

Paikalliset hajautetut verkkoalgoritmit

Mika Laitinen

24. maaliskuuta 2011

- Tyypillisesti tietojenkäsittelytieteessä algoritmit pyritään kehittämään laskennallisesti nopeiksi
 - Yksi kokonaisuus ratkaisee ongelmaa, ja kommunikaatiokustannukset ovat pieniä

- Tyypillisesti tietojenkäsittelytieteessä algoritmit pyritään kehittämään laskennallisesti nopeiksi
 - Yksi kokonaisuus ratkaisee ongelmaa, ja kommunikaatiokustannukset ovat pieniä
- Suuria verkkoja, kuten Internetiä, ihmisaivoja tai sensoriverkkoja on vaikeaa tai mahdotonta hallita keskitetysti
 - Perinteisiin ongelmiin verrattuna kommunikaatiokustannukset ovat merkittävät
 - Laskenta halpaa verrattuna kommunikaatioon

- Tyypillisesti tietojenkäsittelytieteessä algoritmit pyritään kehittämään laskennallisesti nopeiksi
 - Yksi kokonaisuus ratkaisee ongelmaa, ja kommunikaatiokustannukset ovat pieniä
- Suuria verkkoja, kuten Internetiä, ihmisaivoja tai sensoriverkkoja on vaikeaa tai mahdotonta hallita keskitetysti
 - Perinteisiin ongelmiin verrattuna kommunikaatiokustannukset ovat merkittävät
 - Laskenta halpaa verrattuna kommunikaatioon
- Voidaanko tällaisia ongelmia lähestyä erilaisella mallilla?

- Idea: laitetaan jokaiseen verkon solmuun prosessori, jotka voivat kommunikoida vain naapuriprosessoreidensa kanssa
 - Suuntaamaton verkko, jossa prosessorit ovat naapureita keskenään, jos niiden välillä on kaari

- Idea: laitetaan jokaiseen verkon solmuun prosessori, jotka voivat kommunikoida vain naapuriprosessoreidensa kanssa
 - Suuntaamaton verkko, jossa prosessorit ovat naapureita keskenään, jos niiden välillä on kaari
- Jokainen prosessori pyrkii ratkaisemaan paikallisesti oman osansa koko verkkoa koskevasta ongelmasta

- Idea: laitetaan jokaiseen verkon solmuun prosessori, jotka voivat kommunikoida vain naapuriprosessoreidensa kanssa
 - Suuntaamaton verkko, jossa prosessorit ovat naapureita keskenään, jos niiden välillä on kaari
- Jokainen prosessori pyrkii ratkaisemaan paikallisesti oman osansa koko verkkoa koskevasta ongelmasta
- Algoritmin suorituksen alussa jokaiselle prosessorille annetaan rajattu määrä informaatiota
 - Prosessori tietää verkon koostumuksen vain paikallisesti

- Idea: laitetaan jokaiseen verkon solmuun prosessori, jotka voivat kommunikoida vain naapuriprosessoreidensa kanssa
 - Suuntaamaton verkko, jossa prosessorit ovat naapureita keskenään, jos niiden välillä on kaari
- Jokainen prosessori pyrkii ratkaisemaan paikallisesti oman osansa koko verkkoa koskevasta ongelmasta
- Algoritmin suorituksen alussa jokaiselle prosessorille annetaan rajattu määrä informaatiota
 - Prosessori tietää verkon koostumuksen vain paikallisesti
- Kaikki prosessorit ajavat samaa algoritmia

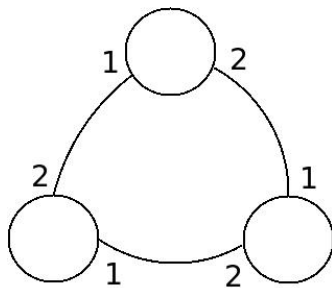


- Mitä informaatiota prosessoreille täytyy antaa?

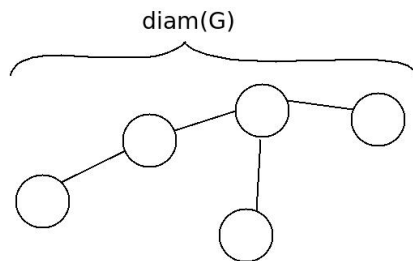
- Mitä informaatiota prosessoreille täytyy antaa?
- Vähimmäisvaatimus: naapureiden kanssa on voitava kommunikoida
- Ensimmäinen ajatus: määritellään, että prosessoreille annetaan naapuriprosessorit järjestettynä listana

- Mitä informaatiota prosessoreille täytyy antaa?
- Vähimmäisvaatimus: naapureiden kanssa on voitava kommunikoida
- Ensimmäinen ajatus: määritellään, että prosessoreille annetaan naapuriprosessorit järjestettynä listana
- Ongelma: prosessorit ajavat samaa algoritmia, ja jos ne ovat symmetrisessä tilanteessa keskenään, niiden täytyy tuottaa sama tulos!

