

Tietorakenteet 2018

Harjoitukset 11, ratkaisut (Viikko 47)

1. Tiedostossa Graphs.java (ei käänny).
2. a) Vaakaan tulosolmu, pystyyn lähtösolmu; eli ensimmäisellä vaakarivillä ovat kaikki solmusta A lähtevät kaaret.

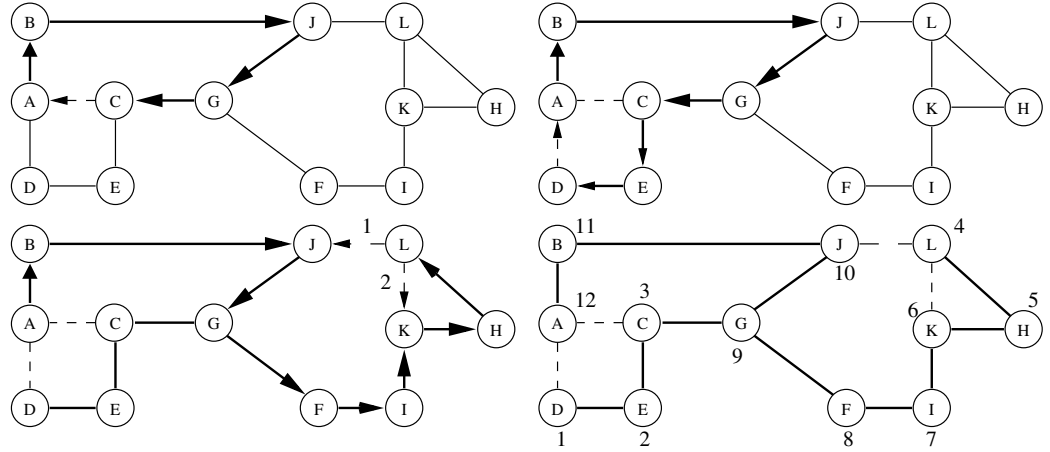
	A	B	C	D	E	F	G	H	I	J	K	L
A		AB		AD								
B				BC		BE						
C	CA											
D						DF				DJ		
E										EJ		
F							FG					
G				GD								
H						HF	HG					
I									IH		IK	
J										JI		
K											KK	
L												

- b) Kaarilistassa AB, AD, BC, BE, CA, DF, DJ, EJ, GD, HF, HG, IH, IK, JI, KK ja jokaisessa kaaressa viite solmuihin, joita kaari yhdistää. Solmulistassa solmut A,...,L.
- c) Kaarilistan lisäksi jokaisessa solmussa on tieto lähtevistä ja tulevista kaarista (esimerkiksi listassa).

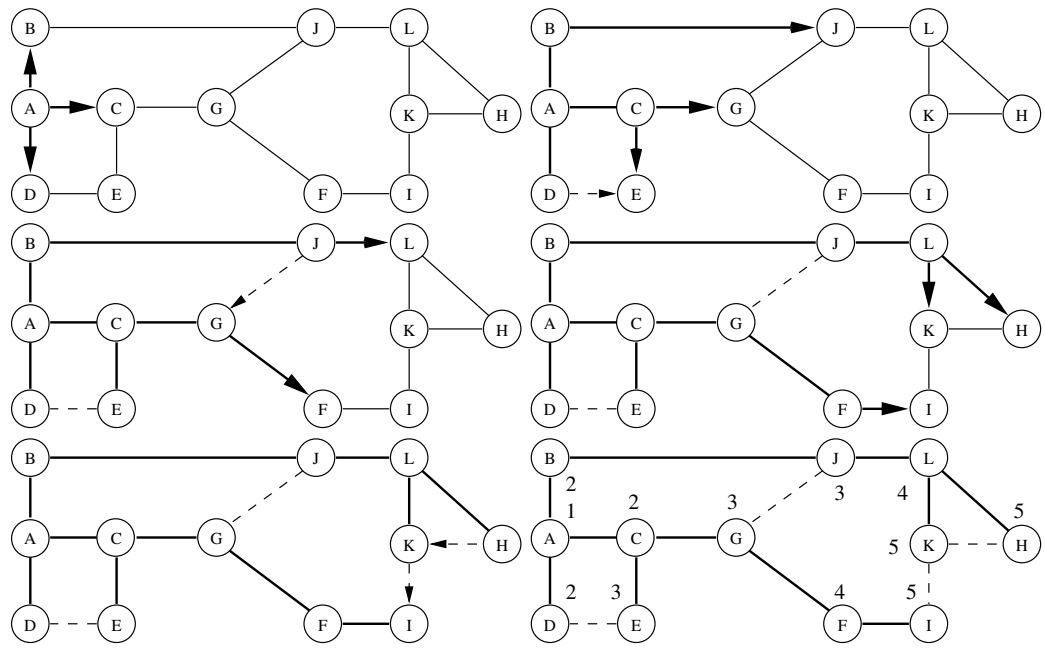
solmu	IN	OUT	solmu	IN	OUT
A	CA	AB, AD	G	FG, HG	GD
B	AB	BC, BE	H	IH	HF, HG
C	BC	CA	I	JI	IH, IK
D	AD, GD	DF, DJ	J	DJ, EJ	JI
E	BE	EJ	K	KK, IK	KK
F	DF, HF	FG	L		

3.
 - a) Ei, L on erikseen.
 - b) Ei kumpikaan, koska graafi sisältää syklejä
 - c) $C, e, A, e, B, e, E, e, J, e, I, e, K$ ja $C, e, A, e, D, e, J, e, I, e, K$
4. Aligraafi muodostuu joistakin graafin solmuista ja joistakin niihin liittyvistä kaarista. Ts. $V' = \{A, B\}$, $E' = \{AB\}$ on aligraafi ja myös $V' = \{A, B\}$, $E' = \emptyset$ on aligraafi. Aligraafista on siis kerrottava mitkä solmut ja särmät siihen kuuluvat.
 - a) esim. $V' = \{A, B, C\}$, $E' = \{AB, BC\}$
 - b) Graafissa on 2 komponenttia ja ne ovat $\{L\}$ ja $V - \{L\}$ (muut solmut).
 - c) ABC, DFG, DJIHG, DJIHFG, KK.

