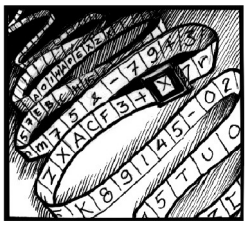

$P = NP$ -ongelma – mikä se on?

Tuomas Korppi

Johdanto

Tässä kirjoitelmassa esittelemme erään kuuluisimmista avoimista matemaattisista ongelmista. Kyse on $P = NP$ -kysymyksestä, joka kysyy, voidaanko tiettyjä tietokoneella suoritettavia laskentoja suorittaa nopeasti. Yhtälössä P tarkoittaa nk. polynomisessa ajassa laskettavia ongelmia ja NP nk. polynomisessa ajassa ei-deterministisesti laskettavia ongelmia. Nämä käsitteet määritellään myöhemmin tässä tekstissä. Yhtälö $P = NP$ siis väittää, että nämä kaksi ongelmaluokkaa ovat samat. Onko näin? Sitä kukaan ei tiedä.

Koska kyse on matemaattisesta ongelmasta, ratkaisuksi vaaditaan todistusta jommalle kummalle seuraavista:

- **Todistus sille, että kyseiset luokat ovat samat**. Tämä voitaisiin todistaa mm. tekemällä tiettyjä NP-luokkaan kuuluvia ongelmia nopeasti ratkova tietokoneohjelma.
- **Todistus sille, että luokat ovat eri.** Tämä tarkoittaisi todistusta sille, että on mahdotonta tehdä tiettyjä NP-luokkaan kuuluvia ongelmia nopeasti ratkovaa tietokoneohjelmaa.

Jos joku saa $P = NP$ -kysymyksen ratkaistua, Clay Mathematics Institute on luvannut ratkaisusta miljoonan dollarin palkinnon. Yllättävää kyllä, palkinto olisi periaatteessa mahdollista saada riittävän hyvällä Miinaharava-pelin analyysilla.

Laskennasta

Tässä kirjoitelmassa tarkoitamme tietokoneohjelmalla ohjelmaa, joka saa aluksi yhden syötteen, joka on äärellinen merkkijono[1], laskee sitä aikansa, ja lopuksi palauttaa joko merkkijonon ”Kyllä” tai merkkijonon ”Ei”. Tavallisessa tietokoneessa on rajallinen määrä muistia, ja tavallisissa yhteyksissä on rajallinen määrä aikaa laskennalle, mutta laskennan teoreettisessa tarkastelussa oletetaan, että nämä suureet ovat riittävän suuria käsillä olevan laskennan loppuunviemiseksi, olivatpa ne kuinka suuria tahansa, kunhan ne ovat äärellisiä. Samoin ohjelman saama syöte saa olla kuinka pitkä tahansa, kunhan se on äärellinen.

Polynominen aika

Olkoon T tietokoneohjelma edellisessä luvussa kuvaillussa mielessä. Merkitään $f(1)$:llä pisintä aikaa, jonka ohjelma käyttää yhden merkin mittaisen syötteen käsittelyyn. Yhden merkin mittaisia syötteitä on useita, ja merkitsemme siis $f(1)$:llä maksimia kaikkien yhden

[1] Yhteen merkkijonoon voidaan koodata vaikka millä mitalla tietoa, eli ohjelmamme voi saada syötteeksi esimerkiksi useita lukuja vaikkapa puolipisteellä erotettuna.

merkin mittaisten syötteiden vaatimista ajoista. Merkitään $f(2)$:lla pisintä aikaa, jonka ohjelma käyttää kahden merkin mittaisen syötteen käsittelyyn ja niin edelleen. Näin saamme funktion $f\colon \mathbb{N} \to \mathbb{N}$ (voidaan olettaa, että laskenta-ajat ovat kokonaislukuja, ja voidaan asettaa $f(0) = 0$).

Tyypillisesti $\lim_{n\to\infty} f(n) = \infty$, ja meitä kiinnostava kysymys on:

Kuinka nopeasti f lähenee ääretöntä?

Sanomme, että ohjelman T vaatima aika on *polynominen*, jos on olemassa (kokonaiskertoiminen) polynomi Q, jolle $f(n) \leq Q(n)$ kaikilla n. Tässä sillä, mikä on ykköstä edustava ajanjakso f:n määritelmässä ei ole merkitystä. Samat ohjelmat ovat polynomisia, valittiinpa tuo ajanjakso lyhyeksi tai pitkäksi.

Jos ohjelman T vaatima aika on polynominen, ohjelmaa T pidetään yleensä nopeana. Tämä johtuu siitä, että n:n kasvaessa minkä tahansa polynomin $Q(n)$ arvo kasvaa kohti ääretöntä suhteellisen hitaasti. Polynominen ohjelma onkin oikeasti nopea, jos polynomin Q aste on pieni. Jos polynomin Q aste on suuri, voi ohjelma olla käytännössä liian hidas, vaikka se olisikin polynominen.

Jos ohjelman T vaatima aika ei ole polynominen, se on niin hidas, että laskenta yhtään pidemmillä syötteillä on käytännössä toivotonta. Eksponenttifunktio $f(n) = 2^n$ kasvaa nopeammin kuin mikään polynomi. Voidaan siis todistaa, että jos Q on polynomi, on olemassa sellainen $n_0 \in \mathbb{N}$, että kaikilla $n > n_0$ pätee $f(n) > Q(n)$. Ei-polynomisten tietokoneohjelmien aikavaatimukset ovatkin melko usein joitakin eksponenttifunktion johdannaisia.

