(30,5) + \cdots + P(30,1) = 1206.$$

Kielletään nyt sellaiset asetelmat, joissa jossakin kahdessa laatikossa on sama määrä palloja. Tässä tapauksessa selvästi enintään yksi laatikko voidaan jättää tyhjäksi. Olkoon $0 \leq k \leq 1$ tyhjien laatikkojen lukumäärä. Jäljellä olevat

$6-k$ laatikkoa tulisi nyt täyttää siten, että missään kahdessa laatikossa ei ole samaa lukumäärää palloja. Tämä voidaan tehdä vaatimalla $1 \leq a_1 < a_2 < \cdots < a_{6-k}$ ja $\sum_{j=1}^{6-k} a_j = 30$. Näin ollen erilaisia asetelmia on yhtä monta kuin joukossa

$$\mathcal{P}'(30, 6-k) = \{(a_1, \ldots, a_{6-k}) \ : \ 1 \leq a_1 < \cdots < a_{6-k} \text{ ja } \sum_{j=1}^{6-k} a_j = 30\}$$

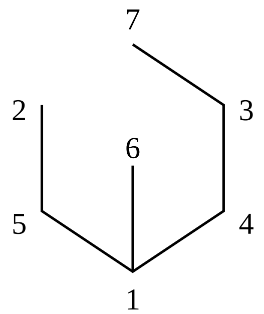
on alkioita. Nyt huomataan, että kuvaus $(a_1, \ldots, a_{6-k}) \mapsto (a_1, a_2 - 1, a_3 - 2, \ldots, a_{6-k} - (6-k-1))$ joukolta $\mathcal{P}'(30, 6-k)$ osituksien joukolle $\mathcal{P}(30 - 1 - 2 - \cdots - (6-k-1), 6-k)$ on bijektio, joten

$$|\mathcal{P}'(30, 6-k)| = P(30 - \frac{(6-k-1)(6-k)}{2}, 6-k),$$

ja näin ollen erilaisia asetelmia on $P(30-15, 6) + P(30-10, 5) = P(15, 6) + P(20, 5) = 26 + 84 = 110$ kappaletta.

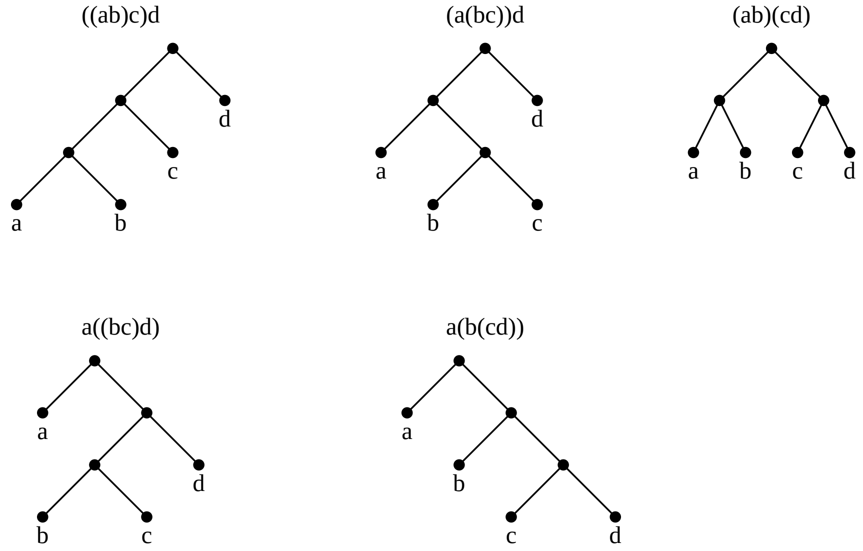
3. Seurataan kirjan algoritmia 3.11. Ensinnäkin, koska listan pituus on 5, on graafin solmuja 7. Kunkin solmun asteluku on yhtä suurempi kuin sen esiintymiskertojen lukumäärä listassa. Listan jatkoksi lisätään (1). Iteroidaan algoritmin mukaisesti: kukin listan alkioista yhdistetään vuorollaan suurinumeroisimpaan solmuun, jonka senhetkinen asteluku on 1 ja vähennetään molempien solmujen astelukua yhdellä.

Solmujen asteluvut	Listan alkio	Kaari
(3,1,2,2,2,1,1)	3	{3,7}
(3,1,1,2,2,1,0)	1	{1,6}
(2,1,1,2,2,0,0)	4	{3,4}
(2,1,0,1,2,0,0)	1	{1,4}
(1,1,0,0,2,0,0)	5	{2,5}
(1,0,0,0,1,0,0)	1	{1,5}



Kuva 3: Prüferin listaesitystä $(3, 1, 4, 1, 5)$ vastaava graafi.

4. Jokaista $n+1$ termin tulon ryhmittelyä kahden alkion tuloiksi vastaa yksikäsitteisesti binääripuu, jonka haaroissa on kertolaskuoperaattori ja lehdissä



Kuva 4: Neljän termin tulon eri ryhmittelytavat ja näitä vastaavat jäsennyspuut.

tulon termit. Kuvassa 4 on esitetty kaikki neljän termin tulon eri ryhmittelytavat ja näitä vastaavat jäsennyspuut.

Täysi binääripuu (full binary tree) on juurellinen puu, jossa jokaisella solmulla on joko nolla tai kaksi lapsisolmua. Oletetaan lisäksi, että jokaisen solmun lapsisolmut on järjestetty, jotta voidaan puhua käsitteistä "vasen" ja "oikea" lapsi. Tällöin on selvää, että jokainen $n+1$ termin jäsennyspuu on täysi binääripuu, jossa on $n+1$ lehtisolmua, ja kääntäen. Induktiolla voidaan osoittaa seuraava aputulos:

Lemma. Täydessä binääripuussa, jossa on $n+1$ lehtisolmua, on täsmälleen n solmua, joilla on kaksi lapsisolmua.

Todistus. Perustapauksessa $n = 0$ väite on ilmeinen. Oletetaan nyt, että väite pätee tapauksissa $0 \leq k \leq n$, ja tarkastellaan täyttä binääripuuta, jossa on $(n+1)+1 \geq 2$ lehtisolmua. On selvää, että tällöin puun juurella tulee olla kaksi lapsisolmua. Olkoon l_1 juuren vasemman lapsen virittämän alipuun lehtisolmujen lukumäärä, ja l_2 vastaavasti juuren oikean lapsen virittämän alipuun lehtisolmujen lukumäärä. Koska juuri ei ole lehtisolmu, on oltava $n+2 = l_1 + l_2$ ja $l_1, l_2 \geq 1$, joten $l_1 \leq n+1$ ja $l_2 \leq n+1$. Näin ollen induktio-oletusta voidaan soveltaa juuren vasemman ja oikean lapsen virittämiin alipuihin, jotka selvästi ovat täysiä binääripuita. Koko puussa on siten $1 + (l_1 - 1) + (l_2 - 1) = l_1 + l_2 - 1 = n + 1$ solmua (juuri mukaanluettuna), joilla on kaksi lapsisolmua. $\square$

Aputuloksen välitön seuraus on, että täydessä binääripuussa, jossa on $n+1$ lehtisolmua, on täsmälleen $2n$ kaarta.

Catalanin perhe C_n koostuu kaikista binäärijonoista, joiden pituus on $2n$, ja jotka ovat tasapainotettuja siten, että jonolle $[a_1, \ldots, a_{2n}] \in \mathbb{Z}_2^{2n}$ pätee

(a) jonossa on täsmälleen n nollaa ja n ykköstä, ja

(b) alijonossa $[a_1, \ldots, a_i]$ on vähintään yhtä monta nollaa kuin siinä on ykköstä kaikilla $1 \leq i \leq 2n$.

Nyt on helppoa johtaa vastaavuus $(n+1)$-lehtisolmuisten täysien binääripuiden ja Catalanin perheen välille:

Olkoon annettu täysi binääripuu, jolla on $n+1$ lehtisolmua. Merkitään kaikki vasempaan lapseen johtavat kaaret nollalla, ja oikeaan lapseen johtavat kaaret ykkösellä. Käydään puu nyt läpi esijärjestyksessä (ensin solmu itse, sitten rekursiivisesti vasen lapsi, ja lopuksi rekursiivisesti oikea lapsi) siten, että tiettyyn solmuun saavuttaessa tulostetaan solmun isään johtavan kaaren numero.

On selvää, että tällöin jokainen $2n$ kaaresta tulostetaan kerran. Lisäksi tulostejono täyttää ehdon (a), koska jokaisella solmulla, jolla on vasen lapsi, on myös oikea lapsi; toisaalta ehto (b) täyttyy, koska jokaisen kaksilapsisen solmun vasempaan lapseen vievän kaaren numero tulostetaan aina ennen oikeaan lapseen vievän kaaren numeroa.

Toisaalta jos on annettu Catalanin perheeseen kuuluva binäärijono $[a_1, \dots, a_{2n}]$, voidaan sitä vastaava täysi binääripuu konstruoida seuraavasti:

(a) Alussa puu koostuu vain sen juuresta, joka on lehtisolmu. Asetetaan juuri työsolmuksi.

(b) Toistetaan järjestyksessä kaikille $i = 1, 2, \dots, 2n$:

   i. Jos $a_i = 0$, luodaan työsolmulle vasen lapsi, ja asetetaan lapsi työsolmuksi.

   ii. Jos $a_i = 1$, edetään työsolmusta puun juurta kohti ensimmäiseen sellaiseen solmuun, jolla ei ole oikeata lapsisolmua. Luodaan tälle solmulle oikea lapsisolmu, ja asetetaan se työsolmuksi.

Konstruktio on hyvin määritelty, koska tasapainoehdon (b) mukaan kohdassa (ii) solmu, jolle voidaan lisätä oikea lapsi, on aina olemassa. Toisaalta tasapainoehdosta (a) saadaan, että jokaiselle solmulle, jolle on lisätty vasen lapsi kohdan (i) mukaisesti, lisätään myös oikea lapsi kohdan (ii) mukaisesti. Konstruktion tuloksena syntyy täysi binääripuu, jossa on $2n+1$ solmua ja $2n$ kaarta. Nyt voidaan jälleen osoittaa induktiolla, että tällaisessa puussa on oltava $n+1$ lehtisolmua, ja siten se vastaa jotakin jäsennyspuuta.

Catalanin perheen C_n ja $(n+1)$-lehtisolmuisten täysien binääripuiden bijektiivinen vastaavuus voitaisiin nyt todistaa esimerkiksi toteamalla esitetyt konstruktiot toistensa käänteiskonstruktioiksi.

Laskuharjoitus 5

15.02.2001

1. Kirjan tehtävä 4.1: Määritä valintajoukot ja kuvaile peräytymisalgoritmit seuraaville ongelmille:

   (a) Etsi kaikki tavat asettaa n kuningatarta $n \times n$-shakkilaudalle siten, etteivät kuningattaret uhkaa toisiaan.

   (b) Etsi kaikki n:n pituiset itseäänvälttävät(?) kävelyt. (Lähtee tasossa origosta; n askelta, kukin pituudeltaan 1; kunkin askelen suunta on ylös, alas, vasempaan tai oikealle, eikä missään pisteessä käydä kahdesti.)

   (c) Etsi graafin $\mathcal{G}$ solmujen k-váritykset.

2. Määritellään graafin $\mathcal{G}$ kromaattinen luku

   $$\chi\left(\mathcal{G}\right) = \min\left\{k \ : \ \mathcal{G} : \text{lla on solmujen } k - \text{varitys}\right\}$$

   ja klikkiluku

   $$\omega\left(\mathcal{G}\right) = \max\left\{k \ : \ \mathcal{G} : \text{ssa on } k : \text{n kokoinen klikki}\right\}.$$

   Kirjan lemman 4.4 mukaan $\omega\left(\mathcal{G}\right) \leq \chi\left(\mathcal{G}\right)$. Osoita, että epäyhtälö voi olla myös aito: etsi graafi, jolle $\omega\left(\mathcal{G}\right) < \chi\left(\mathcal{G}\right)$.

3. Olkoon annettuna graafi $\mathcal{G}$ ja sen solmuille jokin järjestys. Ahne graafinväritysalgoritmi värittää graafin solmu kerrallaan pienimmästä suurimpaan siten, että kulloinkin väritettävän solmun väriksi valitaan pienin (oletetaan värit järjestetyiksi) väri, joka ei esiinny väritettävän solmun jo väritetyissä naapureissa.

   (a) Esitä ääretön perhe graafeja, joiden solmut on järjestetty siten, että ahne väritysalgoritmi ei tuota optimaalista väritystä. (Väritys on optimaalinen jos värien lukumäärä on $\chi(\mathcal{G})$.)

   (b) Onko kaikille graafeille olemassa sellainen solmujen järjestys, että ahne algoritmi tuottaa optimaalisen värityksen?

