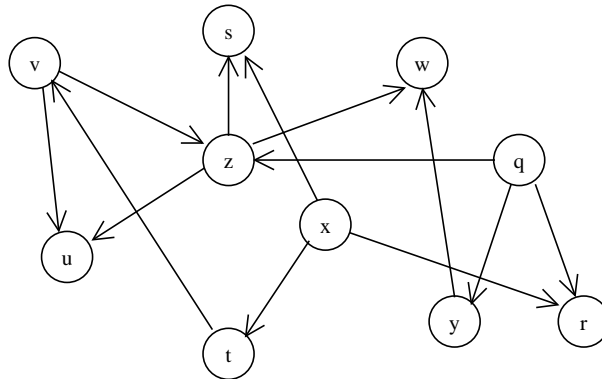
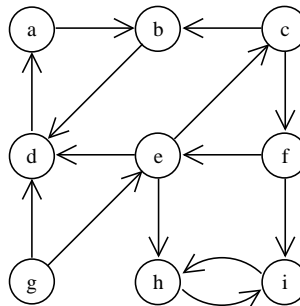


## Tietorakenteet, laskuharjoitus 11, 11-14.4.

1. Monisteessa leveysuuntainen läpikäynti on toteutettu iteratiivisesti (ilman rekursiota), kun taas syvyysuuntainen läpikäynti on toteutettu rekursiivisesti. Esitä ideatasolla, miten leveysuuntaisen läpikäynnin voi toteuttaa rekursiivisesti ja syvyysuuntaisen läpikäynnin iteratiivisesti.
2. Järjestä kuvan verkko topologisesti käyttäen luennolla esitettyä menetelmää. Oleta, että solmut ovat vieruslistoissa aakkosjärjestyksessä.



3. Muodosta allaolevan verkon vahvasti yhtenäiset komponentit luennolla esitettyllä algoritmilla. Oleta, että kaaret on talletettu vieruslistoihin aakkosjärjestyksessä.  
Mikä on pienin määrä lisättäviä kaaria, jolla alkuperäisestä verkosta saataisiin vahvasti yhtenäinen?



4. Jos verkko tallennetaan vieruslistoina, se vie vähän tilaa muistissa, mutta on hidasta tarkistaa, kuuluuko tietty kaari verkkoon. Jos verkko tallennetaan vierusmatriisina, se vie paljon tilaa muistissa, mutta on nopeaa tarkistaa, kuuluuko tietty kaari verkkoon. Suunnittele kolmas tapa tallentaa verkko, joka sekä vie vähän muistia että mahdollistaa nopean kaaren tarkistuksen. Tavoite: tilankäyttö  $O(|E| + |V|)$ , kaaren tarkistus ajassa  $O(\log |V|)$ . Onko ratkaisussasi mitään huonoa?
5. Tarkastellaan suuntaamatonta yhtenäistä verkkoa, jonka solmut kuvaavat tietokoneita ja kaaret niiden välisiä tietoliikenneyhteyksiä. Verkko on *haavoittuva*, jos siinä on solmu, jonka poistaminen verkosta rikkoo sen yhtenäisyyden. Toisin sanoen verkko on haavoittuva, jos yhden tietokoneen poistaminen verkosta voi estää kahden muun tietokoneen viestinnän.  
Esimerkiksi vasemmanpuoleinen on haavoittuva sillä solmun *b* poisto rikkoo yhtenäisyyden. Oikeanpuoleinen verkko taas ei ole haavoittuva.



Esitä algoritmi, joka tarkistaa, onko verkko haavoittuva. Verkko on tallennettu muistiin vieruslistaesityksenä. Määritä lisäksi algoritmisi aikavaativuus.

6. Syötteenä on annettu suunnattu painottamaton verkko  $G = (V, E)$ . Lisäksi sen solmuista on nimetty erityinen lähtösolmu  $s$  ja maalisolmu  $t$ , ja jokaiseen kaareen on liitetty väri *punainen* tai *sininen*. Ongelmana on löytää solmusta  $s$  solmuun  $t$  sellainen polku, että polulla kaikki punaiset kaaret ovat ennen sinisiä; ts. polku koostuu punaisesta alkuosasta ja sinisestä loppuosasta. Jos tällaisia polkuja on useita, niistä pitää valita (kaarten lukumäärällä mitattuna) mahdollisimman lyhyt.

Kehitä ongelmalle tehokas ratkaisualgoritmi ja analysoi sen aikavaativuutta.

Hitaampikin algoritmi kelpaa rastiin, mutta ongelmaan löytyy myös ajassa  $O(|V| + |E|)$  toimiva ratkaisu.

**7. Bonustehtävä jolla voi korvata 2 muuta tehtävää joko näistä tai aiemmista laskareista**

Ohjelma saa syötteenä joukon riippuvuuksia. Syötteenä voisi olla esim. kurssien suoritusjärjestysten keskinäiset riippuvuudet:

```

ohpe ohja
ohja tira
tira lama
tira tiralabra
ohja javalabra
javalabra tiralabra
ohpe ohma
tiralabra ohtuprojekti
javalabra ohtuprojekti
ohma ohtu
ohtu ohtuprojekti
lama ohtuprojekti

```

Eli jokaisella rivillä on ilmaistu yksi riippuvuus, esim. ohja tira määrittelee, että ohja täytyy suorittaa ennen tira:a.

Ohjelma tarkistaa ensin, etteivät riippuvuudet muodosta sykliä. Esim. jos edelliseen lisättäisiin

```
ohtuprojekti tira
```

Niin muodostuisi riippuvuussykli:

```
ohtuprojekti tira tiralabra ohtuprojekti
```

Riippuvuussykliä ei ole pakko tulostaa. Riittää, että ohjelma ilmoittaa että sykli löytyi. Tyylikkääntä on tietysti tulostaa myös sykli.

Jos riippuvuussykliä ei ole, tulostaa ohjelma *yhden* mahdollisen peräkkäisen järjestyksen, joka ei riko yhtään riippuvuutta.

Tee toteutus haluamallasi ohjelmointikielellä.