

Verkon vahvasti yhtenäiset komponentit

Määrittelydokumentti

Suunnattua verkkoa, jossa jokaisesta solmusta on polku jokaiseen solmuun, sanotaan vahvasti yhtenäiseksi. Jos verkko ei ole vahvasti yhtenäinen niin se voidaan jakaa vahvasti yhtenäisiin komponentteihin, eli sellaisiin osiin, joissa jokaisesta solmusta on polku jokaiseen saman vahvasti yhtenäisen komponentin solmuun.

Kukin solmu voi kuulua vain yhteen vahvasti yhtenäiseen komponenttiin. Jos kaksi solmua kuuluvat samaan vahvasti yhtenäiseen komponenttiin, niin niiden välillä on polut kumpaankin suuntaan. Ja toisaalta, jos ne eivät kuulu samaan vahvasti yhtenäiseen komponenttiin, niin niiden välillä ei ole polkua kumpaankaan ta toiseen suuntaan.

Monet suunnatuissa verkoissa (Directed Graph) toimivat algoritmit jakavat verkon ensiksi vahvasti yhtenäisiin komponentteihin (Strongly Connected Components), jonka jälkeen varsinainen algoritmi suoritetaan kullekin komponentille erikseen ja lopuksi tulokset yhdistetään komponenttien rakenteen mukaan. [1, s.615] Tämän vuoksi verkon vahvasti yhtenäisten komponenttien tehokas löytäminen on tärkeää.

Harjoitustyössä on tarkoitus löytää suunnatusta verkosta vahvasti yhtenäiset komponentit mahdollisimman tehokkaasti. Käyttötarkoitus määrittää mitä tarkoitetaan tehokkuudella,. Joskus säästeliäs muistinkäyttö on tehokkuutta, mutta tässä harjoitustyössä keskitytään algoritmien nopeuteen. Nopeutta mitataan generoimalla erikokoisia ja -tiheyksisiä satunnaisia suunnattuja verkkoja ja etsimällä sitten kullakin algoritmilla sen vahvasti yhtenäiset komponentit. Nopeutta mittaava metodi laskee algoritmin suoritukseen kuluvan ajan. Harjoitustyössä ohjelmoidaan kolme yleistä algoritmia verkon vahvasti yhtenäisten komponenttien löytämiseksi.

Kosaraju

Kosarajun algoritmi on luentomonisteissa [2 s. 500] ja Cormanin kirjassa [1 s. 617] esitelty. Algoritmissa tehdään ensin verkon syvyysuuntainen läpikäynti, jossa solmut laitetaan pinoon niistä poistumisen mukaisessa järjestyksessä. Tämän jälkeen suoritetaan verkolle transpoosi, jossa solmujen välisten kaarien suunnat käännetään vastakkaiseen suuntaan. Seuraavaksi suoritetaan uudelleen verkon syvyysuuntainen läpikäynti alkaen pinon pinnalta. Tällöin aina kun yhdestä solmusta ei enää pääse muihin solmuihin on löydetty vahvasti yhtenäinen komponentti. Näin jatketaan niin kauan kuin pinossa on solmuja, joita ei ole vielä käsitelty.

Jos algoritmi toteutetaan siten, että verkon ovat vieruslistoina, niin algoritmin aikavaativuus on $O(V+E)$, mutta jos verkot ovat vierusmatriiseina, niin aikavaativuus on $O(V^2)$.

Tarjan

Tarjan-algoritmissa[3] suoritetaan syvyysuuntaisia läpikäyntejä, niin kauan kuin verkossa on vielä käsittelemättömiä solmuja. Solmut laitetaan pinoon sitä mukaan kun ne tulevat käsittelyyn. Kun läpikäynti palaa niihin alipuusta, solmu otetaan pinosta ja tutkitaan, onko se jonkin vahvasti

yhtenäisen komponentin juuri, jolloin sitä ennen pinosta poistetut solmut muodostavat vahvasti yhtenäisen komponentin.

PathBased

PathBased-algoritmi[4] suorittaa verkolle ensin syvyys-suuntaisen läpikäynnin pitäen samanaikaisesti yllä kahta pinoa, ensimmäinen pino ylläpitää tietoa nykyisessä komponentissa olevista solmuista ja toinen pino pitää yllä listaa nykyisellä läpikäyntipolulla olevista solmuista. Solmut ovat pinossa siinä järjestyksessä, jossa syvyys-suuntainen läpikäynti ne löysi.

PathBased algoritmi on hyvin samankaltainen Tarjan-algoritmin kanssa, jossa toisen pinon sijaan on taulukko, johon kerätään solmun esijärjestysnumero.

Lähteet:

- [1] Introduction to Algorithms 3rd ed. T.H. Corman, C.E. Leiserson, R.L. Rivest, C. Stein. The MIT Press 2009.
- [2] Luentomonisteet: <http://www.cs.helsinki.fi/u/floreen/tira2013/tira.pdf>
- [3] Tarjan: http://en.wikipedia.org/wiki/Tarjan%27s_strongly_connected_components_algorithm
- [4] PathBased: http://en.wikipedia.org/wiki/Path-based_strong_component_algorithm