4. Kirjan tehtävä 4.13: Latinalainen neliö on $n \times n$-taulukko, jonka jokaisessa rivissä ja sarakkeessa kukin luvuista $1, \ldots, n$ esiintyy tasan kerran. Latinalaista neliötä kutsutaan redusoiduksi latinalaiseksi neliöksi, jos sen ensimmäisen rivin ja ensimmäisen sarakkeen alkiot esiintyvät järjestyksessä $1, 2, \ldots, n$. Kirjoita ohjelma, joka laskee peräytymishaulla redusoitujen $n \times n$ latinalaisten neliöiden lukumäärän.

Laskuharjoituksen 5 vastaukset

1. Peräytymishaut voidaan rakentaa esim. seuraavasti.

   (a) Tarkastellaan kutakin laudan riviä $1, \ldots, n$ vuorollaan. Valintajoukkona on tällöin niiden tällä rivillä olevien ruutujen joukko, joita ei vielä uhkaa laudalla oleva kuningatar. Kokeillaan kutakin näistä vuorollaan ja siirrytään seuraavalle riville.

   (b) Valintajoukko voisi olla niiden tämänhetkisen sijainnin naapuripisteiden joukko, joissa ei ole vielä käyty. Kokeillaan näistä kutakin vuorollaan ja jatketaan kustakin niistä rekursiivisesti, ellei ole jo otettu n askelta.

   (c) Naiivi algoritmi: Tarkastellaan solmuja järjestyksessä. Valintajoukkona on kunkin solmun kohdalla kaikkien värien joukko $\{1, 2, \ldots, k\}$, josta kuitenkin on poistettu ne värit, joilla on jo väritetty jokin parhaillaan tarkasteltavana olevan solmun naapuri.

   Ainakin ensimmäisessä kohdassa kannattaa itse asiassa järjestää haku niin, että kulloinkin tarkasteltavaksi riviksi valitaan se, jolla sillä hetkellä on vähiten uhkaamattomia ruutuja. Tällöin haun haarautumiskerroin lähellä puun juurta on pieni, jolloin hakusolmujen määrä pienenee, vaikka kaikki vaihtoehdot käydäänkin yhtä kattavasti läpi.

2. Viiden solmun sykli C_5 (viisikulmio) on kromaattiselta luvultaan 3, mutta sen suurimman klikin koko on 2.

3. Olkoon $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ graafi, ja $\mathcal{W} \subseteq \mathcal{V}$. Solmujoukko $\mathcal{W}$ on *riippumaton* (independent), jos minkään sen kahden solmun välillä ei ole kaarta. Graafi $\mathcal{G}$ on *kaksijakoinen* (bipartite) jos sen solmujoukko $\mathcal{V}$ voidaan jakaa kahteen riippumattomaan joukkoon $\mathcal{W}$ ja $\mathcal{V} - \mathcal{W}$.

   Kaksijakoinen graafi voidaan aina värittää kahdella värillä, koska molempiin riippumattomiin joukkoihin riittää yksi väri.

   (a) *Täydellinen* kaksijakoinen (complete bipartite) graafi sisältää kaikki mahdolliset kaaret sen kahden riippumattoman joukon välillä, ts. kaikilla $u \in \mathcal{W}$ ja kaikilla $v \in \mathcal{V} - \mathcal{W}$ pätee $\{u, v\} \in \mathcal{E}$. Konstruoidaan graafiperhe seuraavasti: Otetaan täydellinen kaksijakoinen graafi, jonka molemmissa riippumattomissa joukoissa on n solmua, ja järjestetään solmut siten, että ensimmäinen riippumaton joukko koostuu parittomista solmuista $(1, 3, \ldots, 2n - 1)$, ja toinen parillisista $(2, 4, \ldots, 2n)$. Poistetaan graafista kaaret $\{1, 2\}, \{3, 4\}, \{5, 6\}, \ldots, \{2n - 1, 2n\}$.

   Ahne algoritmi värittää graafin seuraavasti: Solmulle 1 annetaan väri 1. Solmulle 2 annetaan väri 1, koska graafissa ei ole kaarta $\{1, 2\}$. Solmulle 3 annetaan väri 2, koska graafissa on kaari $\{2, 3\}$, ja väriä 1 ei näin ollen voi käyttää. Solmulle 4 annetaan väri 2, koska kaaren $\{1, 4\}$ takia väriä 1 ei voi käyttää. Solmulle 5 annetaan väri 3, koska kaarien $\{2, 5\}$ ja $\{4, 5\}$ takia värejä 1 ja 2 ei voi käyttää. Solmulle 6 annetaan väri 3, koska kaarien $\{1, 6\}$ ja $\{3, 6\}$ takia värejä 1 ja 2 ei voi käyttää jne.

   Helposti nähdään, että ahneeseen väritykseen värejä tarvitaan lopulta n kappaletta, vaikka graafi voidaan värittää kahdella värillä.

(b) On selvää, että värityksestä riippumatta samalla värillä väritetyt graafin solmut muodostavat riippumattoman joukon. Oletetaan, että käytettävissä on optimaalinen väritys. Jos graafin solmut nyt listataan väriluokka kerrallaan (esim. ensin kaikki punaiset solmut, sitten siniset, jne.), saadaan järjestys, jolla ahne algoritmi tuottaa optimaalisen värityksen. (Ahne väritys ei välttämättä ole identtinen alkuperäisen optimaalisen värityksen kanssa, mutta värien lukumäärä on sama.)

4. Oheinen ohjelma toteuttaa tehtävän yksinkertaisella tavalla. Koska redusoidussa latinalaisessa neliössä ensimmäinen rivi ja sarake ovat määrätyt, $n \times n$-kokoisessa latinalaisessa neliössä riittää sijoittaa $(n-1)^2$ lukua, mikä siis on haun maksimisyvyys. Ohjelma ei pidä yllä konstruoitavaa latinalaista neliötä, vaan ainoastaan boolean-muuttujia `n_in_row_i` ja `n_in_col_j`, joista nähdään, joko luku n on laitettu riville i tai sarakkeeseen j.

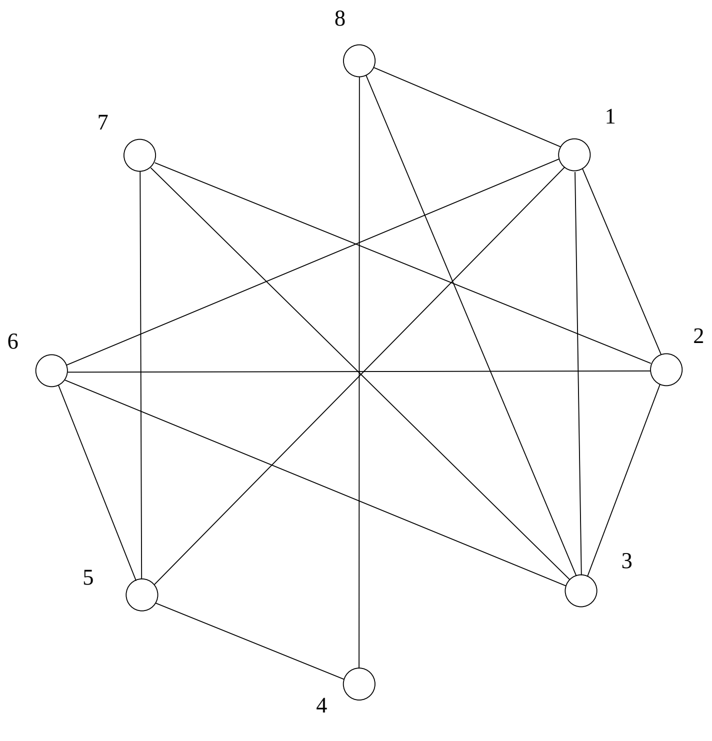
```c
/* Etsi kertalukua N olevat redusoidut latinalaiset neliöt */
#define N 4

int n_in_row_i[N+1][N+1]; /* onko numero n jo rivissä i      */
int n_in_col_j[N+1][N+1]; /* onko numero n jo sarakkeessa j */

int place(int depth) {
  int i,j,n,sum;

  if (depth==(N-1)*(N-1)) /* yksi neliö löytyi, koska on */
    return 1;             /* sijoitettu (N-1)**2 numeroa */

  i=2+depth%(N-1); /* lasketaan depthistä, mihin kohtaan */
  j=2+depth/(N-1); /* valitaan numero seuraavaksi        */

  sum=0;
  for (n=1; n<=N; n++) {
    if (!n_in_row_i[n][i] && !n_in_col_j[n][j]) {
      n_in_row_i[n][i]=1;  /* kullekin n:lle vuorollaan:          */
      n_in_col_j[n][j]=1;  /* jos n:ä ei ole laitettu riviin i    */
      sum+=place(depth+1); /* eikä sarakkeeseen j, laitetaan se   */
      n_in_row_i[n][i]=0;  /* näiden risteyskohtaan ja jatketaan */
      n_in_col_j[n][j]=0;  /* hakua; lopuksi otetaan luku pois    */
    }
  }
  return sum;
}

int main(int argc, char **argv) {
  int i,j;

  for(i=1;i<=N;i++) {        /* nollataan taulukot */
    for(j=1;j<=N;j++) {
      n_in_row_i[i][j]=n_in_col_j[i][j]=0;
    }
  }

  for(i=1;i<=N;i++) {     /* asetetaan luvut 1...n       */
    n_in_row_i[i][i]=1; /* ensimmäiseen sarakkeeseen */
    n_in_col_j[i][i]=1; /* ja riviin                   */
  }
  printf("%d\n", place(0));
  return 0;
}
```

Laskuharjoitus 6

22.02.2001

1. Hae oheisen graafin suurin klikki Carraghan–Pardalosin algoritmilla (kirjan algoritmi 4.19 rajoittavalla funktiolla 4.15).
2. Hae oheisen graafin suurin klikki (luennolla käsitellyllä) Östergårdin algoritmilla.
3. Esitä jokin mielekäs kustannusfunktio ja naapuristo seuraaville vaikeille kombinatorisille optimointiongelmille.
   (a) Symmetrinen kauppamatkustajan ongelma.
   (b) Graafin väritys n värillä siten, että naapurisolmuilla on eri väri.
   (c) Graafin ositus: Jaa graafin solmut, joita on parillinen määrä, kahteen yhtäsuureen osaan siten, että osien välisten kaarien määrä on mahdollisimman pieni.
   (d) Lukujen ositus: jaa joukko $A = \{a_1, a_2, \ldots, a_n\}$ kahteen osaan A_1 ja A_2 siten, että $A_1 \cup A_2 = A$, $A_1 \cap A_2 = \emptyset$ ja

$$\sum_{a \in A_1} a = \sum_{a \in A_2} a.$$



Kuva 5: Eräs graafi

Laskuharjoituksen 6 vastaukset

1. Algoritmissa käydään graafin noodit läpi depth-first-järjestyksessä. Noodit järjestetään aluksi johonkin järjestykseen. Kussakin hakupuun solmussa tarkastellaan vuorollaan kutakin graafin noodia, joka on kaikkien hakupolun varrella senhetkiseen klikkiin lisättyjen solmujen naapuri, ja jonka järjestysluku on suurempi kuin senhetkiseen klikkiin viimeksi lisätyn graafin noodin järjestysluku. Jos jossakin vaiheessa havaitaan, että parempaa klikkiä ei voida löytää, vaikka kaikki jäljelläolevat noodit voitaisiin lisätä klikkiin, karsitaan hakupuuta.

Tämänhetkinen klikki	Yhteiset naapurit	Huom!
-	1,2,3,4,5,6,7,8	
1	2,3,5,6,8	
1,2	3,6	
1,2,3	6	
1,2,3,6	-	{1,2,3,6}
1,2	6	karsitaan
1	3,5,6,8	
1,3	6,8	karsitaan
1	5,6,8	karsitaan
-	2,3,4,5,6,7,8	
2	3,6,7	karsitaan
-	3,4,5,6,7,8	
3	6,7,8	karsitaan
-	4,5,6,7,8	
4	5,8	karsitaan
-	5,6,7,8	karsitaan

Taulukko 4: Carraghan–Pardalosin algoritmin kulku esimerkissä

2. Östergårdin algoritmi toimii kuten Carraghan–Pardalosin algoritmi, mutta huipputasolla graafin noodeja tarjotaan klikin pienimmäksi käänteisessä järjestyksessä. Kun c(k) on suurimman sellaisen klikin koko, jonka järjestysnumeroltaan pienin alkio on k, on c(k) aina joko c(k+1) tai c(k+1)+1. Hakupuuta voidaan nyt karsia edellisen lisäksi silloin, kun tämänhetkisen klikin koko ja järjestysnumeroltaan pienimmän yhteisen naapurin c-arvo yhteensä eivät ylitä suurinta tähän asti löydettyä klikkiä.

3. Tässä tehtävässä ei ole parhaita mahdollisia ratkaisuja, vaan yleensä jotkin ratkaisut toimivat hyvin joissakin tapauksissa ja jotkin toiset ratkaisut joissain muissa tapauksissa.

