

## Verkkojen vahvasti yhtenäiset komponentit

### 4. Viikkoraportti

Vertaispalautteessa oli paljon sellaista, minkä itsekkin tiesin puutteelliseksi, koska projekti ei ole edennyt ihan niin kuin kurssin ohjeissa ehkä toivottiin/edellytettiin. Palautteen perusteella selvensin määrittelydokumenttia ja lisäsin kommentteja algoritmien luokkiin.

Koska harjoitustyössä ei saanut käyttää valmiita Javan valmiita tietorakenteita ohjelmoin Pino-luokan Javan LinkedList-luokan tilalle ja HakuPuu-luokan Javan TreeSet-luokan tilalle. Luokat eivät luonnollisestikaan korvaa Javan valmiita luokkia vaan toteuttavat niistä ainoastaan tarvittavat toiminnot. Näitä yksinkertaisia luokkia on mukava koodailla, tai ainakin työ tuntuu edistyvän, ja niille oli myös suhteellisen helppoa tehdä JUnit-testejä.

Lisäksi ohjelmoin algoritmiluokat palauttamaan verkon vahvasti yhtenäiset komponentit String merkkijonona esim.  $\{\{0\}\{1,2,5\}\{3,4\}\{6,7\}\}$ . Tätä voin hyödyntää kun myöhemmin testaan kokonaisia algoritmiluokkia.

Suorituskykymittauksia aloitellessani ihmettelin, miksi ensimmäisenä ajettu yhden solmun verkko on 10 kertaa hitaampi kuin seuraavana ajettu 5 solmun verkko. Lisäsin ajolistaan toisen yhden solmun verkon ja havaitsin, että tämän toisen verkon suoritus aika vastaa paremmin sitä mitä suuremmat verkot antoivat odottaa. Ilmeisesti jokin virittelee systeemejä kuntoon ohjelman käynnistyksen jäljiltä.

Isompi ongelma on että yli 2500 solmun verkoilla syvyysuuntaisen läpikäynnin rekursiopino on täynnä. Jos halutaan isompi verkkoja vaihtoehtona voisi olla algoritmin toteuttaminen iteratiivisesti. Tai sitten pitäisi osata lisätä ohjelman resursseja.

Testausdokumenttia aloitin kirjoitta viime viikon TestiMini-dokumentin pohjalta, johon lisäsin tietoa uusien apuluokkien yksikkötestauksista ja integraatiotestauksesta.

Toteutusdokumenttiin kirjoitin ohjelmiston yleisrakenteen ja luonnostelin dokumentin rakenteen.

Kurssi loppuu ensi viikolla, joten kaikki olisi saatava valmiiksi. Lienee syytä hieman priorisoida tekemättömiä asioita. Automaattisia testejä lisää varsinkin algoritmiluokille. Koodia syytä siistiä ja pitkiä metodeja jakaa useammaksi pienemmäksi. Käydä koodi läpi ja selkeyttää algoritmit.

Luonnollisesti täytyy valmistautua torstain demo.

Sitten on syytä kirjoittaa dokumentit valmiiksi. IRCiä satunnaisesti seuraavana mielestäni TeaShark lupasi että lopullista palautusta olisi jatkettu sunnuntaihin 16.6. asti. Pitääkö tämä paikkansa vai uskonko edelleen kurssin nettisivua?