- Symmetrisessä tilanteessa algoritmit saavat saman informaation
 - Deterministinen algoritmi tuottaa tällöin saman vastauksen
 - Symmetria on pakko rikkoa, tai joitain ongelmia ei voida ratkaista



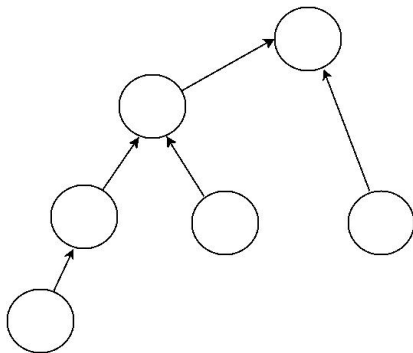
- Symmetrian voi rikkoa satunnaisuuden avulla, mutta determinismistä ei ole pakko luopua
- Yksinkertainen muutos malliin: annetaan jokaiselle prosessorille uniikki tunnistenumero
- \Rightarrow Kaikki ratkeavat ongelmat voidaan ratkaista uudella mallilla $O(\text{diam}(G))$ kommunikaatiokierroksessa



Cole-Vishkin: verkon 6-väritys

- Verkon väritys: löydettävä jokaiselle solmulle väri (kokonaisluku) siten, että millään solmun naapurilla ei ole samaa väriä
- Mitä vähemmän värejä, sen parempi
- Cole-Vishkinin algoritmilla voidaan 6-värittää suunnattu verkko, jossa jokaisella solmulla on maksimissaan yksi jälkeläinen

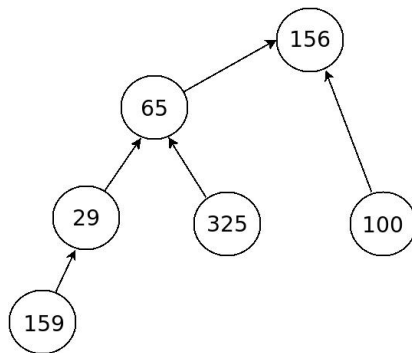
Cole-Vishkin: verkon 6-väritys



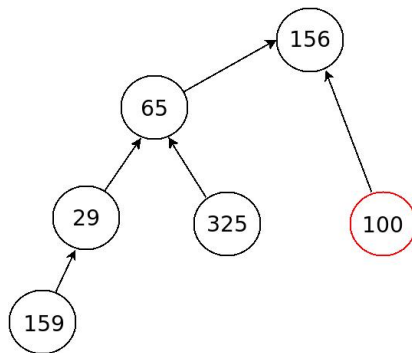
Cole-Vishkin: verkon 6-väritys

- Verkon väritys: löydettävä jokaiselle solmulle väri (kokonaisluku) siten, että millään solmun naapurilla ei ole samaa väriä
- Mitä vähemmän värejä, sen parempi
- Cole-Vishkinin algoritmilla voidaan 6-värittää suunnattu verkko, jossa jokaisella solmulla on maksimissaan yksi jälkeläinen
 - Jokaisella solmulla on myös oltava alussa väri
 - Prosessoreille annetut tunnistenumerot käyvät alkuväreinä
- Lopullisessa verkossa solmuilla voi olla vain kuusi erillistä väriä

