

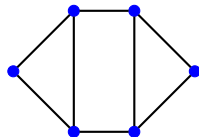
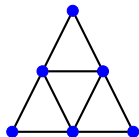
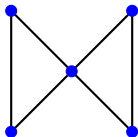
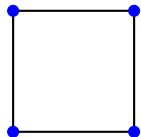
2-yhtenäisyys ja syvyyshaku

March 4, 2017

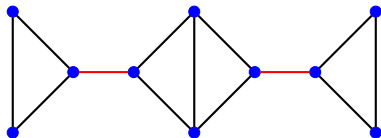
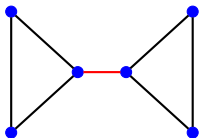
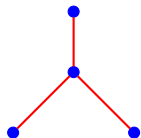
- Verkko on yhtenäinen (connected) jos jokaisesta solmusta voi kulkea jokaiseen solmuun
- Verkko on 2-kaari-yhtenäinen (2-edge-connected) jos minkä tahansa kaaren voi poistaa ja verkko säilyy silti yhtenäisenä

2-kaari-yhtenäisyys

2-kaari-yhtenäisiä verkkoja

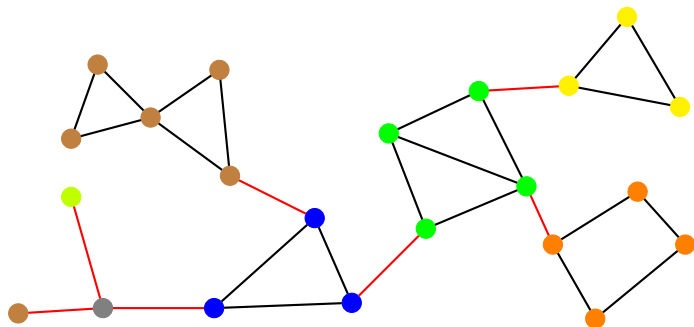


Ei 2-kaari-yhtenäisiä verkkoja



2-kaari-yhtenäisyys

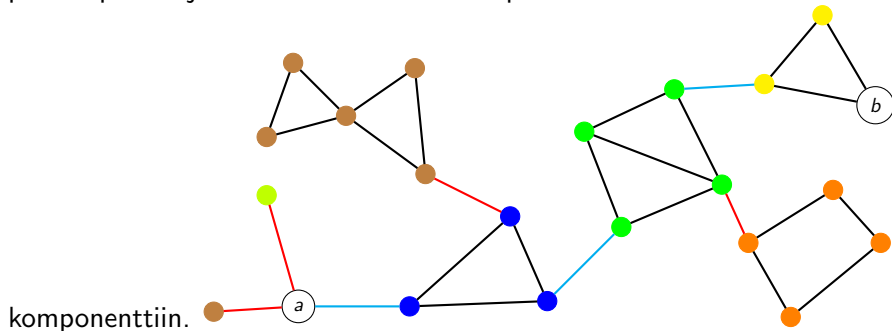
- Kaari jonka poistaminen katkaisee verkon on silta
- 2-kaari-yhtenäinen verkko on siis verkko jossa ei ole yhtään siltaa
- Jos kaikki sillat poistetaan, jäljelle jäävät komponentit ovat 2-kaari-yhtenäisiä
- Sillat ja 2-kaari-yhtenäiset komponentit muodostavat puurakenteen



Esimerkki

Esimerkkiongelman: jos kuljetaan solmusta a solmuun b , mitä kaaria pitkin on pakko kulkea?

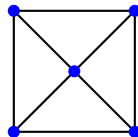
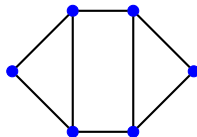
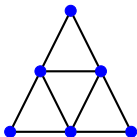
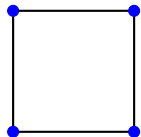
Ratkaisu: kaikkia siltoja jotka ovat 2-kaari-yhtenäisten komponenttien puussa polulla joka menee solmun a komponentista solmun b