   (a) Kustannusfunktiona luonnollisin valinta on reitin pituus. Eräs mahdollisuus naapuristoksi on ns. 2-OPT -siirrot: muodostetaan nykyistä reittiä kuvaavasta graafista uusi reitti poistamalla kaksi kaarta ja lisäämällä kaksi kaarta.

   (b) Tässä kuvataan ns. fixed-k lähestymistapa. Ratkaisu on mikä tahansa solmujen ositus k osaan (tyhjät osat sallitaan). Kaksi väritystä ovat tois-

Tämänhetkinen klikki	Yht. naapurit	Huom	
8	-	{8}	c(8)=1
7	-		c(7)=1
6	-		c(6)=1
5	6,7		
5,6	-	{5,6}	c(5)=2
4	5, 8		
4,5	-		
4	8	karsitaan	c(4)=2
3	6,7,8	karsitaan!	c(3)=2
2	3,6,7		
2,3	6,7		
2,3,6	-	{2,3,6}	c(2)=3
1	2,3,5,6,8		
1,2	3,6		
1,2,3	6		
1,2,3,6	-	{1,2,3,6}	c(1)=4

Taulukko 5: Östergårdin algoritmin kulku esimerkissä

tensa naapurit, jos toinen saadaan toisesta siirtämällä jokin solmu osasta toiseen. Kustannusfunktiona esim. niiden kaarien lukumäärä, joiden molemmat solmut ovat samassa osituksen osassa. Naapuristoa voidaan rajoittaa seuraavasti: nykyistä ratkaisua voidaan muuttaa vain niiden solmujen osalta, joista lähtee ainakin yksi väritysehtoa rikkova kaari; tällöin tosin naapuristo ei enää ole symmetrinen.

(c) Yksinkertainen lähestymistapa: ratkaisujoukkona on kaikki solmujen ositukset kahteen osaan. Siirto on kahden solmun vaihto osien välillä. Kustannusfunktiona kaarien määrä osien välillä.

Parempi suorituskyky on saatu simuloidulla jäähdytyksellä, kun on sallittu mielivaltaiset ositukset kahteen osaan V_1 ja V_2. Siirtoina on nyt yhden solmun siirto osasta toiseen, ja kustannusfunktiossa on lisäksi sakkona summatermi $\alpha(|V_1| - |V_2|)^2$, joka lisää kustannusta aina kun osat ovat erikokoiset. Optimointialgoritmi saattaa sakkotermistä huolimatta päätyä epätasapainoiseen ratkaisuun, jolloin voidaan joko kasvattaa α:n arvoa tai lisätä jokin yksinkertainen heuristiikka, joka muutamaa solmua siirtämällä pääsee tasapainotettuun ratkaisuun.

(d) Otetaan ratkaisujoukoksi kaikki ositukset kahteen osaan. Määritellään k-naapuristo sellaiseksi, jossa yhdellä siirrolla saadaan siirtää enintään k alkiota osituksesta toiseen. Kustannusfunktioksi voidaan ottaa

$$\min \left| \sum_{a \in V_1} a - \sum_{a \in V_2} a \right| .$$

Tämä ongelma on hiukan hankala sikäli, että siinä on pahimmillaan hyvin paljon lokaaleja optimeja. Simuloitu jäähdytys esitetyllä naapuristolla ei pärjää erityisesti tähän ongelmaan kehitetylle Karmarkar-Karp -heuristiikalle, mutta parempaa on kehitetty.

Kotitehtävä 2

Euroopan unionin yhteisvaluutta euro on ollut tilivaluuttana käytössä jo jonkin aikaa, mutta eurokolikot ja setelit tulevat käyttöön vuoden 2002 alussa. Käyttöön tulevat 1 ja 2 euron kolikot sekä 5, 10, 20, 50, 100, 200 ja 500 euron setelit.

Eräs hyvyyskriteeri rahojen nimellisarvovalikoimalle on, millaisia rahasummia voidaan maksaa siten, että vain pieni määrä rahoja (kolikoita tai seteleitä) vaihtaa omistajaa. Esimerkiksi 46 euroa voidaan maksaa 50 euron setelillä ja saada kaksi 2 euron kolikkoa takaisin. Se voidaan myös maksaa 50 euron setelillä ja 1 euron kolikolla ja saada 5 euron kolikko takaisin.

$L(k,m)$ on suurin L, jolle voidaan löytää k nimellisarvon nimellisarvovalikoima $W = \{w_1, \ldots, w_k\}$, jolla jokainen rahasumma $1, 2, 3, \ldots, L$ voidaan maksaa niin, että enintään m rahaa vaihtaa omistajaa. Toisin sanoen, $L(k,m)$ on suurin sellainen L, että löytyy nimellisarvovalikoima, jolla jokainen $1, \ldots, L$ voidaan kirjoittaa muodossa $\sum_{i=1}^{k} m_i w_i$, missä $\sum_{i=1}^{k} |m_i| \leq m$. (kaikki kokonaislukuja)

- Mikä on pienin rahasumma, jonka maksamiseen eurorahoilla ei riitä, että kolme rahaa vaihtaa omistajaa?
- Millainen nimellisarvovalikoima Euroopan unionissa olisi tullut ottaa käyttöön? Laadi (ei-triviaaliin) heuristiikkaan perustuva tietokoneohjelma, joka etsii mahdollisimman hyvän alarajan ja vastaavan nimellisarvovalikoiman $L(9,3)$:lle. Tarkastele myös muita tapauksia $L(k,3)$. Voit tutkia, löytääkö ohjelmasi $L(k,m)$:n tunnettuja arvoja $L(1,3) = 3$, $L(2,3) = 12$, $L(3,3) = 24$, $L(4,3) = 47$, $L(5,3) = 73$ ja $L(6,3) = 110$ vastaavia nimellisarvovalikoimia.

Erityisen ansiokkaista ratkaisuista voi saada bonuspisteen tenttiin. Bonuspisteen tavoittelemiseksi voi esimerkiksi esittää erinomaisen tehokkaan heuristiikan ja löytää tarvittavaan laskenta-aikaan suhteutettuna erinomaisia alarajoja $L(k,m)$:lle eri k:n ja m:n arvoilla tai vertailla ainakin kahden toisistaan poikkeavan algoritmin tehokkuutta.

Palautukseen tulee sisältyä ratkaisun sanallinen kuvaus, ohjelmien lähdekoodit ja kuvaus löydetyistä $L(k,m)$:n alarajoista vastaavine nimellisarvojoukkoineen sekä tarvitusta laskenta-ajasta.

Palautus: Palautus assistentille (Petteri Kaski, Petteri.Kaski@hut.fi, TB358). Kotitehtävän voi palauttaa myös sähköpostitse esim. tekstinä tai hyvälaatuisena Postscriptinä. Viimeinen palautuspäivä on 22.3.2000.

Vihjeitä ja ohjeita

Nimellisarvovalikoiman hyvyyttä arvioitaessa ilmeisin valinta kohdefunktioksi olisi suurin L, jolla kaikki summat $1, \ldots, L$ voidaan maksaa m rahalla. Saattaa olla tehokkaampaa muotoilla ongelma optimointiongelman sijaan hakuongelmana: kiinnitetään ensin L, ja koetetaan löytää arvovalikoima, jolla kaikki summat $1, \ldots, L$ voidaan maksaa. Tällöin kohdefunktio koostuisi vain sakkotermeistä ja se voisi olla yksinkertaisesti peittämättömien summien lukumäärä. Kohdefunktio voi toki olla aivan muunlainenkin.

Alla annetaan rutiini, joka laskee, mitkä rahasummat voidaan maksaa annetulla nimellisarvovalikoimalla. Voitte halutessanne käyttää sitä apuna kohdefunktiota muotoillessanne. Aliohjelman parametrit kuvataan seuraavassa.

arvot on taulukko, jonka alkiot $0, \ldots, k-1$ ovat nimellisarvot.

k on nimellisarvojen lukumäärä.

m on omistajaa vaihtavien rahojen suurin sallittu määrä.

i on muuttuja, jossa pidetään tähän asti kiinnitettyjen m_i-muuttujien määrää. Aliohjelmaa ulkopuolelta kutsuttaessa tämän parametrin tulee olla 0.

curr on muuttuja, jossa aliohjelma pitää tähän asti kiinnitettyjen rahojen summan. Käyttäjän kutsuessa aliohjelmaa parametrin arvon tulee olla 0.

peitetty on globaali taulukko, johon lasketaan, kuinka monella tavalla kukin summa voidaan maksaa.

maxl on peitetty-taulukon viimeisen alkion indeksi.

Aliohjelmakutsun jälkeen **peitetty[i]** kertoo, monellako tavalla i euroa voidaan maksaa käytettävissä olevalla nimellisarvovalikoimalla.

Algorithm 1 peitto.py

```
def peitto(arvot, k, m, i, curr):
    if i==0:
        x = 1
        while x <= maxl:
            peitetty[x]=0
            x=x+1
    if i==k:
        if curr>0 and curr <= maxl:
            peitetty[curr]=peitetty[curr]+1
    else:
        x = -m
        while x <= m:
            peitto(arvot, k, m-abs(x), i+1, curr+x*arvot[i])
            x=x+1
```

Laskuharjoitus 7

01.03.2001

1. Esitä joitakin sopivia tabuehtoja edellisen laskuharjoituksen optimointitehtäviin edellisellä kerralla vastauksena esitettyjen naapuristojen ja kohdefunktioiden kanssa käytettäväksi.

2. Olkoon G ei-tyhjä joukko ja $(g_1, g_2) \mapsto g_1 \cdot g_2$ ehdot (a–c) täyttävä kuvaus joukolta $G \times G$ joukolle G.

   (a) kaikilla $g_1, g_2, g_3 \in G$ pätee $(g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$.

   (b) on olemassa alkio $1 \in G$, jolle pätee $g \cdot 1 = g$ kaikilla $g \in G$.

   (c) kaikilla $g \in G$ on olemassa alkio $g^{-1} \in G$, jolle $g^{-1} \cdot g = 1$.

   Osoita, että alkio $1 \in G$ ei tällöin ole välttämättä yksikäsitteinen.

   Entä jos (b) korvataan ehdolla "on olemassa alkio $1 \in G$, jolle pätee $1 \cdot g = g$ kaikilla $g \in G$" ?

3. Olkoon G äärellinen ryhmä, H ryhmän G aliryhmä, ja K ryhmän H aliryhmä. Olkoon $\{g_1, \ldots, g_n\}$ ryhmän H vasen transversaali ryhmän G suhteen, ja olkoon $\{h_1, \ldots, h_m\}$ ryhmän K vasen transversaali ryhmän H suhteen. Osoita, että mielivaltainen $g \in G$ voidaan aina esittää muodossa $g = g_i \cdot h_j \cdot k$, missä $k \in K$ ja $1 \leq i \leq n$, $1 \leq j \leq m$.

Laskuharjoituksen 7 vastaukset

1. Kuten edellisissä laskuharjoituksissa, tässäkään esitetyt ratkaisut eivät välttämättä ole kaikissa tapauksissa parhaita mahdollisia. On syytä kokeilla erilaisia ratkaisuja ja valita se, joka toimii tarkasteltavissa ongelmainstansseissa parhaiten.

   (a) Esimerkiksi: Lisättyä kaarta ei saa poistaa seuraavalla n askeleella. Poistettua kaarta ei saa lisätä seuraavalla m askeleella. Parametrit n ja m täytyy virittää sopiviksi ongelmainstanssista riippuen.

   (b) Esimerkiksi: Jos solmun väri on muutettu esim. väristä 1 väriin 2, ei sitä saa muuttaa seuraavalla n kierroksella väriin 1. Vaihtoehtoisesti: Jos solmun v väriä on muutettu, ei solmun v väriin saa koskea seuraavalla n kierroksella. Tämä rajoittaa hakua enemmän kuin edellinen ehto.

   (c) Ensimmäisellä naapuristolla: Jos solmut v_1 ja v_2 on vaihdettu keskenään, niin tätä paria ei saa vaihtaa uudelleen seuraavan n:n kierroksen aikana. Vaihtoehtoisesti: Jos v on siirretty osituksesta toiseen, niin sitä ei saa siirtää seuraavan n kierroksen aikana.

   Toisella naapuristolla voidaan myös käyttää edellisen kohdan tabuehtoa.

   (d) Jos $k = 1$, voidaan käyttää suoraan edellisen kohdan tabuehtoja.

   Jos $k > 1$, on tilanne mutkikkaampi: voidaan käyttää samaa tabulistaa kuin edellä, mutta se saattaa rajoittaa valittavia siirtoja liikaa. Yksi mahdollisuus olisi tarkastella summaa $S_1 = \sum_{a \in A_1} a$. Jos jonkin siirron jälkeen $S_1 = x$, seuraavan n:n kierroksen ajan sellaiset siirrot, joiden jälkeen $S_1 = x$, ovat kiellettyjä. Tämä taas ei toimi silloin jos S_1:llä on vain muutamia mahdollisia arvoja.