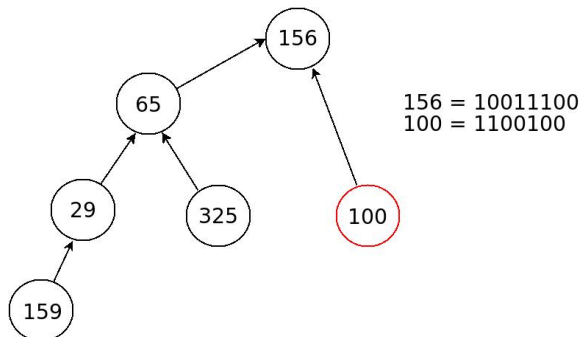
Cole-Vishkin: verkon 6-väritys



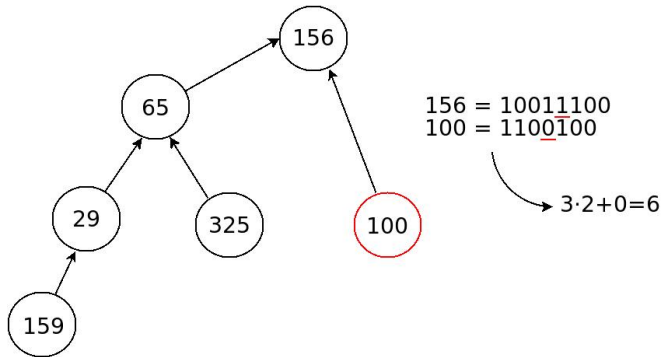
Cole-Vishkin: verkon 6-väritys



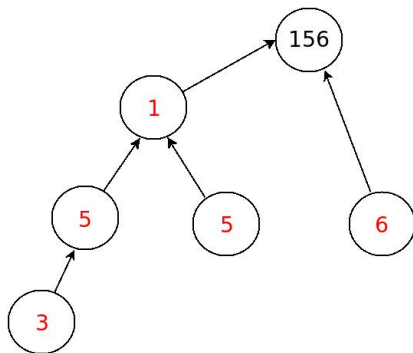
Cole-Vishkin: verkon 6-väritys



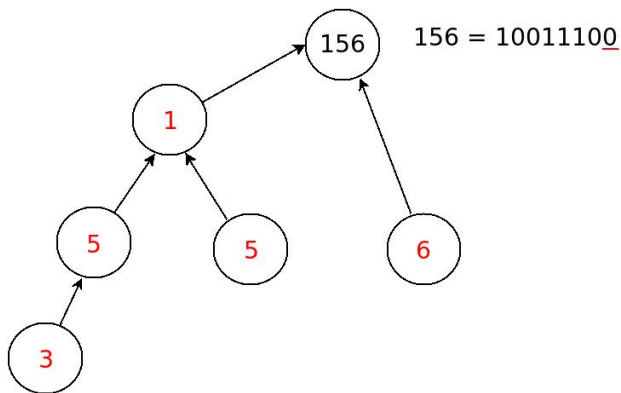
Cole-Vishkin: verkon 6-väritys



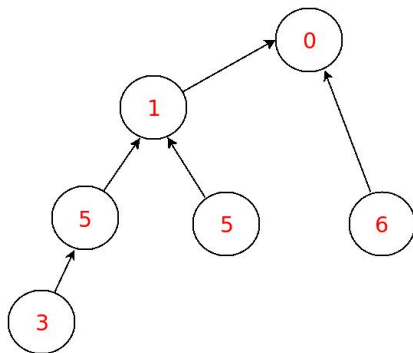
Cole-Vishkin: verkon 6-väritys



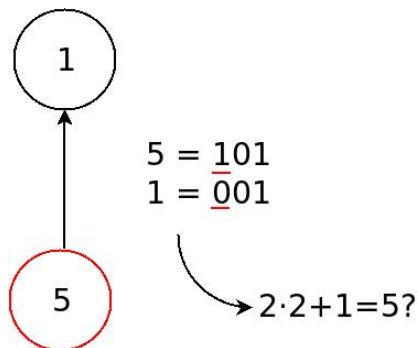
Cole-Vishkin: verkon 6-väritys



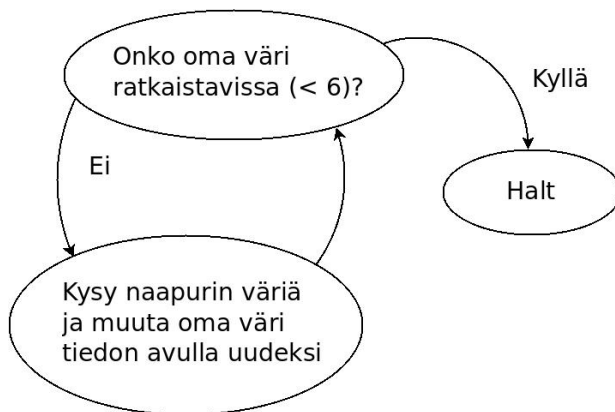
Cole-Vishkin: verkon 6-väritys



Cole-Vishkin: verkon 6-väritys



Cole-Vishkin: verkon 6-väritys



Cole-Vishkin: verkon 6-väritys

- Jokaisella kierroksella värien määrä vähenee ainakin logaritmisesti
- Kommunikaatiokierroksia tarvitaan siis $O(\log^* N)$ kappaletta, jossa N on prosessorien määrä
- Algoritmi toimii, sillä yksikään naapuri ei voi valita kommunikaatiokierroksen jälkeen samaa väriä kuin prosessori itse valitsee, joten algoritmin jokaisessa vaiheessa väritys on käypä

- Käsiteltävän tiedon määrä kasvaa jatkuvasti
 - Paikalliset algoritmit pystyvät ratkaisemaan ainakin joitain ongelmia hyvin nopeasti
 - Prosessoriverkko konstruoitava ennen laskentaa
- Joitain rohkaisevia tutkimustuloksia, sekä biologisia esimerkkejä, kuten aivotoiminta
- Sensoriverkot, mobiiliverkot

Kiitos huomiostanne

- Kysymyksiä?