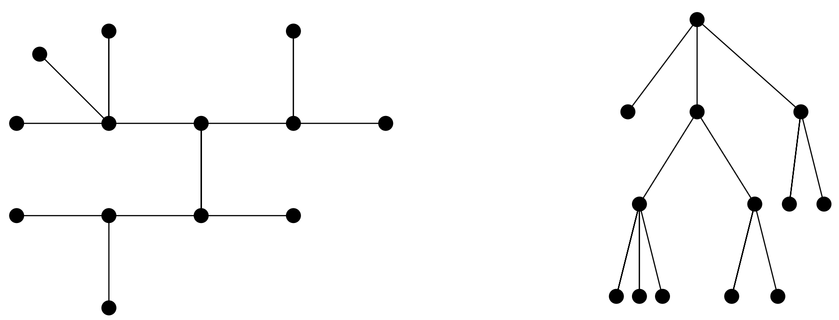
Laskuharjoitus 10

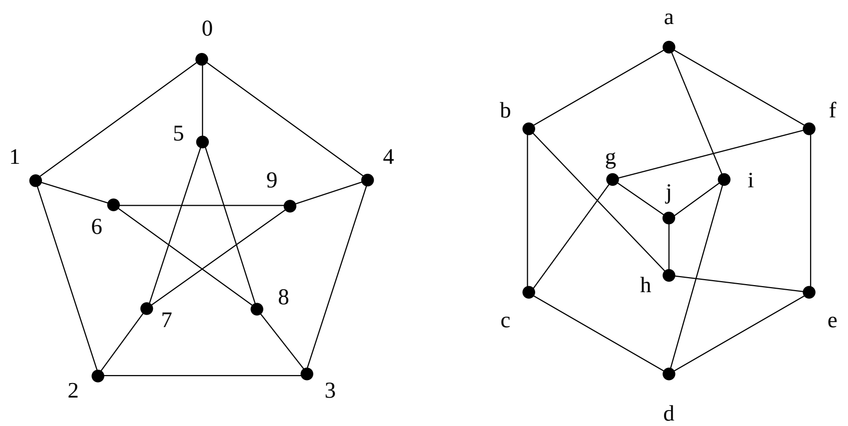
22.03.2001

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.

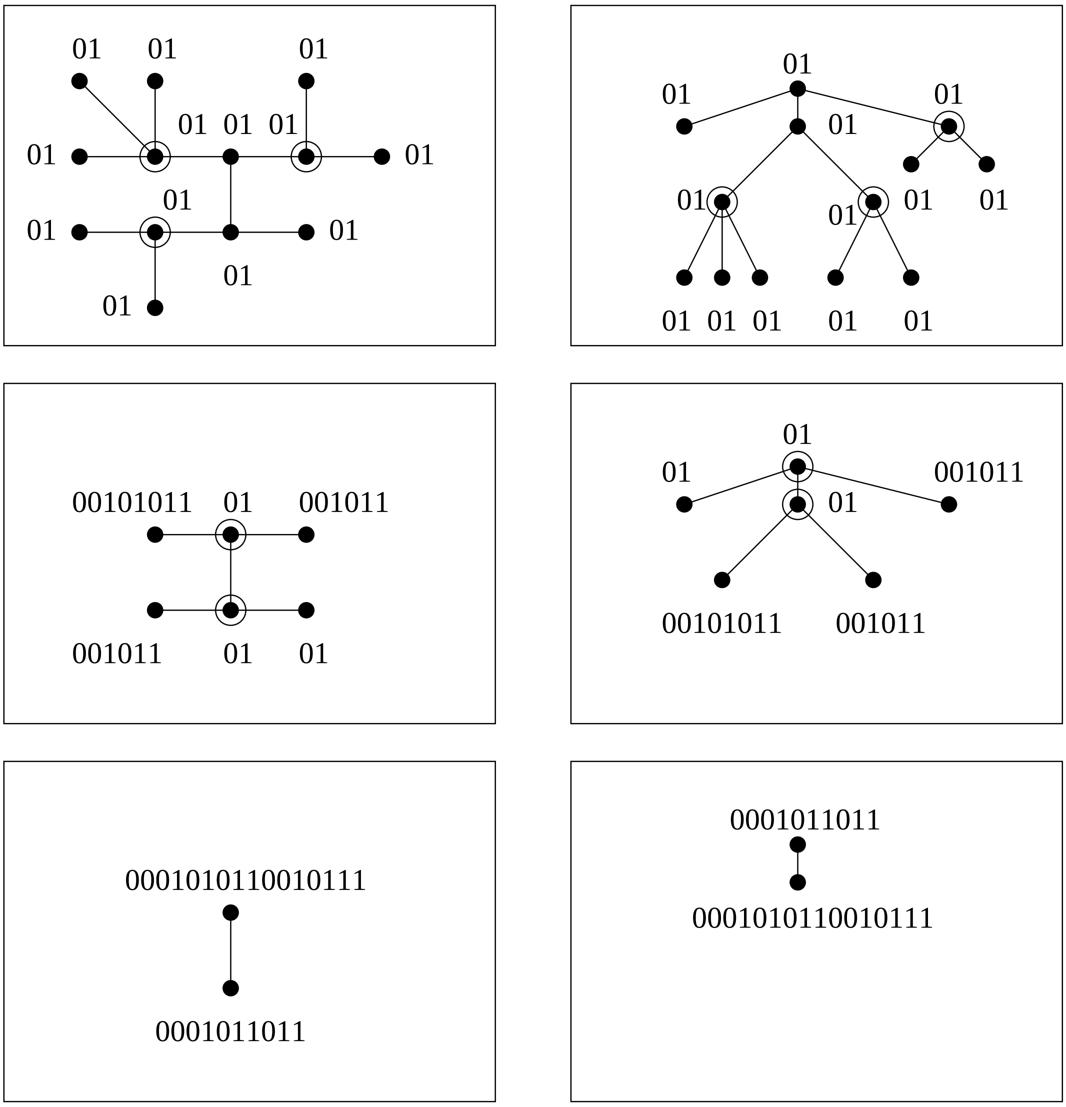
2. Käytetään kirjassa esitettyä nousevan veden algoritmia. Algoritmin kulku on esitetty kuvassa 12.