2. Olkoon G mielivaltainen ei-tyhjä joukko ja määritellään laskutoimitus $(g_1, g_2) \mapsto g_1$. Laskutoimitus on selvästi assosiatiivinen (ehto (a)), koska

$$(g_1 \cdot g_2) \cdot g_3 = g_1 \cdot g_2 = g_1 = g_1 \cdot g_2 = g_1 \cdot (g_2 \cdot g_3).$$

Valitaan nyt alkioksi 1 mielivaltainen joukon G alkio. Ehto (b) on voimassa, koska selvästi on $g \cdot 1 = g$ kaikilla $g \in G$. Toisaalta ehto (c) on voimassa, koska voidaan asettaa $g^{-1} := 1$ kaikille $g \in G$. Näin ollen jos G:ssä on vähintään 2 alkiota ykkösalkio ei ole yksikäsitteinen.

Jos ehto (b) muutetaan muotoon “on olemassa alkio $1 \in G$, jolle pätee $1 \cdot g = g$ kaikilla $g \in G$“ voidaan ykkösalkio osoittaa yksikäsitteiseksi: Valitaan mielivaltainen $g \in G$ ja otetaan käyttöön lyhennysmerkinnät $g_1 g_2 := g_1 \cdot g_2$, $g' := g^{-1}$ ja $g'' := (g^{-1})^{-1}$. Nyt saadaan ehtojen (a–c) perusteella

$$g1 = 1(g1) = (1g)1 = ((g''g')g)1 = (g''(g'g))1 = (g''1)1 =$$

$$= g''(11) = g''1 = g''(g'g) = (g''g')g = 1g = g,$$

joten $1g = g1 = g$ kaikilla $g \in G$. Näin ollen jos on olemassa alkiot $1 \in G$ ja $1' \in G$, jotka molemmat täyttävät ehdot (b) ja (c), voidaan päätellä $1' = 11' =$

1. Kun ykkösalkio on todettu yksikäsitteiseksi, voidaan vasen käänteisalkio osoittaa myös oikeaksi käänteisalkioksi käyttämällä aksioomia (a–c):

$$gg' = g(1g') = (g1)g' = (g(g'g))g' = ((gg')g)g' = (gg')(gg'),$$

joten on oltava $gg' = 1$. Nyt käänteisalkion yksikäsitteisyys voidaan todeta seuraavasti:

$$g' = g'1 = g'(g\hat{g}') = (g'g)\hat{g}' = 1\hat{g}' = \hat{g}'$$

.

3. Jos H on äärellisen ryhmän G aliryhmä ja $\{g_1, \ldots, g_n\}$ ryhmän H vasen transversaali G, niin määritelmän mukaan joukko $\{g_1H, \ldots, g_nH\}$, missä $g_iH := \{g_ih \ : \ h \in H\}$ kaikilla $i = 1, \ldots, n$, on juokon G ositus (G:n jako ei-tyhjiin pistevieraisiin osajoukkoihin). Valitaan nyt jokin $g \in G$. Koska joukot g_iH osittavat joukon G, on olemassa yksikäsitteinen i, jolle $g \in g_iH$. Edelleen on olemassa yksikäsitteinen $h \in H$ jolle $g = g_ih$, koska jos on $g_ih = g = g_ih'$, voidaan päätellä

$$h = 1h = (g_i^{-1}g_i)h = g_i^{-1}(g_ih) = g_i^{-1}(g_ih') = (g_i^{-1}g_i)h' = 1h' = h'.$$

Näin ollen jokainen $g \in G$ voidaan esittää yksikäsitteisesti muodossa g_ih, missä $h \in H$. Vastaavalla päättelyllä voidaan nyt todeta, että em. esityksen $h \in H$ voidaan esittää yksikäsitteisesti muodossa $h = h_jk$, missä $k \in K$ ja $h_j \in \{h_1, \ldots, h_m\}$. Päättelyä voitaisiin selvästi jatkaa myös tästä eteenpäin jos K:lla olisi aliryhmä jne.

Edellisen kaltaisesta esityksestä on hyötyä, jos halutaan esimerkiksi käsitellä äärellisiä ryhmiä tietokoneella. Tällöin ryhmää voidaan käsitellä kokoelmana transversaaleja tietyn aliryhmäketjun suhteen. Esim. Schreier-Sims esityksessä (kirjan kappale 6.2.3) aliryhmäketju koostuu sisäkkäisistä pisteenstabiloija-aliryhmistä.

Laskuharjoitus 8

08.03.2001

1. Kirjan tehtävä 6.7: Tutkitaan joukon $\{0, 1, \ldots, 9\}$ permutaatioryhmää G, jonka Schreier-Sims-esitys on

$$\overrightarrow{G} = (\beta : [\mathcal{U}_0, \mathcal{U}_1, \ldots, \mathcal{U}_9]) ,$$

missä $\beta = \mathbf{I}$ ja

$$\mathcal{U}_0 = \left\{ \begin{array}{l} \mathbf{I},\ (0,1,3,6)\,(2,5,9,7)\,(4,8)\,,\ (0,2,5,9,6)\,(1,4,8,3,7)\,, \\ (0,3)\,(1,6)\,(2,9)\,(5,7)\,,\ (0,4,9,6)\,(1,2,5,8)\,(3,7)\,, \\ (0,5,6,2,9)\,(1,8,7,4,3)\,,\ (0,6,9,5,2)\,(1,7,3,8,4)\,, \\ (0,7,6)\,(1,2,8)\,(3,4,9)\,,\ (0,8,4,1,9)\,(2,5,3,6,7)\,, \\ (0,9,2,6,5)\,(1,3,4,7,8) \end{array} \right\},$$

$$\mathcal{U}_1 = \left\{ \begin{array}{l} \mathbf{I},\ (1,2)\,(3,4)\,(6,7)\,,\ (1,3,6)\,(2,4,7)\,(5,9,8)\,, \\ (1,4)\,(2,3)\,(6,7)\,(8,9)\,,\ (1,6)\,(2,7)\,(5,9)\,,\ (1,7,3,2,6,4)\,(5,8,9) \end{array} \right\},$$

$$\mathcal{U}_2 = \{\mathbf{I}\}\,, \qquad \mathcal{U}_3 = \{\mathbf{I},\ (3,6)\,(4,7)\,(5,8)\}\,,$$

$$\mathcal{U}_4 = \mathcal{U}_5 = \mathcal{U}_6 = \mathcal{U}_7 = \mathcal{U}_8 = \mathcal{U}_9 = \{\mathbf{I}\}\,.$$

Mitkä seuraavista permutaatioista kuuluvat ryhmään G?

(a) $\alpha = (0,1,2,3,4,5,6)$

(b) β=$(0,1,2,3,4)\,(5,6,7,8,9)$

(c) γ=$(0,3,5,8,7)\,(1,9,2,6,4)$

0	1	2
3	4	5
6	7	8

Kuva 6: Neliö ja 9 pikkuneliötä.

2. Neliö jaetaan 9 pikkuneliöön, joista osa väritetään mustiksi. Kaksi väritystä samaistetaan, jos toinen saadaan ensimmäisestä kiertämällä ja/tai peilaamalla koko neliötä.

(a) Kuinka monella eri tavalla voidaan värittää 5 pikkuneliötä?

(b) Anna kaikki kahdeksan erilaista 2 pikkuneliön väritystä.

Laskuharjoituksen 8 vastaukset

1. Edetään kutakuinkin kirjan algoritmin 6.8 tapaan:

   (a) Koska α kuvaa alkion 0 alkiolle 1, etsitään $h_0 \in \mathcal{U}_0$, joka kuvaa alkion 0 alkiolle 1. Sellainen löytyy: $h_0 = (0,1,3,6)\,(2,5,9,7)\,(4,8)$. Kertomalla α tämän käänteiselementillä saadaan $h_0^{-1}\alpha = (0)\,(1,7,9,5,3,8,4,2)\,(6)$. Koska tässä 1 kuvautuu 7:lle, etsitään $h_1 \in \mathcal{U}_1$, joka kuvaa 1:n 7:lle. Sellainen löytyy: $h_1 = (1,7,3,2,6,4)\,(5,8,9)$. Nyt $h_1^{-1}h_0^{-1}\alpha = (0)\,(1)\,(2,4,3,5,7,8,6)\,(9)$. Koska tässä 2 kuvautuu 4:lle, etsitään $h_2 \in \mathcal{U}_2$, joka kuvaa 2:n 4:lle. Sellaista ei löydy, joten $\alpha \notin G$.

   (b) Koska β kuvaa alkion 0 alkiolle 1, etsitään $h_0 \in \mathcal{U}_0$, joka kuvaa alkion 0 alkiolle 1. Sellainen löytyy: $h_0 = (0,1,3,6)\,(2,5,9,7)\,(4,8)$. Kertomalla β tämän käänteispermutaatiolla saadaan $h_0^{-1}\beta = (0)\,(1,7,4,6,9,2)\,(3,8,5)$. Koska tässä 1 kuvautuu 7:lle, etsitään $h_1 \in \mathcal{U}_1$, joka kuvaa 1:n 7:lle. Sellainen löytyy: $h_1 = (1,7,3,2,6,4)\,(5,8,9)$. Nyt $h_1^{-1}h_0^{-1}\beta = (0)\,(1)\,(2,4)\,(3,5,7,6,8,9)$. Koska tässä 2 kuvautuu 4:lle, etsitään $h_2 \in \mathcal{U}_2$, joka kuvaa 2:n 4:lle. Sellaista ei löydy, joten $\beta \notin G$.

   (c) Koska γ kuvaa alkion 0 alkiolle 3, etsitään $h_0 \in \mathcal{U}_0$, joka kuvaa alkion 0 alkiolle 3. Sellainen löytyy: $h_0 = (0,3)\,(1,6)\,(2,9)\,(5,7)$. Kertomalla γ tämän käänteispermutaatiolla saadaan $h_0^{-1}\gamma = (0)\,(1,2)\,(3,7)\,(4,6)\,(5,8)\,(9)$. Koska tässä 1 kuvautuu 2:lle, etsitään $h_1 \in \mathcal{U}_1$, joka kuvaa 1:n 2:lle. Sellainen löytyy: $h_1 = (1,2)\,(3,4)\,(6,7)$. Nyt $h_1^{-1}h_0^{-1}\gamma = (0)\,(1)\,(2)\,(3,6)\,(4,7)\,(5,8)\,(9)$. Tässä 2 kuvautuu 2:lle; etsitään $h_2 \in \mathcal{U}_2$, joka kuvaa 2:n 2:lle. Sellainen löytyy: $h_2 = \mathbf{I}$. Nyt $h_2^{-1}h_1^{-1}h_0^{-1}\gamma = (0)\,(1)\,(2)\,(3,6)\,(4,7)\,(5,8)\,(9)$. Koska tässä 3 kuvautuu 6:lle, etsitään $h_3 \in \mathcal{U}_3$, joka kuvaa 3:n 6:lle. Sellainen löytyy: $h_3 = (3,6)\,(4,7)\,(5,8)$. Nyt $h_3^{-1}h_2^{-1}h_1^{-1}h_0^{-1}\gamma = \mathbf{I}$. Tämä kuvaa kaikki alkiot itselleen, ja $\mathbf{I} \in \mathcal{U}_n$ kaikille $n \geq 4$, joten $\gamma \in G$.

2. Annetun neliön symmetriaryhmä G on isomorfinen ryhmän D_8 kanssa. Ryhmän muodostavat permutaatiot

$$\begin{aligned}
g_0 &= (0)(1)(2)(3)(4)(5)(6)(7)(8)\\
g_1 &= (0,2,8,6)(1,5,7,3)(4)\\
g_2 &= (0,8)(1,7)(2,6)(5,3)(4)\\
g_3 &= (0,6,8,2)(1,3,7,5)(4)\\
g_4 &= (0,2)(3,5)(6,8)(1)(4)(7)\\
g_5 &= (1,5)(0,8)(3,7)(2)(4)(6)\\
g_6 &= (0,6)(1,7)(2,8)(3)(4)(5)\\
g_7 &= (1,3)(2,6)(5,7)(0)(4)(8).
\end{aligned}$$

   Permutaatioiden tyypit ovat

$$\begin{aligned}
\mathrm{type}(g_0) &= (9,0,0,0,0,0,0,0,0)\\
\mathrm{type}(g_{1,3}) &= (1,0,0,2,0,0,0,0,0)\\
\mathrm{type}(g_2) &= (1,4,0,0,0,0,0,0,0)\\
\mathrm{type}(g_{4,5,6,7}) &= (3,3,0,0,0,0,0,0,0).
\end{aligned}$$

(a) Montako 5-osajoukkoa kukin permutaatio kuvaa itselleen? Jos permutaation minkä tahansa syklin jokin alkio kuuluu osajoukkoon, kaikkien ko. syklin alkioiden on kuuluttava osajoukkoon, jotta permutaatio kuvaisi osajoukon itselleen. Lasketaan siis, montako tapaa on valita permutaation syklejä siten, että syklien yhteenlaskettu pituus on 5.

$$\begin{aligned}
\chi_5(g_0) &= \binom{9}{5} = 126 \\
\chi_5(g_{1,3}) &= \binom{2}{1} = 2 \\
\chi_5(g_2) &= \binom{4}{2} = 6 \\
\chi_5(g_{4,5,6,7}) &= \binom{3}{1}\binom{3}{2} + \binom{3}{3}\binom{3}{1} = 12
\end{aligned}$$

Burnsiden lemmalla

$$N_k = \frac{1}{|G|}\sum_{g\in G}\chi_k(g)$$

saadaan

$$N_5 = \frac{1}{8}(126 + 2\cdot 2 + 6 + 4\cdot 12) = 23$$

(b) Muodostetaan k-osajoukkojen ratojen edustajajoukot R_k, kun $0 \le k \le 2$. Joukko R_2 tulee olemaan vastaus tehtävään.

