

Ilkka Lehikoinen  
013787637

Aineopintojenharjoitustyö:  
Tietorakenteet ja algoritmit

## Verkon vahvasti yhtenäiset komponentit

### Testausdokumentti

#### Yksikkötestaus

Apuluokkien Solmu, Pino, PuuSolmu ja HakuPuu metodit ovat testatut JUnitilla niitä varten tehdyillä testitapauksilla, jotka ottavat huomioon monia erilaisia tapauksia.

Koska algoritmiluokkani (Kosaraju, Tarjan ja PathBased) eivät palauta mitään en ole keksinyt niille, enkä monelle niiden metodeista, vielä automaattitestausta, joten tässä raportissa kuvailen, miten olen manuaalisesti testannut nämä algoritmit.

Olen eri lähteistä kopioinut kolme suunnattua kahdeksan solmun verkkoa, joiden vahvasti yhtenäiset komponentit on tunnettu.

<code>{0,1,4},{2},{3,7},{5,6}</code>	<code>{0,1,4},{2,3},{5,6},{7}</code>	<code>{0,1,4},{2,3},{5,6},{7}</code>
<code>static int[][] esim1 = new int[][] {</code>	<code>static int[][] esim2 = new int[][] {</code>	<code>static int[][] esim3 = new int[][] {</code>
<code>    {0, 0, 0, 0, 1, 0, 0, 0},</code>	<code>    {0, 1, 0, 0, 0, 0, 0, 0},</code>	<code>    {0, 0, 0, 0, 1, 0, 0, 0},</code>
<code>    {1, 0, 0, 0, 0, 0, 0, 0},</code>	<code>    {0, 0, 1, 0, 1, 0, 0, 0},</code>	<code>    {1, 0, 0, 0, 0, 0, 0, 0},</code>
<code>    {0, 1, 0, 0, 0, 0, 1, 0},</code>	<code>    {0, 0, 0, 1, 0, 0, 1, 0},</code>	<code>    {0, 0, 0, 1, 0, 0, 0, 0},</code>
<code>    {0, 0, 1, 0, 0, 0, 0, 1},</code>	<code>    {0, 0, 1, 0, 0, 0, 0, 1},</code>	<code>    {0, 0, 1, 0, 0, 0, 0, 0},</code>
<code>    {1, 1, 0, 0, 0, 0, 0, 0},</code>	<code>    {1, 0, 0, 0, 0, 1, 0, 0},</code>	<code>    {0, 1, 0, 0, 0, 0, 0, 0},</code>
<code>    {0, 1, 0, 0, 1, 0, 1, 0},</code>	<code>    {0, 0, 0, 0, 0, 0, 1, 0},</code>	<code>    {0, 1, 0, 0, 0, 0, 1, 0},</code>
<code>    {0, 0, 0, 0, 0, 1, 0, 0},</code>	<code>    {0, 0, 0, 0, 0, 1, 0, 1},</code>	<code>    {0, 0, 1, 0, 0, 1, 0, 0},</code>
<code>    {0, 0, 0, 1, 0, 0, 1, 0}};</code>	<code>    {0, 0, 0, 0, 0, 0, 0, 1}};</code>	<code>    {0, 0, 0, 1, 0, 0, 1, 1}};</code>

Algoritmiluokkien testauksessa käytettyjä vierusmatriiseja.

Näitä verkkoja olen käyttänyt syötteenä kullekin algoritmille ja todennut että ne antavat oikean vastauksen.

Lisäksi olen generoinut erikokoisia ja -tiheyksisiä satunnaisia verkkoja ja todennut, että kukin algoritmi antaa vahvasti yhtenäiset komponentit, joissa jokainen solmu esiintyy jossakin komponentissa ja kukin solmu esiintyy ainoastaan yhdessä komponentissa. Lisäksi olen verrannut algoritmien antamaa vastausta toisiinsa ja todennut, että kaikki kolme algoritmia antaa samalle verkolle samat vahvasti yhtenäiset komponentit.

Tämän perusteella olen varma, että kaikki kolme algoritmia löytävät suunnatusta verkosta sen vahvasti yhtenäiset komponentit.

Nyt kun algoritmiluokat ja osa niiden metodeista palauttavat String-merkkijonoja niin algoritmiluokille on tarkoitus ohjelmoida myös automaattisia JUnit-testejä.

VYK-luokan, jonka tärkein tehtävä on olla eräänlainen komentokeskus algoritmiluokkien testauksessa, testaus on keskeneräinen.

## ***Integrointitestaus***

VYK-luokassa manuaalisesti tehtyjä verkkoja ja generoituja satunnaisverkkoja on annettu kullekin algoritmiluokalle ja nämä ovat kaikki aina palauttaneet samat vahvasti yhtenäiset komponentit. Manuaalisesti tehdyillä verkoilla nämä komponentit ovat olleet oikein.

## ***Järjestelmätestaus***

- toimiiko kokonaisuus niin kuin vaatimusdokumentissa sanotaan?

## ***Regressiotestaus***

- muutosten ja bugikorjausten jälkeen varmistetaan ettei muutokset riko mitään ehjää.

## ***Lähteet:***

[1]

[2]