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.

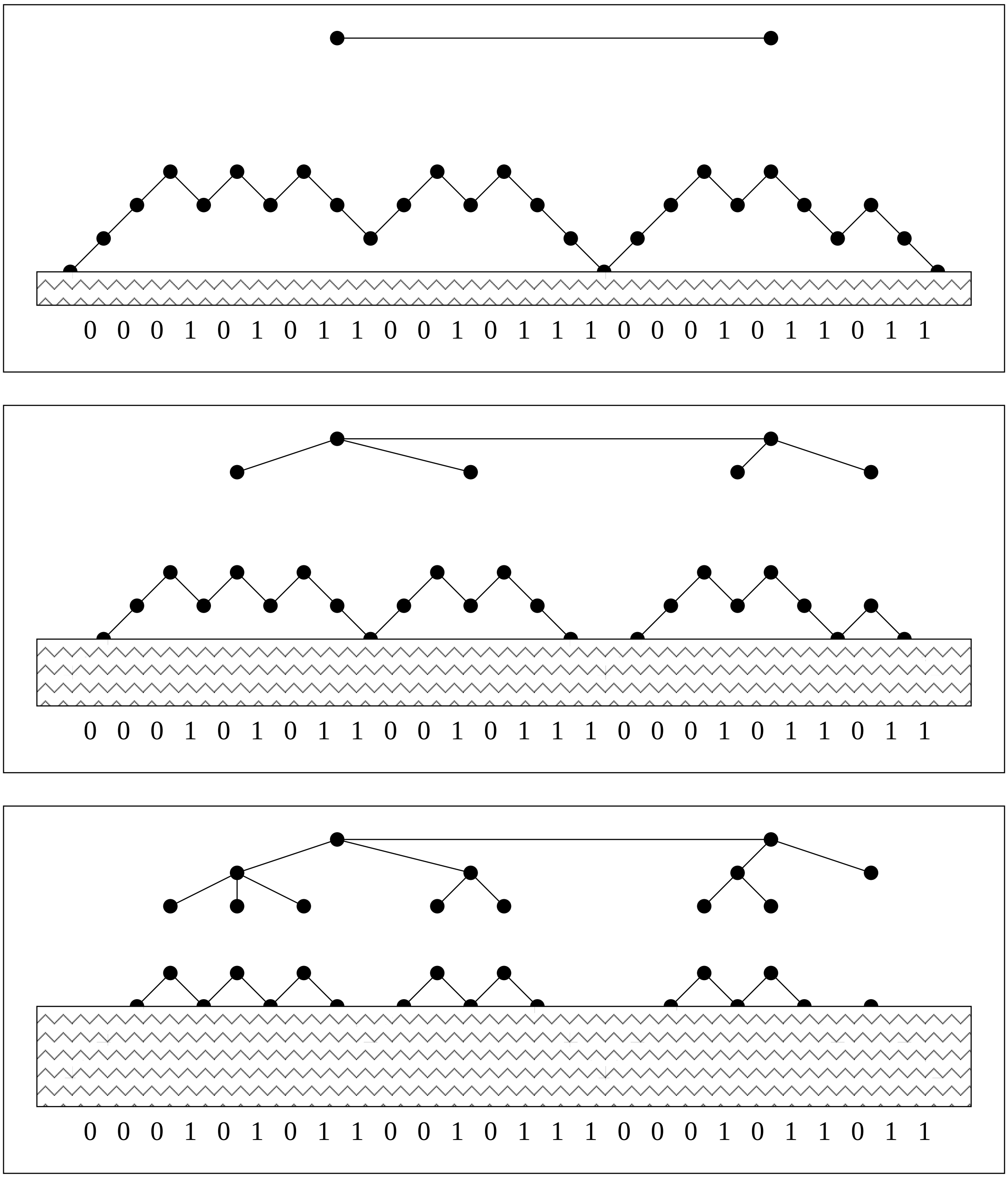
   Hieman parempaan tulokseen voidaan päästä esimerkiksi käyttämällä sertifikaattien määrittämisen yhteydessä esitettyjä ideoita. Jokin isomorfismi graafien välille voidaan määrittää esimerkiksi laskemalla sertfikaatti molemmille graafeille, ja tallentamalla haun yhteydessä (jokin) solmupermutaatio π, joka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