$R_0 = \{\emptyset\}$, sillä tyhjä joukko on ainoa 9 alkion osajoukko, jonka koko on 0. Lisätään nyt vuorollaan kuhunkin R_0:n alkioon kukin alkioista $\{0,\ldots,8\}$ ja lisätään syntynyt tulos joukkoon R_1, ellei joukossa R_1 ole jo leksikaalisessa järjestyksessä aikaisemmin tullutta 1-osajoukkoa samalta radalta.

$$R_1 = \{\{0\},\{1\},\overbrace{\{2\}}^{\{0\}},\overbrace{\{3\}}^{\{1\}},\{4\},\overbrace{\{5\}}^{\{1\}},\overbrace{\{6\}}^{\{0\}},\overbrace{\{7\}}^{\{1\}},\overbrace{\{8\}}^{\{0\}}\} = \{\{0\},\{1\},\{4\}\}$$

Otetaan nyt vuorollaan kukin R_1:n joukoista ja lisätään siihen vuorollaan kukin alkio, joka on suurinta joukon alkiota suurempi. Lisätään syntynyt joukko R_2:een, ellei siellä jo ole aikaisempaa 2-osajoukkoa samalta radalta.

$$\begin{aligned}
R_2 = \{\; & \{0,1\},\{0,2\},\overbrace{\{0,3\}}^{\{0,1\}},\{0,4\},\{0,5\},\overbrace{\{0,6\}}^{\{0,2\}},\overbrace{\{0,7\}}^{\{0,5\}},\{0,8\}, \\
& \overbrace{\{1,2\}}^{\{0,1\}},\{1,3\},\{1,4\},\overbrace{\{1,5\}}^{\{1,3\}},\overbrace{\{1,6\}}^{\{0,5\}},\{1,7\},\overbrace{\{1,8\}}^{\{0,5\}}, \\
& \overbrace{\{4,5\}}^{\{1,4\}},\overbrace{\{4,6\}}^{\{0,6\}},\overbrace{\{4,7\}}^{\{1,4\}},\overbrace{\{4,8\}}^{\{0,4\}}\;\}
\end{aligned}$$

ja jäljelle jää siis

$$R_2 = \{\{0,1\},\{0,2\},\{0,4\},\{0,5\},\{0,8\},\{1,3\},\{1,4\},\{1,7\}\}.$$

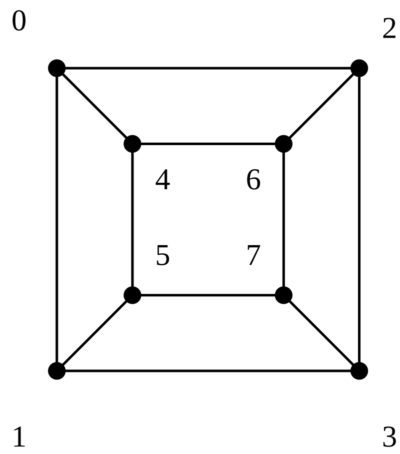
Huomautus: Yllä R_{k+1}:ä konstruoitaessa riittää lisätä edellisen R_k:n joukkoihin vuorollaan aiempia suuremmat alkiot, sillä tässä algoritmin versiossa kunkin radan edustajana on aina sen leksikaalisesti pienin alkio. Mille tahansa $k+1$-osajoukolle $S \in R_{k+1}$, joka siis on ratansa leksikaalisesti pienin alkio, pätee, että joukko $S' = S \setminus \{\max(S)\}$ on myös ratansa pienin alkio, ts. $S' \in R_k$. Tästä seuraa, että jokainen R_{k+1}:n alkio voidaan konstruoida jostakin R_k:n alkiosta lisäämällä siihen aiempia suurempi alkio.

Laskuharjoitus 9

15.03.2001

1. Ramseyn luku $R(k,l)$ on pienin kokonaisluku n, jolle pätee, että kaikissa n solmun täydellisen graafin $\mathbf{K}_n$ kaarten 2-värityksissä esiintyy väistämättä $\mathbf{K}_k$ ensimmäisellä tai $\mathbf{K}_l$ toisella värillä. Osoita, että $R(3,4) > 8$, konstruoimalla $\mathbf{K}_8$:n kaarten 2-väritys, jossa ei esiinny $\mathbf{K}_3$:a ensimmäisellä eikä $\mathbf{K}_4$:ä toisella värillä ja jolla on syklinen symmetria.

2. Olkoon $\mathcal{G}$ oheinen graafi. Määritä osajoukkojen $\{0,7\}$ ja $\{0,1,2,3\}$ radat ja stabiloija-aliryhmät ryhmän $\mathrm{Aut}(\mathcal{G})$ suhteen. Määritä lisäksi molempien stabiloija-aliryhmien jokin vasen transversaali ryhmän $\mathrm{Aut}(\mathcal{G})$ suhteen. (Ryhmän $\mathrm{Aut}(\mathcal{G})$ alkiot on listattu kirjan taulukossa 6.1, ja Schreier-Sims esitys esimerkissä 6.7.)



Kuva 7: Kuutio.

Laskuharjoituksen 9 vastaukset

1. Käytetään K_8:n symmetriaryhmänä diedraaliryhmää D_8, jonka generoivat permutaatiot $(0,1,2,3,4,5,6,7)$ ja $(0,7)(1,6)(2,5)(3,4)$. (Syklinen ryhmä C_8 on D_8:n aliryhmä, joten D_8-symmetrinen väritys on myös C_8-symmetrinen.) Värityksen symmetrisyys tarkoittaa sitä, että samanväriset kaaret kuvautuvat samanvärisiksi symmetriaryhmän alkion operoidessa kaarien solmuihin, ts. jokaiseen D_8-rataan voidaan käyttää vain yhtä väriä.

   Graafin K_8 kaarijoukon D_8-radat ovat (ks. myös kuva 8)

$$\begin{aligned}
\Delta_1 &= \{\{0,1\},\{1,2\},\{2,3\},\{3,4\},\{4,5\},[5,6\},\{6,7\},\{0,7\}\},\\
\Delta_2 &= \{\{0,2\},\{2,4\},\{4,6\},\{0,6\},\{1,3\},\{3,5\},\{5,7\},\{1,7\}\},\\
\Delta_3 &= \{\{0,3\},\{0,5\},\{1,4\},\{1,6\},\{2,5\},\{2,7\},\{3,6\},\{4,7\}\},\\
\Delta_4 &= \{\{0,4\},\{1,5\},\{2,6\},\{3,7\}\}.
\end{aligned}$$

   3-osajoukoilla on 5 D_8-rataa $\Gamma_1,\ldots,\Gamma_5$, joiden leksikografisesti pienimmät edustajat ovat

$$\gamma_1 = \{0,1,2\},\ \gamma_2 = \{0,1,3\},\ \gamma_3 = \{0,1,4\},\ \gamma_4 = \{0,2,4\},\ \gamma_5 = \{0,2,5\}.$$

   Vastaavasti 4-osajoukoilla on 8 D_8-rataa $\Sigma_1,\ldots,\Sigma_8$, joiden leksikografisesti pienimmät edustajat ovat

$$\begin{gathered}
\sigma_1 = \{0,1,2,3\},\ \sigma_2 = \{0,1,2,4\},\ \sigma_3 = \{0,1,2,5\},\\
\sigma_4 = \{0,1,3,4\},\ \sigma_5 = \{0,1,3,5\},\ \sigma_6 = \{0,1,3,6\},\\
\sigma_7 = \{0,1,4,5\},\ \sigma_8 = \{0,2,4,6\}.
\end{gathered}$$

   2-osajoukkojen ja 3-osajoukkojen ratainsidenssimatriisiksi saadaan

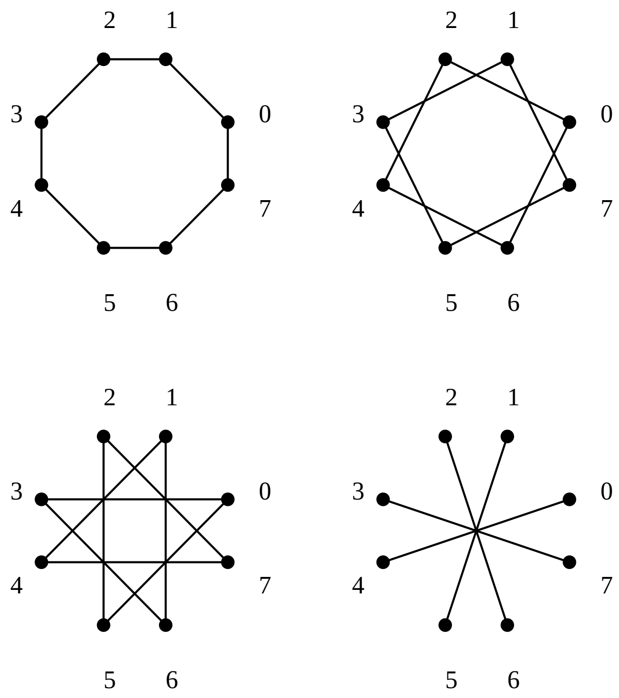
$$\begin{array}{cccccc}
 & \Gamma_1 & \Gamma_2 & \Gamma_3 & \Gamma_4 & \Gamma_5\\
\Delta_1 & 2 & 2 & 2 & 0 & 0\\
\Delta_2 & 1 & 2 & 0 & 2 & 1\\
\Delta_3 & 0 & 2 & 2 & 0 & 2\\
\Delta_4 & 0 & 0 & 4 & 2 & 0
\end{array}.$$

   Vastaavasti 2-osajoukkojen ja 4-osajoukkojen ratainsidenssimatriisiksi saadaan

$$\begin{array}{ccccccccc}
 & \Sigma_1 & \Sigma_2 & \Sigma_3 & \Sigma_4 & \Sigma_5 & \Sigma_6 & \Sigma_7 & \Sigma_8\\
\Delta_1 & 3 & 4 & 2 & 2 & 2 & 1 & 1 & 0\\
\Delta_2 & 2 & 4 & 1 & 1 & 4 & 2 & 0 & 1\\
\Delta_3 & 1 & 2 & 2 & 2 & 4 & 3 & 1 & 0\\
\Delta_4 & 0 & 4 & 2 & 2 & 4 & 0 & 2 & 1
\end{array}.$$

   Ratainsidenssimatriisin alkio rivillä i sarakkeessa j kertoo kuinka moneen radan Γ_j (vastaavasti Σ_j) alkioon radan Δ_i mielivaltainen alkio sisältyy. Esim. radan Δ_1 kaari $\{0,1\}$ sisältyy täsmälleen radan Γ_1 3-osajoukkoihin $\{0,1,2\}$ ja $\{0,1,6\}$.

   Haluttu kaariväritys voidaan nyt ratkaista ratainsidenssimatriisien avulla. Selvästi jokainen 3-osajoukko vastaa yksikäsitteisesti kolmesta kaaresta koostuvaa kolmiota ja toisaalta jokainen 4-osajoukko $\binom{4}{2} = 6$ kaaresta koostuvaa 4-klikkiä. Jos kaarivärityksessä esiintyy jokin kolmio värillä 1, värityksen symmetrian perusteella kaikki tämän kolmion kanssa samalla radalla Γ_j olevat



Kuva 8: K_8:n kaarien D_8-radat.

kolmiot on väritetty värillä 1. Tällöin siis kaikkien mahdollisten radan Γ_j kolmioihin (3-osajoukkoihin) sisältyvien kaarien on oltava väritetty värillä 1. Ratainsidenssimatriisin rakenteen perusteella tämä on selvästi mahdollista täsmälleen silloin jos värillä 1 väritettyihin ratoihin sisältyvät kaikki ne radat Δ_i, joilla on ratainsidenssimatriisin sarakkeessa j nollasta poikkeava arvo.

Näin ollen värillä 1 voidaan värittää ratajoukot $\{\Delta_1\}$, $\{\Delta_2\}$, $\{\Delta_3\}$, $\{\Delta_4\}$, $\{\Delta_1, \Delta_3\}$, $\{\Delta_1, \Delta_4\}$ ja $\{\Delta_3, \Delta_4\}$. Koska loput radat pitää pystyä värittämään värillä 2 siten, että värillä 2 ei esiinny 4-klikkiä, saadaan ehdot täyttäviä 2-värityksiä jälkimmäisen ratainsidenssimatriin perusteella 2 kappaletta: ratkaisussa 1 väritetään värillä 1 radat $\{\Delta_1, \Delta_4\}$ ja värillä 2 radat $\{\Delta_2, \Delta_3\}$; ratkaisussa 2 värillä 1 radat $\{\Delta_3, \Delta_4\}$, värillä 2 radat $\{\Delta_1, \Delta_2\}$.