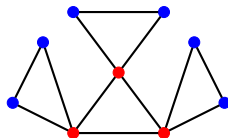
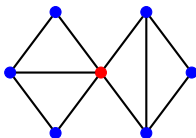
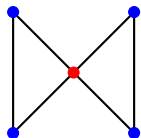
- Verkko on 2-solmu-yhtenäinen (2-vertex-connected) jos minkä tahansa solmun voi poistaa ja verkko säilyy yhtenäisenä
- 2-solmu-yhtenäinen verkko on aina myös 2-kaari-yhtenäinen, mutta ei toisinpäin
- Kirjallisuudessa 2-yhtenäinen (biconnected) tarkoittaa yleensä 2-solmu-yhtenäistä

2-solmu-yhtenäisyys

2-solmu-yhtenäisiä verkkoja

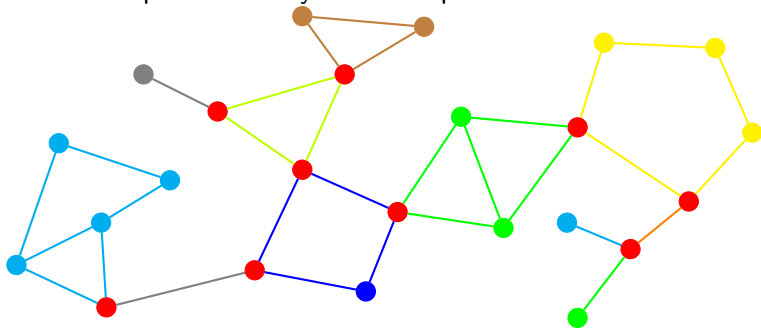


Verkkoja jotka ovat 2-kaari-yhtenäisiä mutta eivät ole 2-solmu-yhtenäisiä



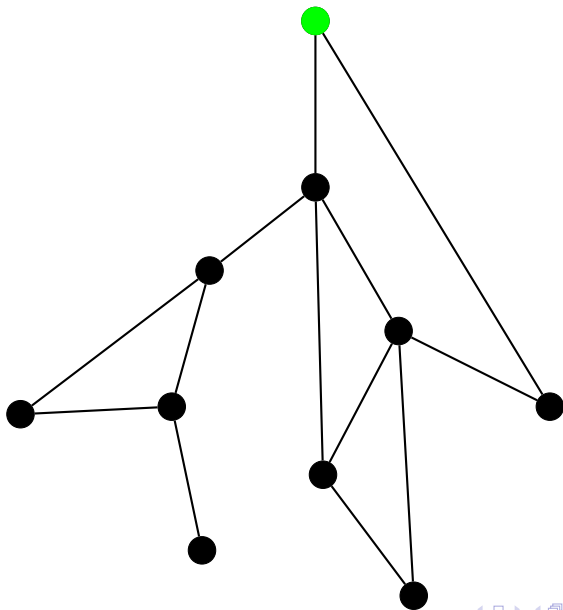
2-solmu-yhtenäisyys

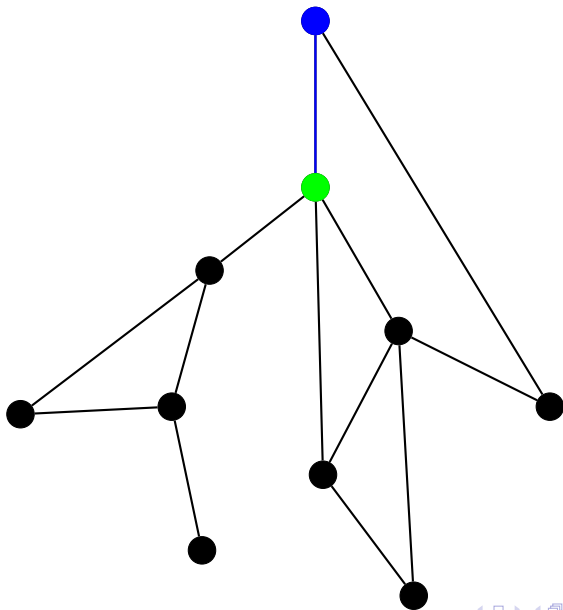
- Solmu jonka poistaminen katkaisee verkon on artikulaatiopiste
- Verkko voidaan jakaa artikulaatiopisteiden perusteella komponenteiksi jotka muodostavat puurakenteen
- Tässä puurakenteessa artikulaatiopisteet kuuluvat useaan komponenttiin, mutta jokainen kaari ja jokainen solmu joka ei ole artikulaatiopiste kuuluu yhteen komponenttiin

