

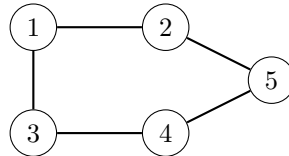
582359 Algoritmit ongelmanratkaisussa (kevät 2013)

Viikon 7 ratkaisuja

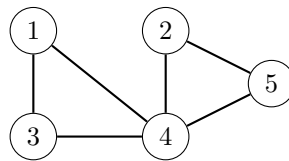
Tämän viikon tehtävissä kaikki verkot ovat suuntaamattomia ja painottamattomia.

1. Tutki seuraavista verkoista, onko niissä Eulerin tai Hamiltonin kierros:

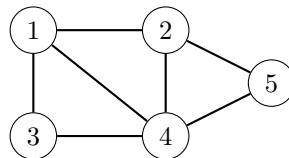
(a)



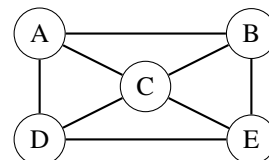
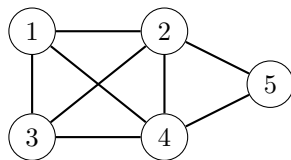
(b)



(c)



Tutki lisäksi, ovatko seuraavat verkot isomorfiset:



Ratkaisu:

(a) Verkossa on Eulerin kierros $1 \rightarrow 2 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 1$. Sama kierros on myös Hamiltonin kierros.

(b) Verkossa on Eulerin kierros $1 \rightarrow 4 \rightarrow 2 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 1$. Verkossa ei ole Hamiltonin kierrosta.

(c) Verkossa on Hamiltonin kierros $1 \rightarrow 2 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 1$. Verkossa ei ole Eulerin kierrosta.

Verkot eivät ole isomorfiset, koska vasemmassa verkossa solmun 5 asteluku on 2, mutta oikeassa verkossa ei ole mitään solmua, jonka asteluku olisi 2.

Tehtävissä 2–5 käytetään tiedostoa `Verkko.java`, joka sisältää verkon toteutuksen.

2. Toteuta algoritmi, joka muodostaa Eulerin kierroksen annettuun verkkoon. Jos verkossa ei ole Eulerin kierrosta, algoritmin täytyy huomata tämä. *Vihje:* Hierholzerin algoritmi on hyvä valinta.

Ratkaisu:

Tiedostossa `Euler.java` on Hierholzerin algoritmin toteutus. Lisäksi tiedostossa `Verkko.java` metodi `euler` tarkistaa solmujen asteista, onko verkossa Eulerin kierrosta.

3. Toteuta algoritmi Hamiltonin kierroksen muodostamiseen. Voit toteuttaa brute-force-algoritmin, dynaamisen ohjelmoinnin algoritmin tai molemmat.

Ratkaisu:

Tiedostossa `Hamilton.java` on brute-force-algoritmi Hamiltonin kierroksen muodostamiseen.

4. Toteuta algoritmi, joka tarkistaa, ovatko kaksi verkkoa isomorfiset.

Ratkaisu:

Tiedoston `Verkko.java` metodi `isomorfinen` tarkistaa, onko verkko isomorfinen annetun toisen verkon kanssa. Metodi tarkistaa ensin, että verkoissa on yhtä monta solmua ja samat määrät eriasteisia solmuja. Tämän jälkeen metodi käy läpi kaikki mahdolliset tavat yhdistää solmut toisiinsa.

5. Tee ohjelma, jolle annetaan syötteenä verkon solmujen määrä n . Tämän jälkeen ohjelma ilmoittaa

- montako erilaista (ei-isomorfista) n -solmuista verkkoa on olemassa
- moniko niistä on yhtenäinen
- monessako niistä on Eulerin kierros
- monessako niistä on Hamiltonin kierros

Ratkaisu:

Tiedostossa `Verkkotutkimus.java` on tehtävänantoa vastaava ohjelma.

Tässä ovat tulokset tapauksiin 1–7:

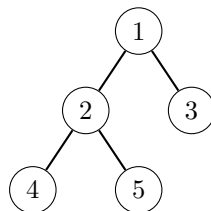
n	verkkoja	yhtenäisiä	Euler	Hamilton
1	1	1	1	1
2	2	1	1	0
3	4	2	2	1
4	11	6	3	3
5	34	21	7	8
6	156	112	15	48
7	1044	853	52	383

6. Suunnittele polynomiaikainen algoritmi, joka tarkistaa, ovatko kaksi puuta isomorfisia. Verkko on puu, jos se on yhtenäinen ja syklitön.

Ratkaisu:

Oletetaan aluksi, että yksi puun solmuista on juuri. Nyt puusta voi muodostaa yksikäsitteisen kuvauksen, jossa jokaisen solmun kuvaus on muotoa " $[A_1][A_2] \dots [A_n]$ ", jossa A_k on solmun lapsisolmun kuvaus. Jos solmulla ei ole lapsisolmuja, niin sen kuvaus on "X". Lisäksi lapsisolmujen kuvaukset ilmoitetaan aakkosjärjestyksessä, jotta kuvaus on yksikäsitteinen.

Esimerkiksi seuraavan puun kuvaus on " $[X][[X][X]]$ ":



Nyt kaksi puuta ovat isomorfisia tarkalleen silloin, kun niiden kuvaus on sama. Kuvaus on mahdollista muodostaa polynomisessa ajassa.

Tutkitaan sitten tilannetta, jossa puiden juuria ei ole valittu. Nyt riittää valita yksi ensimmäisen puun solmuista juureksi ja käydä läpi kaikki mahdolliset toisen puun juuret. Jos jollakin valinnalla juurelliset puut ovat isomorfiset, niin alkuperäiset puut ovat isomorfiset. Eri vaihtoehtoja juuriksi on polynominen määrä (solmujen määrä), joten kokonaisalgoritmi on polynominen.