2. Graafin $\mathcal{G}$ automorfismiryhmässä $\mathrm{Aut}(\mathcal{G})$ on 48 alkiota:

id	$(0,2,3,1)(4,6,7,5)$	$(0,5,3)(2,4,7)$
$(2,4)(3,5)$	$(0,2,6,7,5,1)(3,4)$	$(0,5)(2,7)$
$(1,2)(5,6)$	$(0,3,6)(1,7,4)$	$(0,5,3,6)(1,7,2,4)$
$(1,2,4)(3,6,5)$	$(0,3,5,6)(1,7,4,2)$	$(0,5,6)(1,7,2)$
$(1,4,2)(3,5,6)$	$(0,3,6,5)(1,2,7,4)$	$(0,6,3)(1,4,7)$
$(1,4)(3,6)$	$(0,3)(1,2)(4,7)(5,6)$	$(0,6,3,5)(1,4,2,7)$
$(0,1,3,7,6,4)(2,5)$	$(0,3,5)(2,7,4)$	$(0,6,5,3)(1,2,4,7)$
$(0,1,3,2)(4,5,7,6)$	$(0,3)(4,7)$	$(0,6,5)(1,2,7)$
$(0,1,5,4)(2,3,7,6)$	$(0,4,6,7,3,1)(2,5)$	$(0,6)(1,7)(2,4)(3,5)$
$(0,1,5,7,6,2)(3,4)$	$(0,4,5,1)(2,6,7,3)$	$(0,6)(1,7)$
$(0,1)(2,3)(4,5)(6,7)$	$(0,4,6,2)(1,5,7,3)$	$(0,7)(1,6)(2,5)(3,4)$
$(0,1)(2,5)(3,4)(6,7)$	$(0,4)(1,5)(2,6)(3,7)$	$(0,7)(1,6)(2,3)(4,5)$
$(0,2,6,4)(1,3,7,5)$	$(0,4,5,7,3,2)(1,6)$	$(0,7)(1,5)(2,6)(3,4)$
$(0,2)(1,3)(4,6)(5,7)$	$(0,4)(1,6)(2,5)(3,7)$	$(0,7)(1,5,4,6,2,3)$
$(0,2,3,7,5,4)(1,6)$	$(0,5,6,3)(1,4,7,2)$	$(0,7)(1,3,2,6,4,5)$
$(0,2)(1,6)(3,4)(5,7)$	$(0,5)(1,4)(2,7)(3,6)$	$(0,7)(1,3)(2,5)(4,5)$

(Ryhmän generoivat permutaatiot $(0,1,3,7,6,4)(2,5)$ ja $(0,1,3,2)(4,5,7,6)$.) Stabiloija-aliryhmät voi määrittää joko algoritmisesti ensin generoimalla ryhmälle Schreier-Sims esityksen (huomaa, että kirjan algoritmissa 6.9 on painovirhe; errata löytyy osoitteesta `http://www.math.mtu.edu/~kreher/cages.html`) ja tämän jälkeen käymällä algoritmin 6.6 avulla ryhmän kaikki alkiot läpi ja tulostamalla ne, jotka kuvaavat joukot $\{0,7\}$ ja $\{0,1,2,3\}$ itselleen. Toisaalta ainakin joukon $\{0,7\}$ stabiloijat

id	$(1,4,2)(3,5,6)$	$(0,7)(1,5)(2,6)(3,4)$
$(2,4)(3,5)$	$(1,4)(3,6)$	$(0,7)(1,5,4,6,2,3)$
$(1,2)(5,6)$	$(0,7)(1,6)(2,5)(3,4)$	$(0,7)(1,3,2,6,4,5)$
$(1,2,4)(3,6,5)$	$(0,7)(1,6)(2,3)(4,5)$	$(0,7)(1,3)(2,5)(4,5)$

on helppo nähdä ryhmän sykliesityksestä suoraan. Stabiloija-aliryhmän $\mathrm{Aut}(\mathcal{G})_{\{0,7\}}$ koko on siis 12, joten joukon $\{0,7\}$ radan ja toisaalta stabiloijan vasemman transversaalin koko on

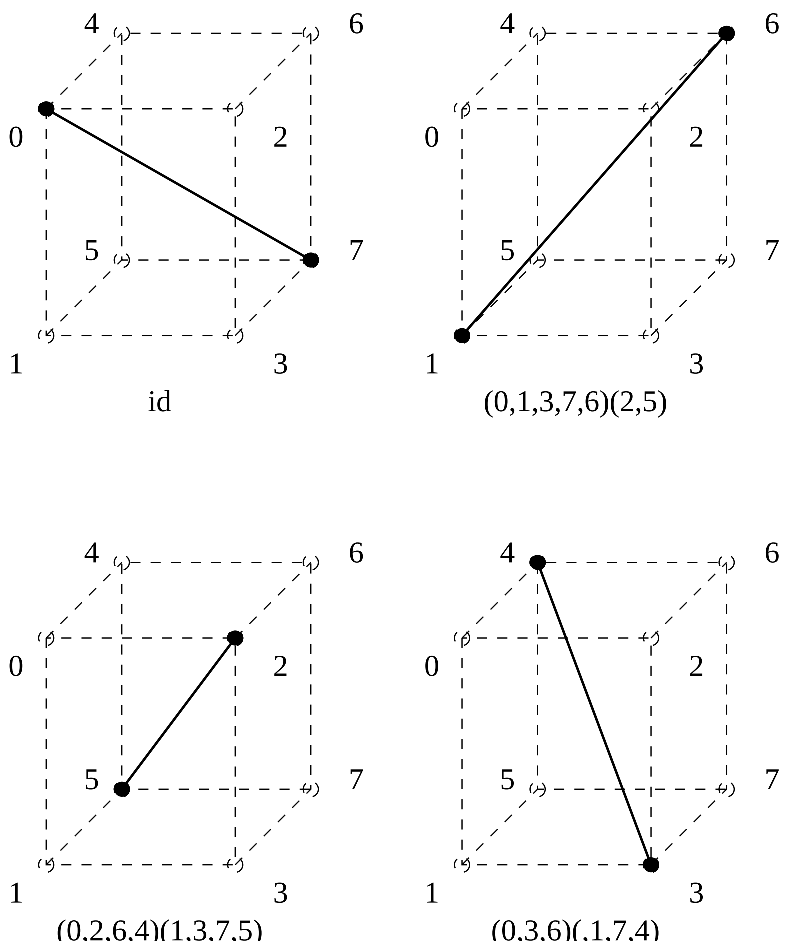
$$|\mathrm{Aut}(\mathcal{G})|/|\mathrm{Aut}(\mathcal{G})_{\{0,7\}}| = 48/12 = 4.$$

Vasemman transversaalin voi määrittää esim. kirjan algoritmilla 6.17, joka laskee juuri vasemman transversaalin (tätä ei erikseen mainita kirjan tekstissä). Erääksi vasemmaksi transversaaliksi saadaan

$$\begin{array}{ll} g_1 = id & g_3 = (0,2,6,4)(1,3,7,5) \\ g_2 = (0,1,3,7,6,4)(2,5) & g_4 = (0,36)(1,7,4) \end{array}.$$

Osajoukon $\{0,7\}$ rata ja vastaava transversaali on esitetty kuvassa 9. Vastaavasti osajoukon $\{0,1,2,3\}$ stabiloija-aliryhmä $\mathrm{Aut}(\mathcal{G})_{\{0,7\}}$ koostuu permutaatioista

$$\begin{array}{ll} id & (0,2)(1,3)(4,6)(5,7) \\ (1,2)(5,6) & (0,2,3,1)(4,6,7,5) \\ (0,1,3,2)(4,5,7,6) & (0,3)(1,2)(4,7)(5,6) \\ (0,1)(2,3)(4,5)(6,7) & (0,3)(4,7) \end{array},$$

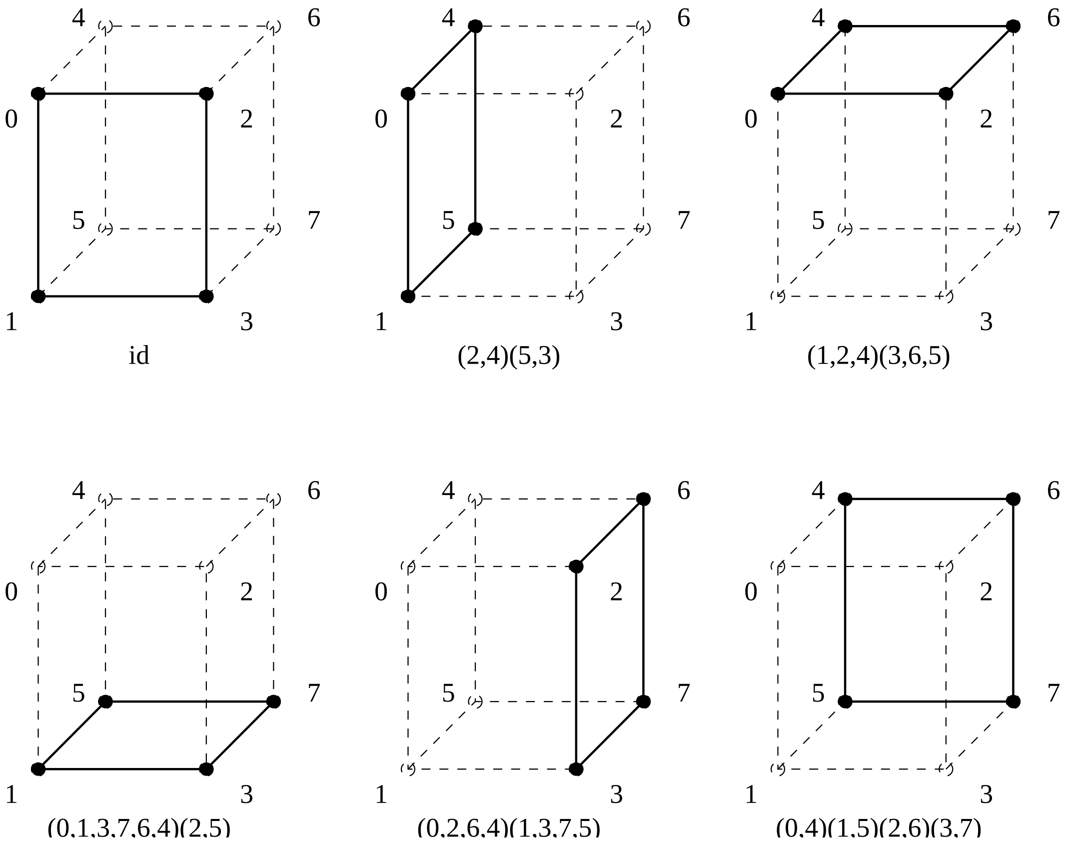


Kuva 9: Osajoukon $\{0, 7\}$ rata ja tätä vastaavat stabiloijan $\mathrm{Aut}(\mathcal{G})_{\{0,7\}}$ erään vasemman transversaalin alkiot.

joten osajoukon $\{0, 1, 2, 3\}$ radan pituus ja toisaalta $\mathrm{Aut}(\mathcal{G})_{\{0,7\}}$:n vasemman transversaalin koko on $48/8 = 6$. Eräs vasen transversaali koostuu permutaatioista

$$\begin{array}{lll} g_1 = id & g_3 = (1,2,4)(3,6,5) & g_5 = (0,2,6,4)(1,3,7,5) \\ g_2 = (2,4)(5,3) & g_4 = (0,1,3,7,6,4)(2,5) & g_6 = (0,4)(1,5)(2,6)(3,7) \end{array}.$$

Osajoukon rata ja vastaavat transversaalin alkiot on esitetty kuvassa 10.

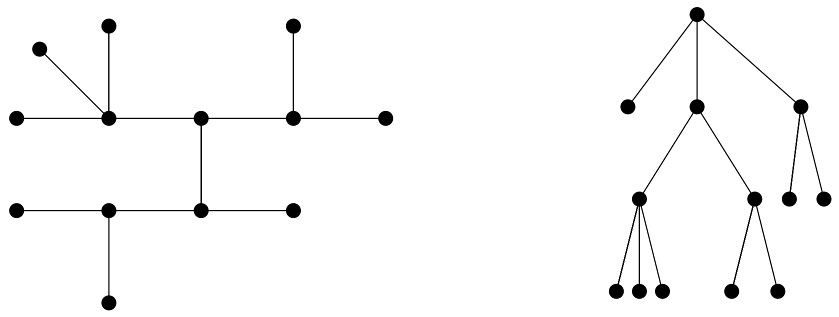


Kuva 10: Osajoukon $\{0, 1, 2, 3\}$ rata ja tätä vastaavat stabiloijan $\mathrm{Aut}(\mathcal{G})_{\{0,1,2,3\}}$ erään vasemman transversaalin alkiot.