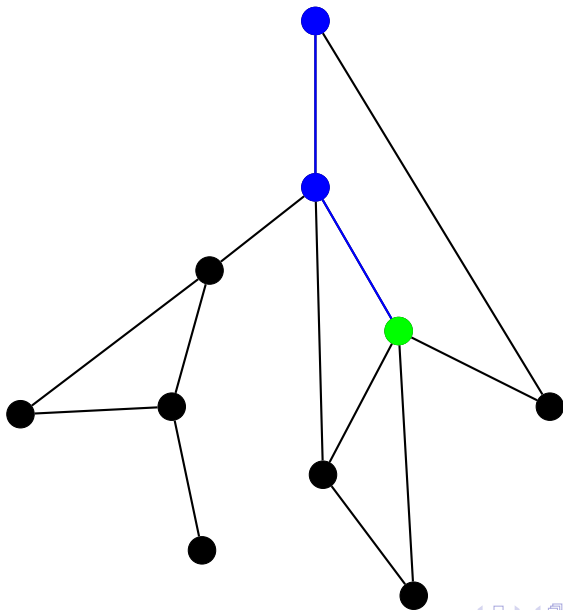


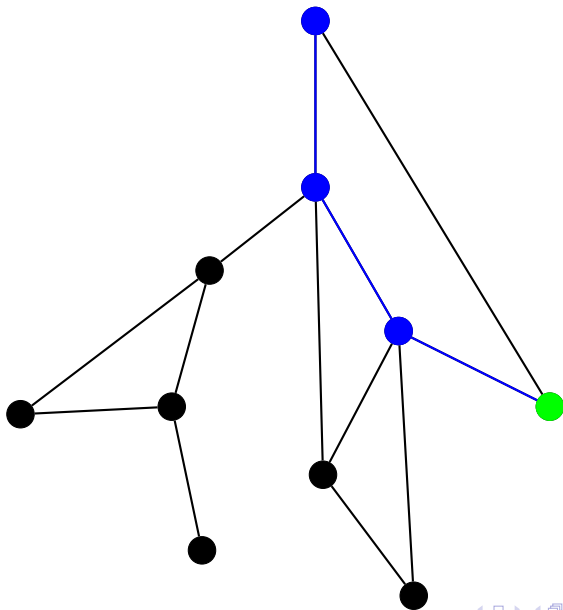
- Syvyyshaku on yksinkertainen verkkoalgoritmi, mutta sen ymmärtäminen ja soveltaminen on tärkeää monessa edistyneemmässä verkkoalgoritmossa

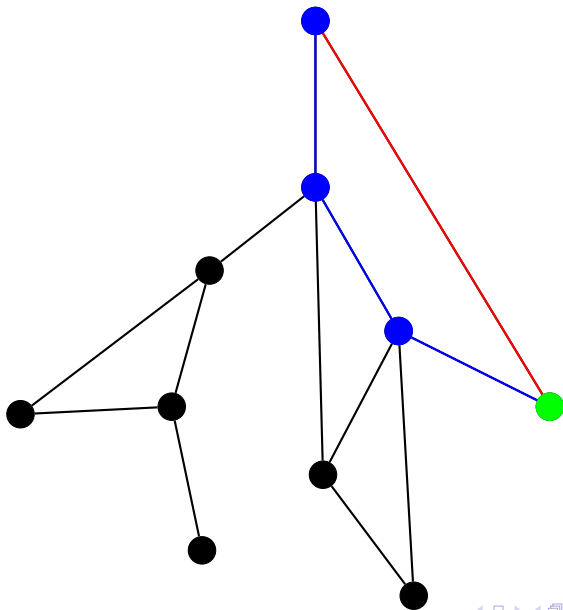
```
void haku(int s) {
    if (z[s]) return;
    z[s] = 1;
    // solmun s käsittely tähän
    for (auto u: v[s]) {
        haku(u);
    }
}
```