   Jos permutaatio $\pi_1 \in \mathrm{Sym}(\{0, \dots, 9\})$ tuottaa sertifikaatin vasemman graafin naapuruusmatriisista A, ja $\pi_2 \in \mathrm{Sym}(\{0, \dots, 9\})$ 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.



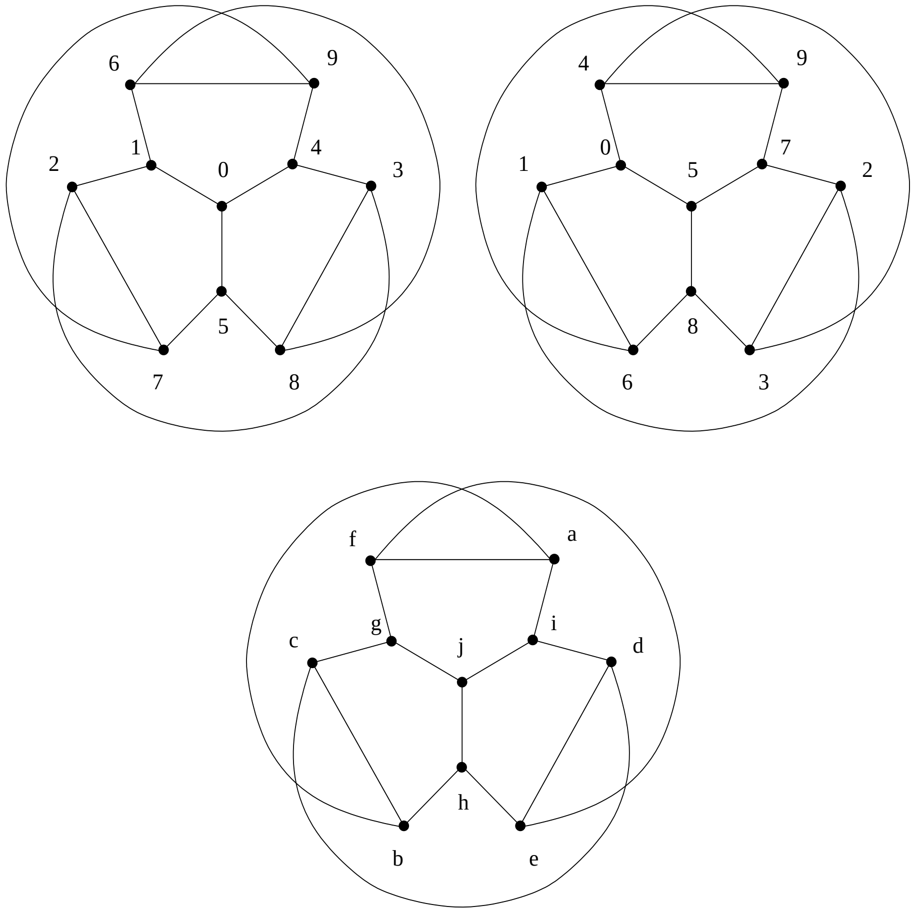
Kuva 12: Nousevan veden algoritmin kulku.

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.