Laskuharjoitus 10

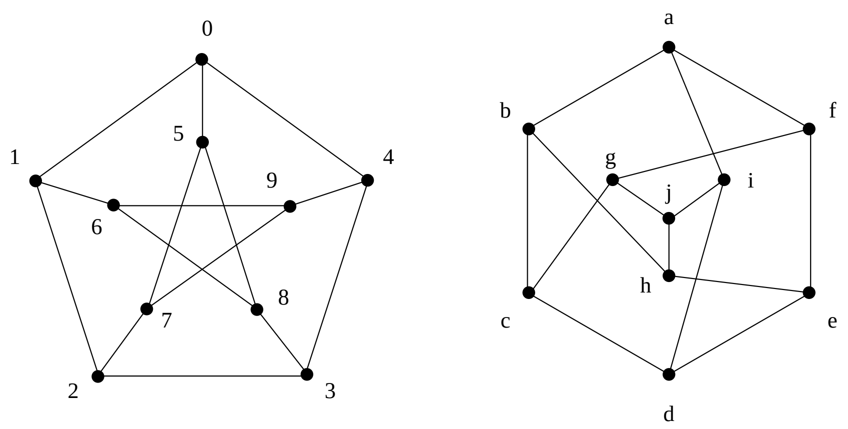
22.03.2001

1. Tutki laskemalla sertifikaatti, ovatko oheiset puut isomorfisia.



2. Muunna edellisen tehtävän tuloksena saatu sertifikaatti takaisin puuksi.

3. (Kirjan tehtävä 7.1 lisäyksellä) Etsi isomorfismi seuraavien graafien välille, ja määritä graafien automorfismiryhmien koko. (Vihje: graafien automorfismiryhmät ovat transitiivisia[1] solmujoukossa.)



[1]Permutaatioryhmä $G \subseteq \mathrm{Sym}(\{0, 1, \ldots, n-1\})$ on *transitiivinen* jos kaikilla $x, y \in \{0, 1, \ldots, n-1\}$ on olemassa $g \in G$ jolle $y = g(x)$.

Laskuharjoituksen 10 vastaukset

1. Käytetään kirjan kappaleessa 7.3.1. esitettyä algoritmia hieman tarkennettuna.

   (a) Nimeä puussa kaikki solmut 01:ksi.

   (b) Niin kauan kuin puussa on enemmän kuin kaksi solmua, toista seuraavaa.

      i. Määritä kaikki ne ei-lehtisolmut (solmut, joiden asteluku on suurempi kuin 1), joilla on naapurina enintään yksi ei-lehtisolmu, ja aseta nämä solmut joukkoon T.

      ii. Tee kullekin $x \in T$ seuraavaa:

         A. Kerää joukkoon Y x:n naapurina olevien lehtisolmujen nimet sekä solmun x nimi, josta on poistettu alku-0 ja loppu-1.

         B. Korvaa x:n nimi nimellä, joka saadaan järjestämällä Y:n alkiot leksikaaliseen järjestykseen, liittämällä ne yhdeksi bittijonoksi ja lisäämällä alkuun 0 ja loppuun 1.

         C. Poista x:n naapurina olevat lehtisolmut.

   (c) Jos jäljellä on vain yksi solmu, sen nimi on puun sertifikaatti.

   (d) Jos jäljellä on kaksi solmua, niiden nimet leksikaalisessa järjestyksessä yhteenliitettyinä muodostavat puun sertifikaatin.

   Algoritmin eteneminen tehtävässä annetuille puille on esitetty kuvassa 11. (Ympyröidyt solmut ilmoittavat kulloinkin joukkoon T kuuluvat solmut.) Selvästi tuloksena saadaan molemmille puille sama sertifikaatti, joten puut ovat isomorfiset.

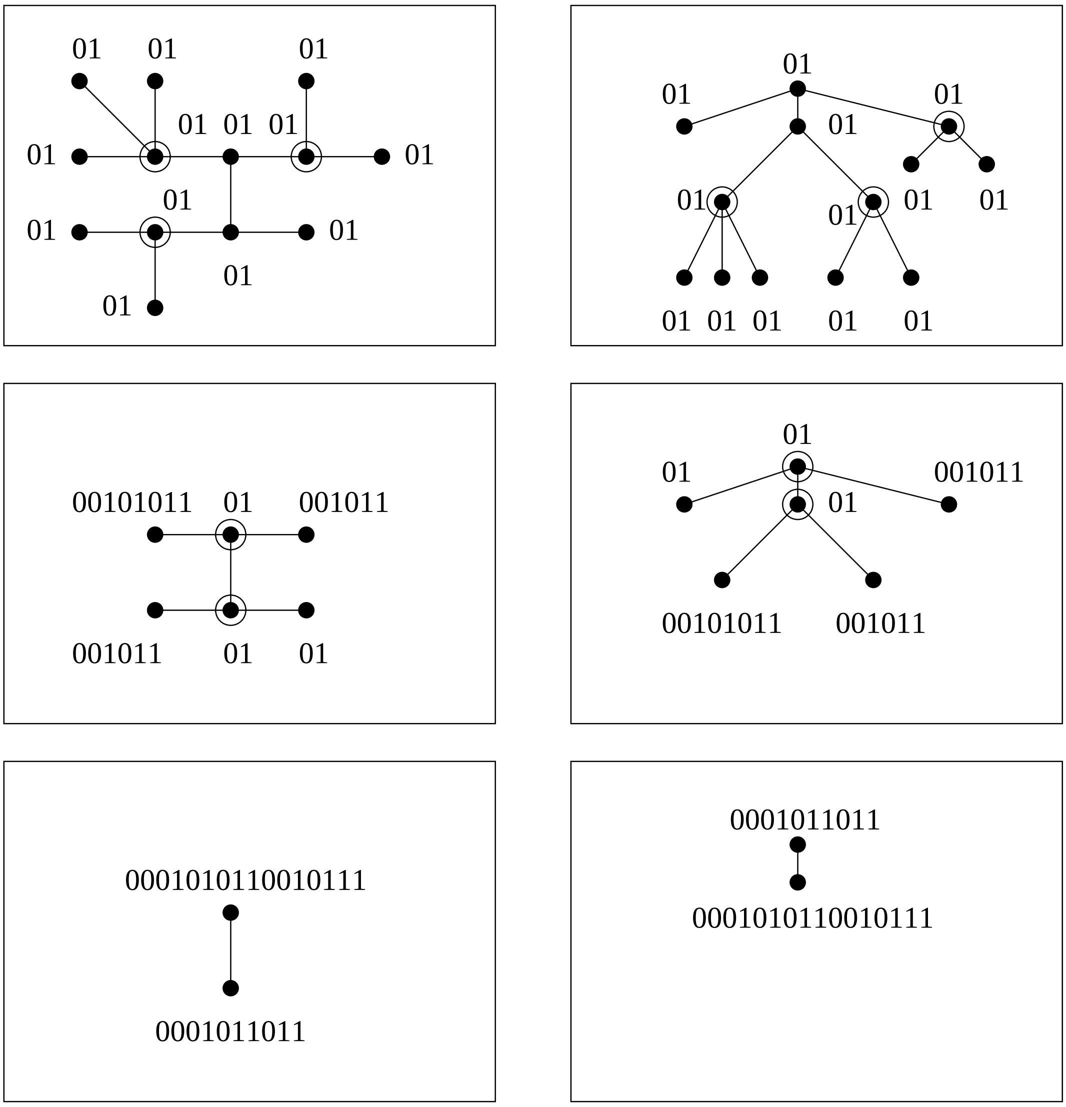
2. Käytetään kirjassa esitettyä nousevan veden algoritmia. Algoritmin kulku on esitetty kuvassa 12.

3. Tehtävän Petersenin graafi on esimerkki erittäin säännöllisestä graafista, jolle mikään invariantin indusoiva funktio ei osita solmujoukkoa useampaan kuin yhteen osaan. Näin ollen esimerkiksi kirjan algoritmi 7.1 tekee brute-force peräytyvän haun etsiessään isomorfismeja riippumatta käytetyistä invariantin indusoivista funktioista.

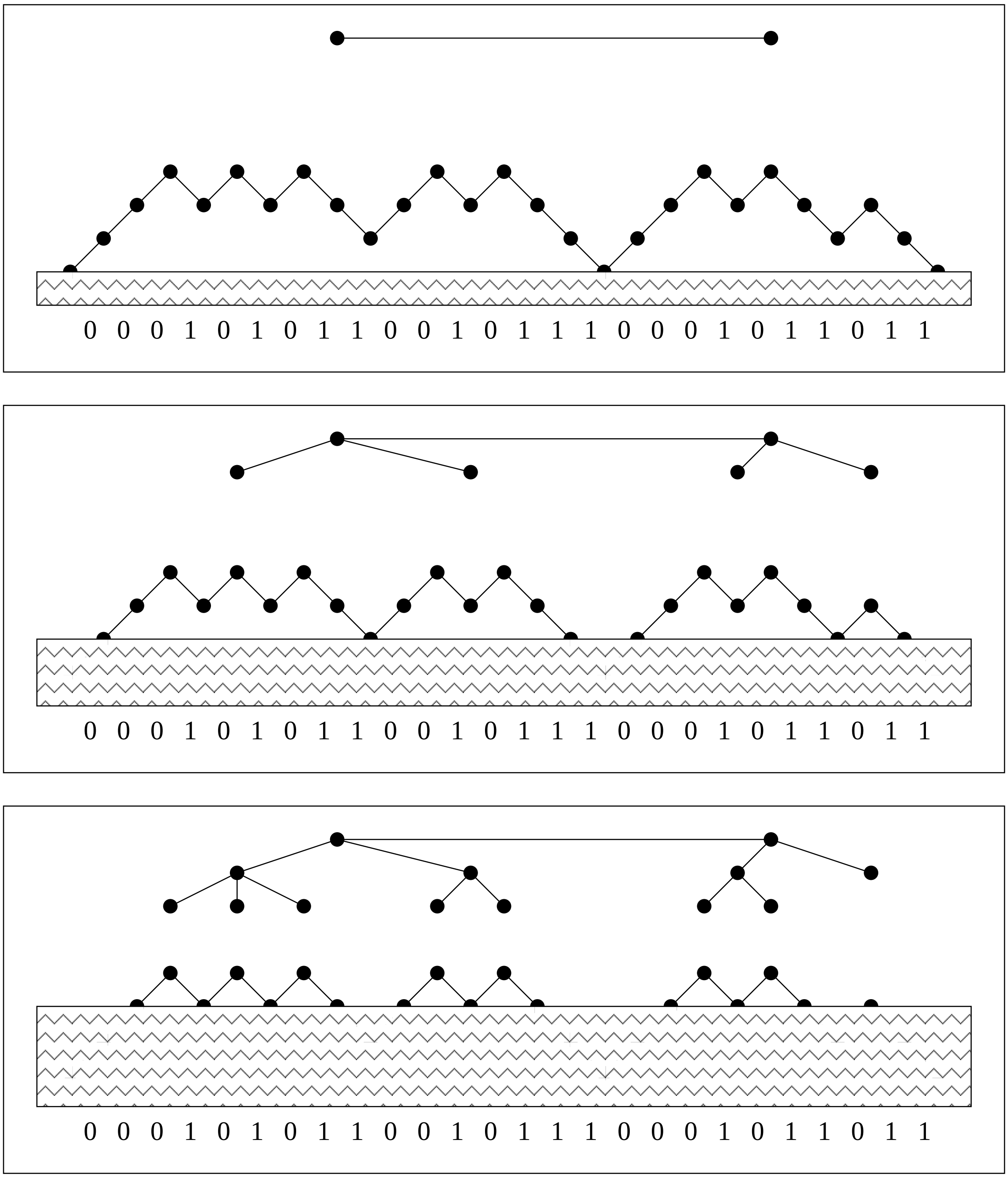
   Hieman parempaan tulokseen voidaan päästä esimerkiksi käyttämällä sertifikaattien määrittämisen yhteydessä esitettyjä ideoita. Jokin isomorfismi graafien välille voidaan määrittää esimerkiksi laskemalla sertfikaatti molemmille graafeille, ja tallentamalla haun yhteydessä (jokin) solmupermutaatio π, joka tuottaa pienimmän sertifikaattimatriisin. (Oletetaan oikeanpuoleisen graafin solmut numeroiduiksi siten, että $a \mapsto 0$, $b \mapsto 1$, jne.)