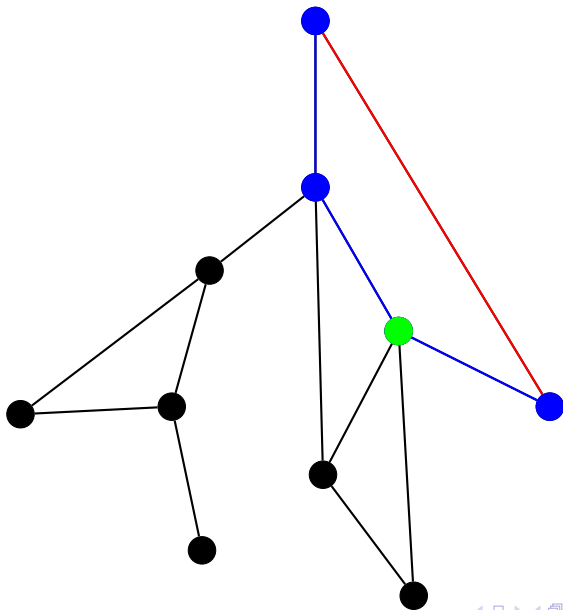


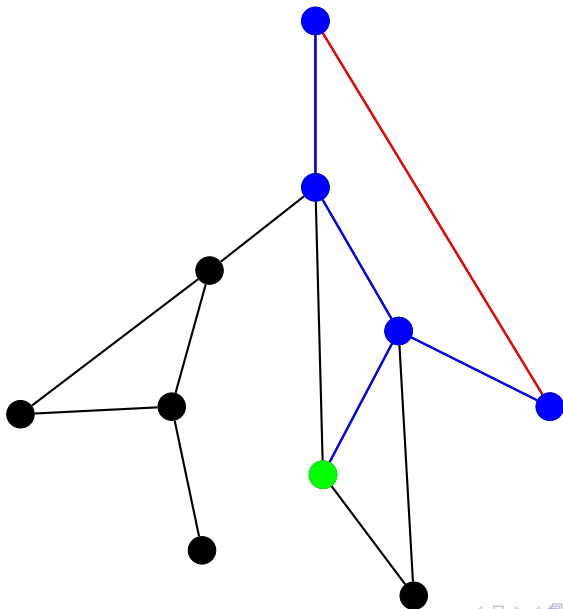


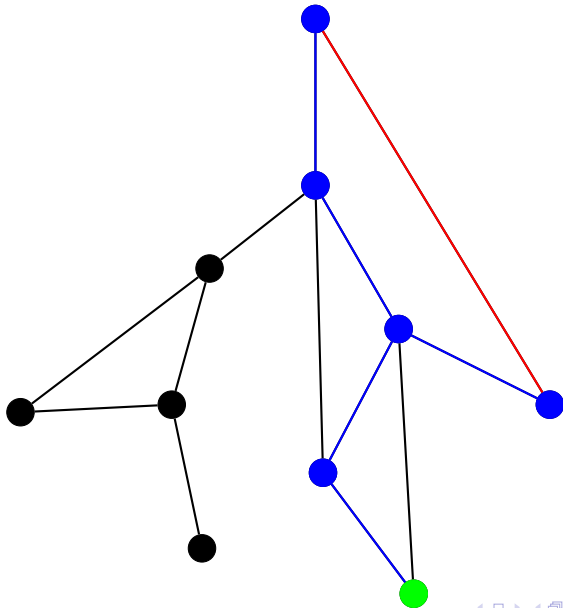


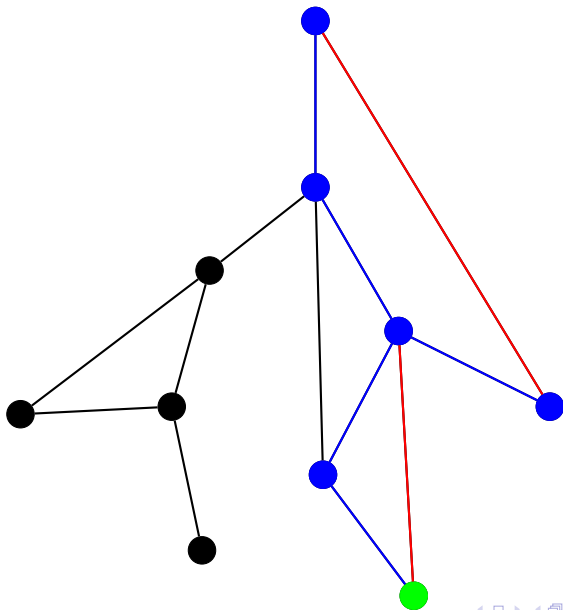


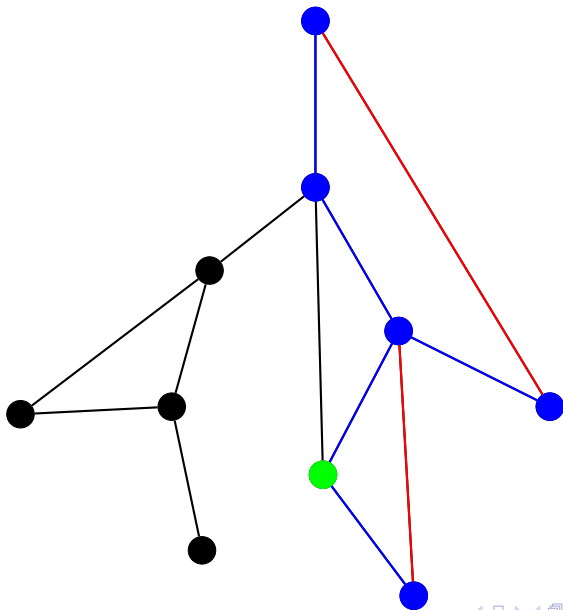


