

Voit saada kustakin harjoituskerrasta max. 6 pistettä.

### Tehtävä 1. Tekoölytutkimus vs. sci-fi

Selaile netistä löytyviä tekoölyaiheisia artikkeleja viime vuosilta. Löydät niitä mm. konferenssijulkaisuista (IJCAI, AAAI, UAI) sekä lehdistä (IJAR, JAIR, AI).

- Minkälaisen vaikutelman saat nykyisestä tekoölytutkimuksesta?
- Minkätyyppisiä kysymyksiä käsitellään?
- Sopivatko aiheet tämän kurssin aihepiireihin?
- Minkälaisia opintoja tarvittaisiin, jotta selailemiasi artikkeleja voisi lukea (ja kirjoittaa)?

Kuinka realistiselta vaikuttaa ajatus, että artikkeleissa esitetyt tekoölymenetelmät tulisivat itsestään tietoisiksi ja hyökkäisivät ihmiskunnan kimppuun, kuten Terminator-elokuvissa?

### Tehtävä 2. Turingin testi

Selitä miten *kuvavarmennus* (CAPTCHA) liittyy Turingin testiin.

Tutustu lyhyesti Amazonin Mechanical Turk -palvelun ideaan. Palvelua käytetään mm. kuvavarmennuksen kiertämiseen: blogiin tai muuhun palveluun roskapostia tuottavat ohjelmit lähettävät kuvia halpatyövoiman tulkittavaksi.

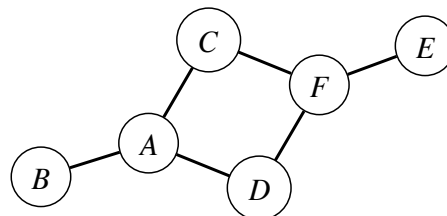
Kuka tässä tapauksessa on testaaaja ja kuka testattava Turingin testissä?

### Tehtävä 3. Etsintä: leveys- ja syvyysuuntainen haku

Ajatellaan oheista verkkoa.

Esitä verkon leveys- ja syvyysuuntainen läpikäynti alkaen solmusta *A*, kun maalisolmua ei ole (etsintä ei siis pääty, ennen kuin kaikki solmut on käsitelty).

Esitä luennon pseudokoodialgoritmin (tai ks. tehtävä 6 alla) *Solmulista*:n sisältö kussakin etsinnän vaiheessa, kummallakin etsintätavalla (leveys- ja syvyysuuntaisella).



#### Tehtävä 4. Etsintä ongelmanratkaisuna

Esitä kaaviona (verkkona) kannibaalit ja lähetyssaarnaajat -ongelman (ks. luentokalvot) sallitut tilat ja niiden väliset tilasiirtymät. Tilaesitys voi olla vaikkapa

(0,1,3,2,V)

mikä tarkoittaa, että joen vasemmalla rannalla on 0 lähetyssaarnaajaa ja 1 kannibaali, joen oikealla rannalla on 3 lähetyssaarnaajaa ja 2 kannibaalia.

Etsi kaavion avulla kelvollinen ratkaisu ja luettele sitä vastaavat siirrot, joilla kaikki matkustajat saadaan hengissä joen toiselle puolelle.

#### Tehtävä 5. Etsintä: perustietorakenteet

Toteuta haluamallasi ohjelmointikielellä (python, Java, ...) tietorakenne *Solmu*, johon voit tallettaa etsintäongelman tilan. Voit itse valita esitettävän tilan: esim. kaupungin nimi tai tehtävässä 4 veneen sijainti sekä kannibaalien ja lähetyssaarnaajien määrä kullakin rannalla.

Toteuta sen lisäksi *Pino* (last-in-first-out) ja *Jono* (first-in-first-out), johon voit tallettaa *Solmu*:ja ja josta voit poimia solmuja asiaankuuluvassa järjestyksessä.

Voit myös käyttää valmista koodia.

Testaa toteuttamiesi (tai valmiina löytämiesi) tietorakenteiden toimintaa tallettamalla pinoon ja jonoon joukko alkioita ja poimimalla ne sen jälkeen. Tarkista että alkiot poimitaan pinosta käänteisessä talletusjärjestyksessä ja jonosta talletusjärjestyksessä.

#### Tehtävä 6. Etsintä: hakualgoritmit (1–2 pistettä)

Toteuta leveys- ja syvyysuuntainen haku seuraavan pseudokoodin mukaisesti:

ETSINTÄ(Alkusolmu)

```
Solmulista = [Alkusolmu]
while Solmulista ei tyhjä
  Solmu = EKA(Solmulista)
  Solmulista = LOPUT(Solmulista)
  if MAALI(Solmu) return("ratkaisu", Solmu)
  Solmulista = LISÄÄ(NAAPURIT(Solmu), Solmulista)
return("ei ratkaisua")
```

Kun lisäät solmuja *Solmulista*:an, tarkista ettet lisää solmuja, jotka on jo kerran lisätty (vaikka ne olisikin jo poistettu).

Saat leveysuuntaiset haun, kun käytät *Solmulista*:na jonoa ja syvyysuuntaisen haun, kun käytät pinoja. Testaa algoritmeja tulostamalla verkon läpikäyntijärjestys tehtävän 3 verkolla. (Huomaa, että leveys- tai syvyysuuntainen läpikäynti ei ole yksikäsitteinen.)

Ylimääräinen piste: Kun lisäät uuden solmun *Solmulista*:an, talleta lisättyyn solmuun tieto nykyisestä solmusta, jotta voit ”peruuttaa” maalisolmusta takaisin alkusolmuun. Näin saat etsinnän päätyttyä tulostettua reitin alkusolmusta maalisolmuun. Testaa taas algoritmeja tehtävän 3 verkolla, kun maalisolmuna on *F*.

(Huom: Voit korvata ylimääräisellä pisteellä muita tehtäviä. Maksimipistemäärä per harjoituskerta on joka tapauksessa 6.)