5. a) Syvyysshaun toimintaperiaate lyhyesti: Edetään lähtösolmusta mahdollisimman kauas niin, että edetään vain solmuihin, joissa ei olla jo käyty. Solmut merkitään käydyksi niihin tultaessa. Kaaret, joita pitkin edetään merkitään *löytökaariksi* (discovery edge) ja kaaret, joita pitkin löydetään jo käyty solmu *paluukaariksi* (back edge). Haun aikana edetään aina solmusta ensimmäistä käymätöntä kaarta pitkin, jonka päässä on käymätön solmu. Loput kaarista kokeillaan, kun ollaan peruutettu takaisin. Kun polulla ei päästä enää etenemään, peruutetaan sitä pitkin, kunnes voidaan edetä johonkin käymättömään solmuun ja jatketaan jälleen etenemistä. Viimeisessä kuvassa solmut on numeroitu syvyysshaun päättymisjärjestyksessä (solmu 1. tuli ensin valmiiksi eli siitä ei enää päästy eteenpäin, vaan jouduttiin peruuttamaan).



- b) Leveyshaussa jatketaan jokaista edellisellä kierroksella löydettyistä solmuista eteenpäin kaikkia solmujen kaaria pitkin. Jos kaarta pitkin edetään solmuun, joka on jo löydetty, merkitään kaari *vastakkaiskaareksi* (cross edge). Kaari jota pitkin edetään merkitään *löytökaareksi*. Viimeisessä kuvassa on merkitty solmujen tason (sen kierroksen, jolla ne on löydetty) numerot.



6. a) Esim. $V' = \{B, A, D\}, E' = \{AB, AD\}$ (täytyy olla yhtenäinen, eikä saa olla syklejä)
- b) Esim. $V' = \{B, A, D, J, L, K\}, E' = \{AB, AD, BJ, LK\}$ (ei saa olla syklejä, ei tarvitse olla yhtenäinen)
- c) Saadaan esimerkiksi poistamalla kaaret AD, AC, GJ ja HL.

7. Oheinen syvyyshakuun perustuva algoritmi IsCyclic löytää syklit suunnatusta graafista. Algoritmin toinen osa CycleDFS tutkii, onko graafissa sykliä jollain polulla alkaen solmusta v . IsCyclic toistaa algoritmia CycleDFS käymättömille solmuille, kunnes koko graafi on käyty läpi.

Algoritmin toimintaideana on merkitä graafin solmu, joista se on edennyt eteenpäin ja käydä syvyysshaun avulla läpi rekursiivisesti kaikki solmut joihin päästään nykyisestä solmusta kaaria pitkin. Jos solmusta päästään jotain reittiä takaisin itseensä on graafissa sykli. Tämä voidaan testata merkkamalla kesken ne solmut, joita pitkin hakupolku kulkee. Jos haku osuu solmuun, joka kuuluu johonkin sykliin, niin solmu ei voi tulla valmiiksi ennen kuin jossain vaiheessa ollaan löydetty sykli. Huomaa, että yksikään syklin solmu ei ole voinut tulla valmiiksi (ja näin katkaista hakupolkua, joka kulkee syklin läpi) ennen kuin algoritmi löytää sen, koska tällöin algoritmi olisi aloittanut syklin läpikäynnin kyseisestä solmusta ja löytänyt syklin silloin.

IsCyclic (G)

alkuehto: $G = (V, E)$ on graafi, jolla solmut $V = \{v_1, v_2, \dots, v_n\}$ ja särmät $E = \{e_1, e_2, \dots, e_m\}$.

jättöehto: Palauttaa true, jos graafi sisältää syklin, muutoin palauttaa false.

```
1: merkitse graafin solmut  $V$  käymättömiksi
2: cyclic  $\leftarrow$  false
3:  $i \leftarrow 1$ 
4: while  $i \leq n$  and not cyclic do /*Käydään graafin komponentit läpi*/
5:   if  $v_i$  on käymätön then
6:     cyclic  $\leftarrow$  CycleDFS( $v_i$ )
7:   end if
8:    $i \leftarrow i + 1$ 
9: end while
10: return cyclic
```

CycleDFS (v)

```
1: merkitse solmu  $v$  kesken
2: Olkoot  $k_1, k_2, \dots, k_j$  kaikki solmusta  $v$  lähtevät kaaret
3: cyclic  $\leftarrow$  false
4:  $i \leftarrow 1$ 
5: while  $i \leq j$  and not cyclic do
6:   if kaaren  $k_i$  loppusolmu  $w$  on käymätön then
7:     cyclic  $\leftarrow$  CycleDFS( $w$ )
8:   else if kaaren  $k_i$  loppusolmu  $w$  on kesken then
9:     cyclic  $\leftarrow$  true
10:  end if
11:   $i \leftarrow i + 1$ 
12: end while
13: merkitse solmu  $v$  käydyksi
14: return cyclic
```