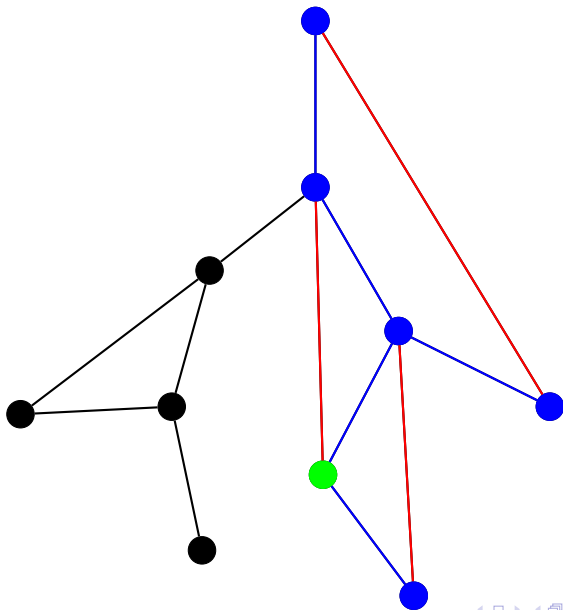


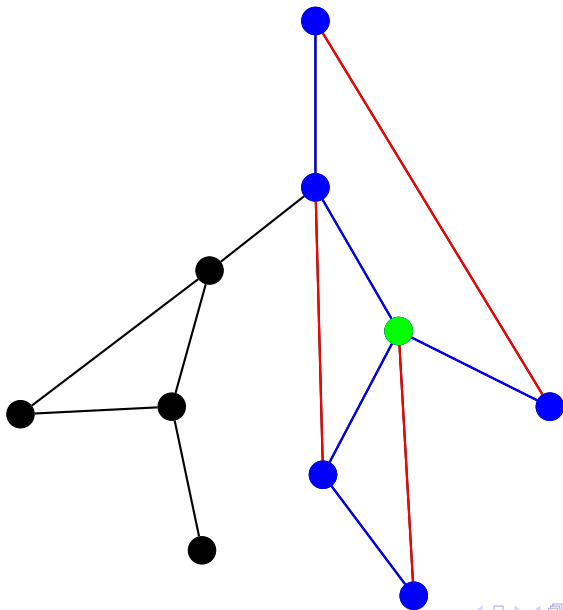


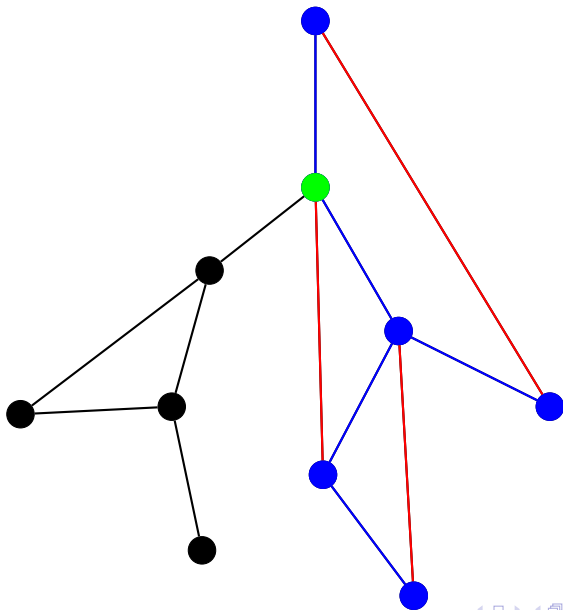


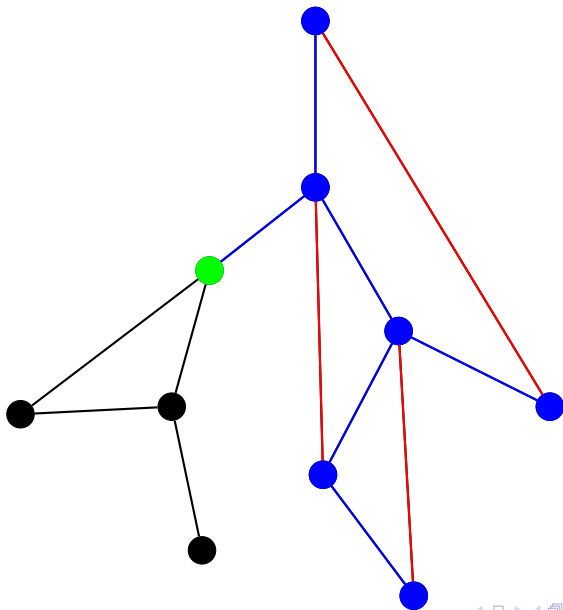


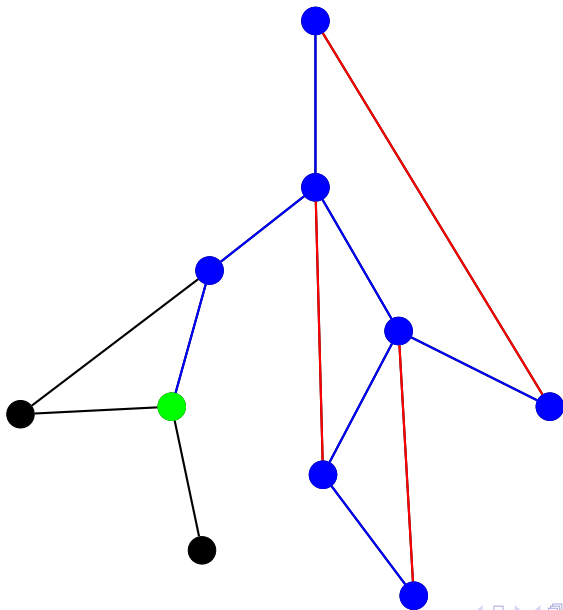


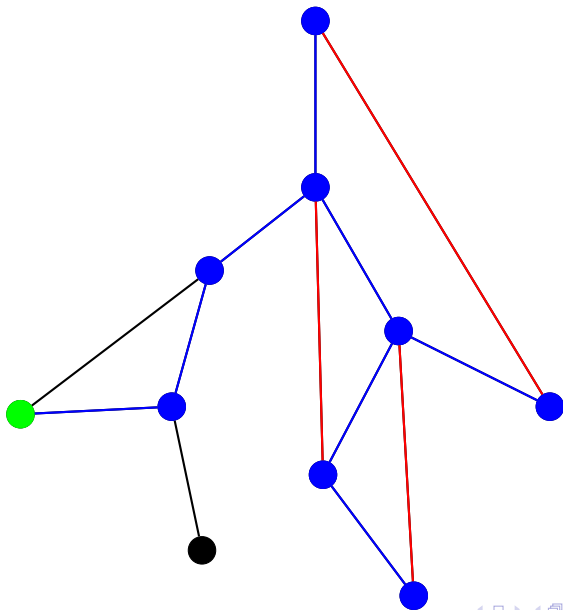


