

## Tietorakenteet 2018

### Harjoitukset 12 (Viikko 49)

- **Huomioikaa, että yliopiston uuden tutkintosäännöksen mukaan kursista voi saada arvosanan hylätty rekisteriin.**

Mikäli opiskelija ei osallistu opetukseen eikä peru kurssipaikkaansa tai keskeyttää kurssin, hänen opintosuorituksensa arvioidaan arvosanalla hylätty.

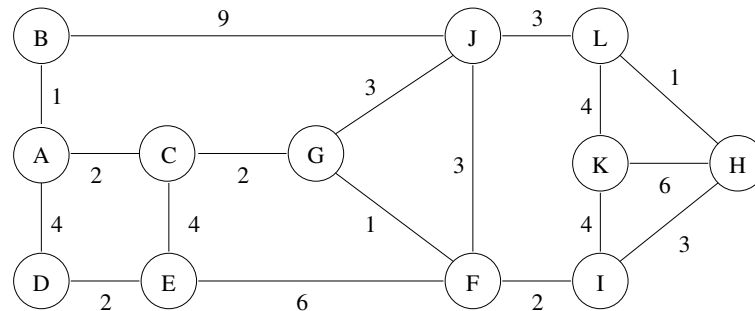
- **Harjoitusryhmiin osallistuvien opiskelijoiden tulee olla paikalla ennen kuin harjoitusryhmä alkaa (klo 12.15/14.15/16.15). Myöhässä tulevat opiskelijat eivät saa rasteja tehdyistä tehtävistä.**
- **Katsokaa hyvissä ajoin ennen harjoitusryhmään tuloa ratkaistujen tehtävien numerot! Näin säästetään aikaa rastilistan täyttämisessä.**
- **Huomatkaa, että pseudokoodi ei tarkoita samaa kuin Java-koodi. Pseudokoodi on ohjelmointikielestä riippumaton esitys algoritmista.**

- 1.-2. Tiedostossa *Graph.java* on toteutettuna suuntaamaton graafi vierekkyyssmatrisitoteutuksena ja joitakin sen operaatioita (mm. syvyyshaku). Solmujen ja särmien merkitseminen on toteutettu niille tehtyjen luokkien sisältämien *marker*-muuttujien avulla.

Tehtävänä on toteuttaa leveyshaku jonon avulla. Aiemmin kurssilla toteutettu jono on annettuna tiedostossa *Queue.java*. Ohje: Leveyshaun toteutus on lähes identtinen syvyysshaun kanssa, tosin rekursion sijaan käytetään jonoa while-silmukassa. Jonoa voidaan hyödyntää solmujen läpikäymisessä seuraavalla tavalla: Aloita lisäämällä haun aloitussolmu jonoon ja toista seuraavaa kunnes jono on tyhjä: poista jonosta poistovuorossa oleva solmu  $v$ , tarkasta käsittelemättömät solmuun  $v$  liittyvät kaaret ja lisää käymättömät solmut jonon loppuun.

Testaa algoritmiasi joillain yksinkertaisilla graafeilla käyttäen testiohjelmaa *GraphTest.java*.

3. Etsi oheisessa graafissa lyhimät polut alkaen solmusta H käyttäen Dijkstran algoritmia.



4. Muodosta tehtävän 3 graafille minimivirittävä puu Kruskalin algoritmilla.
5. Muodosta tehtävän 3 graafille minimivirittävä puu Primin-Jarnikin algoritmilla.
6. Kuvaa miten brute-force match sovittaa merkkijonoa *ababc* merkkijonoon *abcbabcbababcbab*.
7. Muodosta *KMP*-algoritmin epäonnistumisfunktio merkkijonolle *abcababcd* ja sovita sitä *Knuth-Morris-Pratt* -algoritmilla merkkijonoon *abcabcbabcbababcdacd*.