   Jos permutaatio $\pi_1 \in \mathrm{Sym}(\{0, \ldots, 9\})$ tuottaa sertifikaatin vasemman graafin naapuruusmatriisista A, ja $\pi_2 \in \mathrm{Sym}(\{0, \ldots, 9\})$ vastaavasti oikeanpuoleisen graafin naapuruusmatriisista B, saadaan $A_{\pi_1}[i, j] = B_{\pi_2}[i, j]$ kaikilla i, j, koska graafit ovat isomorfiset ja sertifikaatit siten samat. Nyt

$$\begin{aligned} B[i,j] &= B[\pi_2(\pi_2^{-1}(i)), \pi_2(\pi_2^{-1}(j))] = B_{\pi_2}[\pi_2^{-1}(i), \pi_2^{-1}(j)] = \\ &= A_{\pi_1}[\pi_2^{-1}(i), \pi_2^{-1}(j)] = (A_{\pi_1})_{\pi_2^{-1}}[i,j] = A_{\pi_1 \pi_2^{-1}}[i,j], \end{aligned}$$



Kuva 11: Puusertifikaattialgoritmin kulku.



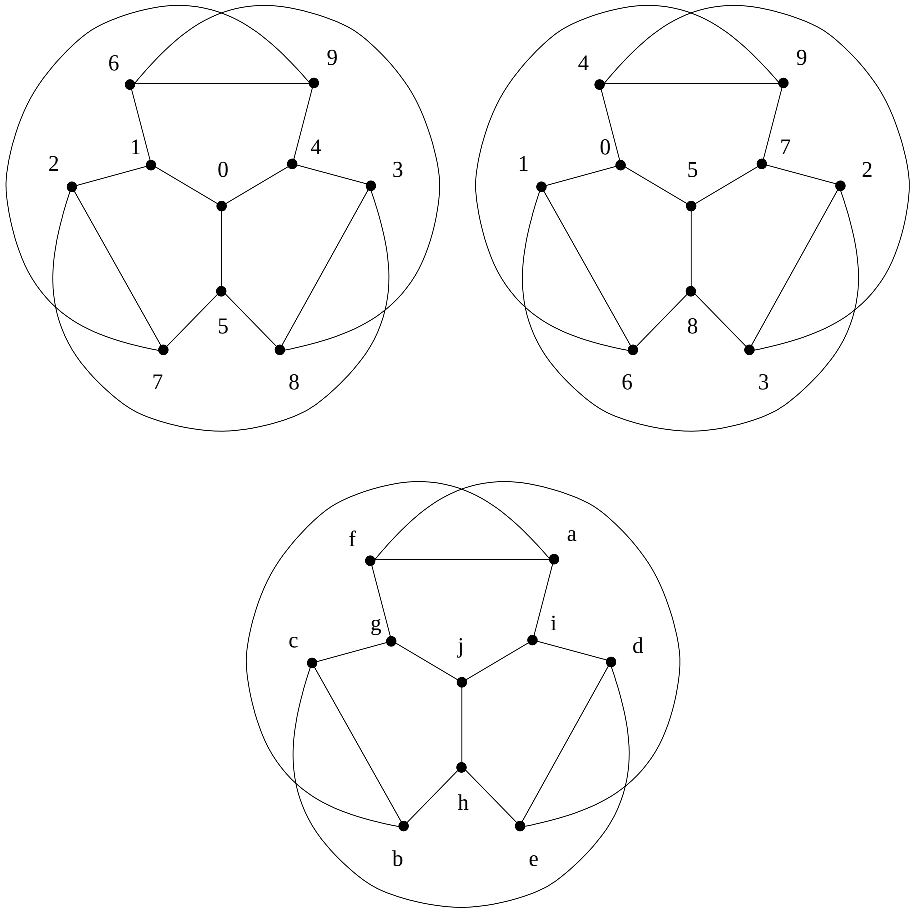
Kuva 12: Nousevan veden algoritmin kulku.

joten permutaatio $\pi_1\pi_2^{-1}$ on haluttu isomorfismi, joka kuvaa vasemmanpuoleisen graafin solmut oikeanpuoleisen graafin solmuille siten, että solmujen naapuruussuhteet säilyvät.

Kynällä ja paperilla graafien välisen isomorfismin voi löytää esimerkiksi tarkastelemalla jonkin solmun ympäristöä (erityisesti transitiivisen automorfismiryhmän tapauksessa tämä lähestymistapa toimii hyvin, koska lähtösolmuksi voi valita minkä tahansa solmun). Kuvan 13 vasemmanpuolimmaisessa graafissa on tarkasteltu solmun 0 ympäristöä, ja oikeanpuolimmaisessa solmun 5 ympäristöä: solmun 0 naapureita ovat solmut 1,4 ja 5. Edelleen solmun 1 uusia naapureita ovat solmut 2 ja 6, solmun 4 uusia naapureita ovat solmut 3 ja 9, ja solmun 5 uusia naapureita ovat solmut 7 ja 8. Jos tämä ympäristö piirretään säteittäin solmun 0 ympärille, saadaan (eräässä tapauksessa) tulokseksi kuvan vasemmanpuolimmainen graafi. Jos vastaava konstruktio tehdään alkuperäiselle oikeanpuoleiselle graafille alkaen solmusta j, saadaan (eräässä tapauksessa) selvästi tulokseksi kuvan 13 alimmainen graafi, josta isomorfismi voidaan nähdä suoraan.

Graafin automorfismiryhmän koko voidaan määrittää joko kirjassa esitetyillä algoritmeilla (esim. algoritmi 7.2 tai 7.9), tai vaihtoehtoisesti symmetriatarkastelulla. Kuvan 13 graafien rakenteesta voidaan nähdä, että kun lähtösolmu on valittu, voidaan sen naapurisolmut selvästi asettaa mielivaltaiseen järjestykseen, ja edelleen jäljellejäävät solmut voidaan järjestää siten, että tulokseksi saadaan sama graafi. Kun lähtösolmun naapurien järjestys on kiinnitetty, voidaan vielä selvästi yhden "haaran" (esim. vasemmanpuoleisessa kuvassa solmun 1 "haaran" muodostavat solmut 2 ja 6) solmujen järjestys kiinnittää siten, että jäljelle jäävät solmut sopivasti järjestämällä tulokseksi saadaan sama graafi. "Haaran" kiinnittämisen jälkeen tämä järjestys on selvästi yksikäsitteinen.

Koska alkuperäisellä vasemmanpuoleisella graafilla on selvästi syklinen kiertosymmetria, ja toisaalta kuvassa 13 on esitetty identtiset naapurustograafit solmuille 0 ja 5, on selvää, että naapurustograafin automorfismiryhmän on oltava transitiivinen, ts. samanlaisen naapurustograafin voi aina piirtää riippumatta lähtösolmusta. Nyt lähtösolmun voi siis valita 10 tavalla, lähtösolmun naapurien järjestyksen $3! = 6$ tavalla, ja lopuksi yhden "haaran" järjestyksen kahdella tavalla. Näin ollen naapurustograafin voi kuvata itselleen täsmälleen $10 \cdot 6 \cdot 2 = 120$ tavalla.



Kuva 13: Petersenin graafien isomorfismi.

Kotitehtävä 3

Kolmannessa kotitehtävässä etsitään virheenkorjaavia vakiopainobinäärikoodeja, joissa on mahdollisimman monta koodisanaa.

Koodisana on järjestetty lista bittejä. Koodisanan *pituus* on sen bittien lukumäärä. Koodisanan *paino* on sen 1-bittien lukumäärä. Kahden koodisanan *Hamming-etäisyys* on niiden positioiden lukumäärä, joissa koodisanat eroavat toisistaan.

Virheenkorjaava (n, d)-koodi, missä n on koodin *pituus* ja d koodin *minimietäisyys*, on joukko koodisanoja, joiden pituus on n, ja kaikkien koodisanaparien Hamming-etäisyys on vähintään d. Vakiopainokoodit ovat virheenkorjaavien koodien erikoistapaus. (n, d, w)-vakiopainokoodi on virheenkorjaava (n, d)-koodi, jossa jokaisen koodisanan paino on w.

Olkoon $A(n, d, w)$ on suurin mahdollinen koodisanojen lukumäärä (n, d, w)-vakiopainokoodissa. Tällaisia (n, d, w)-vakiopainokoodeja sanotaan *optimaalisiksi*. Kotitehtävässä määritellään $A(n, d, w)$:lle täsmällisiä arvoja ja alarajoja esittämällä vakiopainokoodien konstruointiongelma klikkiongelmana.

Muunnos klikkiongelmaksi

Optimaalisten vakiopainokoodien konstruointiongelman voi esittää suurimman klikin etsimisenä graafista seuraavasti. Muodostetaan graafi, jonka solmut koostuvat kaikista n bitin koodisanoista, joiden paino on w. Kahden solmun välillä on kaari jos ja vain jos solmut voivat kuulua samaan vakiopainokoodiin; toisin sanoen, jos vastaavien koodisanojen Hamming-etäisyys on vähintään d. Syntyneen graafin klikit vastaavat nyt (n, d, w)-vakiopainokoodeja; erityisesti suurimmat graafin klikit ovat kokoa $A(n, d, w)$ ja vastaavat optimaalisia koodeja.

Symmetrioiden käyttö

Kun graafin koko kasvaa suureksi, käy työlääksi etsiä sen suurin klikki. Tällöin voidaan rajoittua tarkastelemaan vakiopainokoodeja, joilla on tiettyjä automorfismeja.

Automorfismien esittämiseksi tarkastellaan vakiopainokoodeja nyt joukkojärjestelmänä. Joukkojärjestelmän lohkot B_i vastaavat koodisanoja: lohko on niiden positioiden joukko, joissa vastaavassa koodisanassa on ykkönen. Näin saadaan joukkojärjestelmä $(\mathcal{X}, \mathcal{B})$, missä $\mathcal{X} = \{1, \ldots, n\}$ ja $\mathcal{B} = \{B_1, \ldots, B_k\}$, missä $B_i \subseteq \mathcal{X}$ ja $|B_i| = w$. Minimietäisyysehto voidaan esittää muodossa $|(B_i \setminus B_j) \cup (B_j \setminus B_i)| \geq d$, kun $0 < i < j \leq k$.

Nyt voidaan vaatia, että joukon $\mathcal{X}$ permutaatio $\pi = (1, 2, \ldots, n)$ on joukkojärjestelmän automorfismi (ks. kirjan määritelmä 6.4). Tässä tapauksessa joukkojärjestelmän on oltava yhdiste permutaation π generoiman *syklisen ryhmän* C_n radoista joukon $\mathcal{X}$ w-osajoukkojen joukossa. Lisäksi huomataan, että jotkut radat eivät kelpaa joukkojärjestelmän osiksi, koska ne sisältävät kaksi joukkoa, jotka eivät täytä minimietäisyysehtoa.

Kun minimietäisyysehdon täyttävät C_n-radat on konstruoitu, ongelma voidaan esittää painoltaan suurimman klikin etsimisenä painotetusta graafista. *Painotetussa graafissa* jokaisella solmulla on paino, ja painoltaan suurin klikki on klikki, jonka solmujen painojen summa on suurin. Graafin solmut koostuvat kaikista minimietäisyysehdon täyttävistä radoista, ja kunkin solmun paino on vastaavan radan pituus. Kahden solmun välillä on kaari jos ja vain jos molempien ratojen koodisanat voidaan sisällyttää koodiin rikkomatta etäisyysehtoa. Nyt painoltaan suurimman klikin paino vastaa suurimman vakiopainokoodin kokoa, jolla on automorfismina π.

Osoitteesta `http://www.tcs.hut.fi/~pat/wclique.html` löytyy ohjelma, jolla voi etsiä annetusta graafista sen painoltaan suurimman klikin.

Tehtävä

1. Rajoittamatta hakua symmetrioin: etsi $A(n, d, w)$, kun $d = 4$, $w = 3$. Aloita haku esimerkiksi arvosta $n = 6$ ja kasvata koodin pituutta, kunnes laskenta vie liikaa aikaa.

2. Kun laskenta alkaa viedä liikaa aikaa, rajoita tarkastelu koodeihin, joilla on ylläkuvatunlainen syklinen automorfismi. Jos löytämässäsi koodissa on k koodisanaa, saat alarajan $A(n, d, w) \geq k$ — koska on olemassa (n, d, w)-vakiopainokoodi, jossa on k koodisanaa, parhaan (n, d, w)-vakiopainokoodin koko $A(n, d, w)$ on varmasti ainakin k. Vertaa saamiasi alarajoja oheisen taulukon arvoihin.

n	11	12	13	14	15	16	17	18	19	20
$A(n, 4, 3)$	17	20	26	28	35	37	44	48	57	60

n	21	22	23	24	25	26	27	28	29	30
$A(n, 4, 3)$	70	73	83	88	100	104	117	121	134	140

Taulukko 6: $A(n, 4, 3)$, kun $11 \leq n \leq 30$

Palautusohje

Palautukseen tulee sisältyä ratkaisun sanallinen kuvaus, ohjelmien lähdekoodit ja yhteenveto löydetyistä $A(n, d, w)$:n arvoista ja alarajoista.

Palautus: Palautus assistentille (Petteri Kaski, Petteri.Kaski@hut.fi, TB358). Kotitehtävän voi palauttaa myös sähköpostitse esim. tekstinä tai hyvälaatuisena Postscriptinä. Viimeinen palautuspäivä on torstai 19.4.2001.