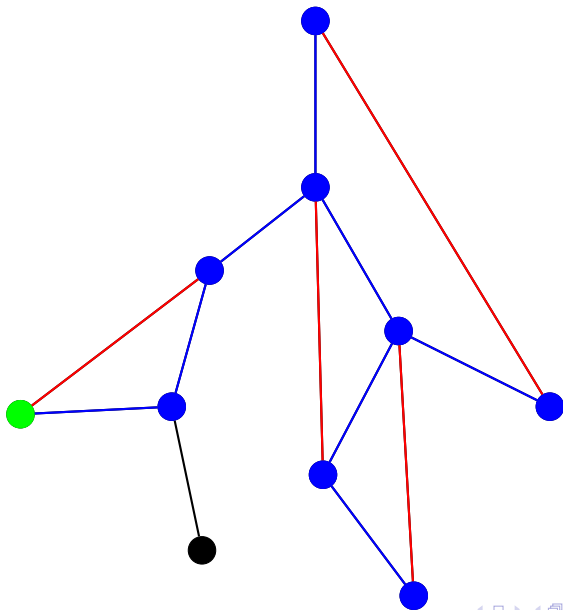


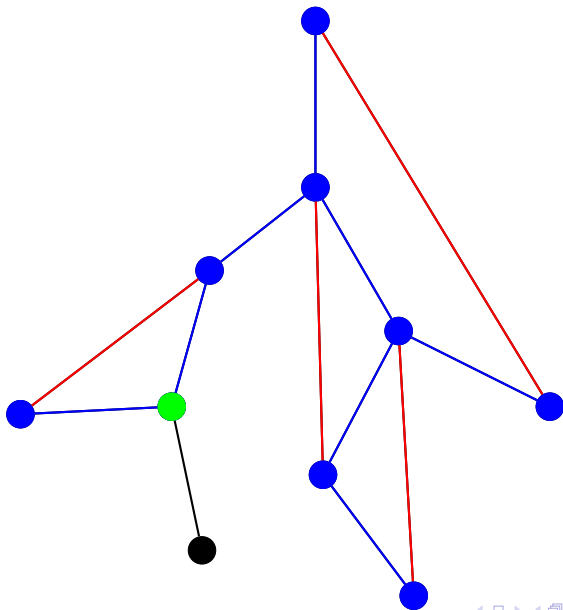


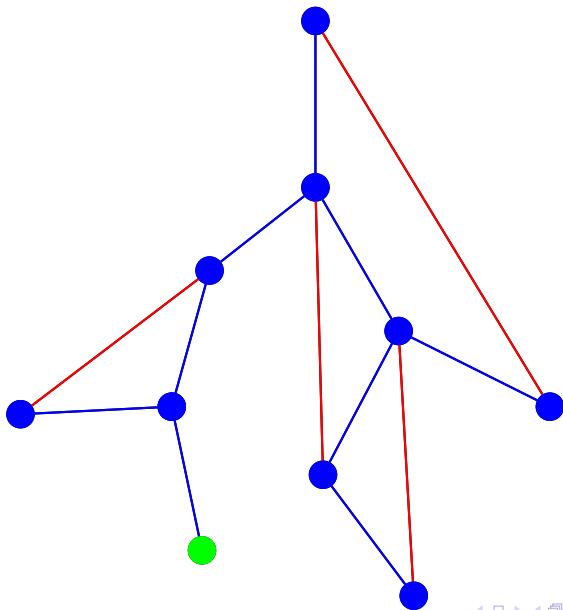


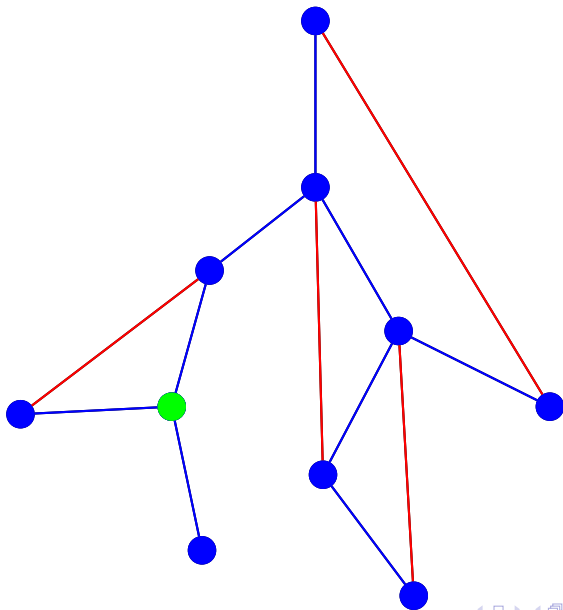


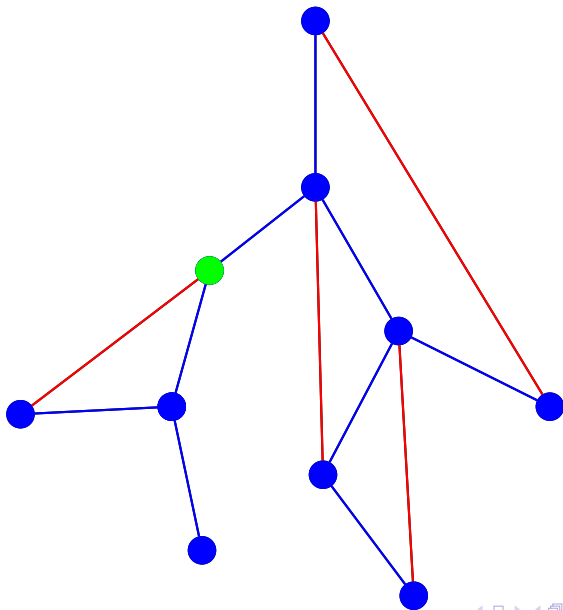


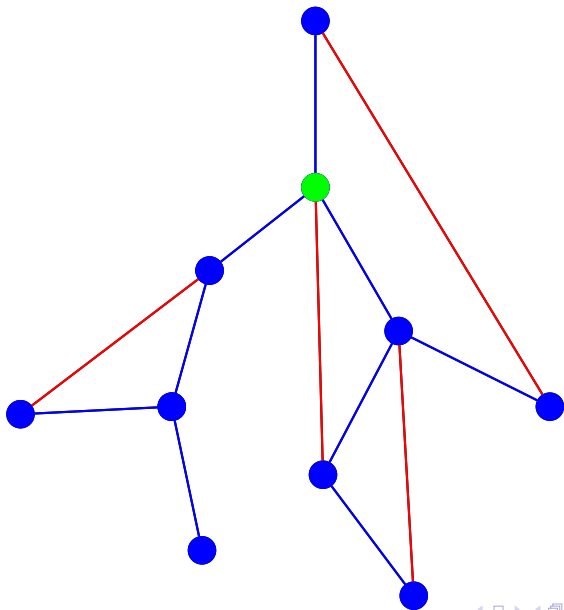


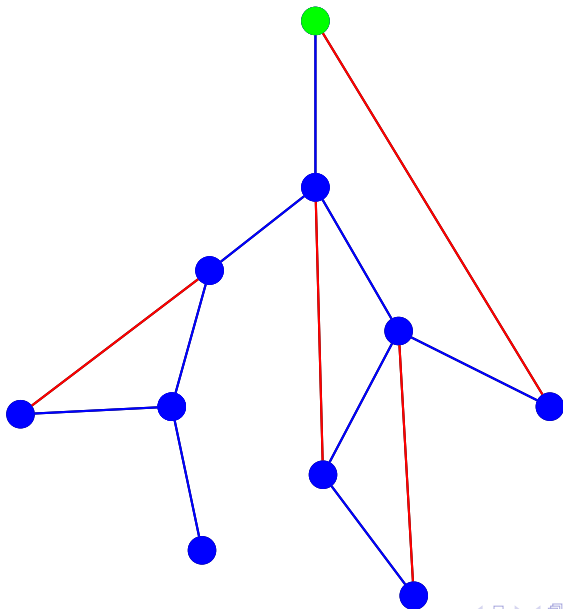




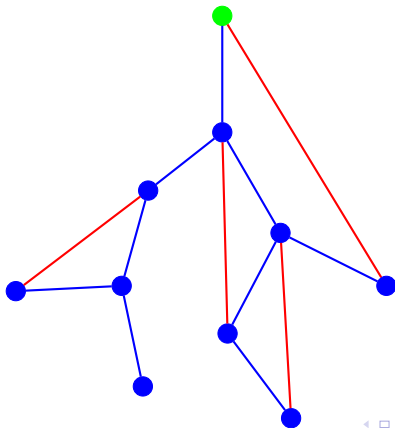








- Syvyyshaku löytää jonkun virittävän puun verkosta
- Tällä virittävällä puulla on erityisominaisuus: Kaaret jotka eivät kuulu tähän virittävään puuhun kulkevat solmusta johonkin sen esivanhempaan. "Poikittaisia" kaaria ei ole.



- Verkon sillat ovat täsmälleen ne virittävän puun kaaret joiden "yli" ei kulje yhtään kaarta