Esimerkki 1. *Olkoon T ohjelma, joka saa syötteenään listan lukuja sekä luvun k, ja joka tutkii käymällä listan läpi, onko luku k listassa. Tällaiselle ohjelmalle parhaan polynomin Q aste on yksi, eli T on polynominen ja nopea.*

Esimerkki 2. *Tässä esimerkissä puhutaan alkulukutesteistä, eli ohjelmista, jotka saavat syötteenään kymmenjärjestelmässä esitetyn luvun ja palauttavat tiedon siitä, onko kyseessä alkuluku. Huomautamme, että kun puhumme siitä, onko joku alkulukutesti polynominen, emme tarkoita, että onko ohjelman suoritusaika korkeintaan joku syötteenä saadun luvun polynomi, vaan sitä, onko ohjelman suoritusaika korkeintaan joku syötteenä saadun luvun kymmenjärjestelmäesityksen pituuden polynomi. Esimerkiksi luvun 1000000 kymmenjärjestelmäesityksessä on seitsemän merkkiä, mikä on huomattavasti vähemmän kuin miljoona.*

Jos käymme läpi kaikki kaikki luonnolliset luvut, jotka ovat korkeintaan syötteenä annetun luvun neliöjuuri, ja testaamme jokaisen kohdalla, onko kyseessä syötteen tekijä, saamme aikaan alkulukutestin, mutta sen vaatima laskenta-aika on niin pitkä, että testimme ei ole polynominen. Jos n on syötteenä saadun luvun kymmenjärjestelmäesityksen pituus, syötteenä saadun luvun neliöjuurta pienempiä lukuja on yli $10^{n/2-2} = \frac{1}{100}\sqrt{10}^n$, mikä on suurempi kuin 2^n, kun n on riittävän suuri.

Paras tunnettu varmasti toimiva alkulukutesti on polynominen, mutta paras sille tunnettu polynomi Q on astetta seitsemän. Näin suuri aste tarkoittaa sitä, että käytännössä tätä ohjelmaa ei käytetä alkulukujen testaamiseen, vaan alkulukutesteinä käytetään nopeampia ohjelmia, jotka toimivat vain tietyllä todennäköisyydellä, joka tosin saadaan huomattavan korkeaksi.

Päätösongelmat

Olkoon M (jossain äärellisessä aakkostossa esitettyjen) äärellisten merkkijonojen joukko, ja M', M'' joukon M jako kahteen osaan eli *päätösongelma*. Olkoon $m \in M$. Meitä kiinnostaa, kumpaan osaan, M' vai M'', m kuuluu. Olkoon T tietokoneohjelma, joka ratkaisee tämän, eli T on ohjelma, joka saa syötteenään m:n, ja T kertoo, kumpaan osaan, M' vai M'', syöte m kuuluu. Oletamme siis, että T toimii näin kaikkien M:n alkioiden kohdalla. Tällaisessa tapauksessa sanomme, että T *ratkaisee päätösongelman* M', M''.

Jos jollekin päätösongelmalle M', M'' on mahdollista kirjoittaa sen ratkaiseva tietokoneohjelma, joka toimii polynomisessa ajassa, sanomme, että M', M'' on polynominen. Polynomisten päätösongelmien luokkaa merkitään kirjaimella P.

Kauppamatkustajan ongelma

Tutkitaan seuraavaa päätösongelmaa: On annettu joukko kaupunkeja sekä hinnat kaikkien kahden kaupungin välisille matkoille. On lisäksi annettu maksimihinta h. Kauppamatkustajan ongelma kuuluu: Voidaanko tehdä kiertomatka, joka alkaa jostain kaupungista ja päättyy samaan kaupunkin niin, että jokaisessa kaupungissa käydään kerran ja kierroksen yhteishinta on korkeintaan h?

Jos haluamme saada tämän päätösongelman edellisessä kappaleessa esitettyyn formalismiin, M' siis kuvaa niitä systeemejä s (jossain merkkijonokoodauksessa esitettynä; tässä siis s sisältää tiedot kaupungeista, niiden välisten matkojen hinnoista sekä maksimihinnan h), joille kyseisenlainen kiertomatka on olemassa, ja M'' kuvaa muita systeemejä.

Kukaan ei tiedä, onko tämä päätösongelma polynominen, eli onko nopein mahdollinen tämän päätösongelman ratkaiseva tietokoneohjelma polynominen. Luki-

jalla kävi ehkä mielessä, että ongelma voitaisiin ratkaista käymällä kaikki mahdolliset reitit läpi ja katsomalla jokaisen reitin kohdalla, onko sen hinta korkeintaan h. Syötteen pituuden kasvaessa tällaisen laskennan viemä aika kasvaa kuitenkin nopeammin kuin mikään syötteen pituuden polynomi, eli tämä ei kelpaa polynomiseksi ratkaisuksi. Polynomisessa ajassa tämän ongelman ratkaisevan tietokoneohjelman pitäisi siis olla huomattavasti ovelammin laadittu.

Kuitenkin seuraavanlainen polynominen tietokoneohjelma T on olemassa: Olkoon s kuten edellä ja k kiertomatka systeemissä s. T saa syötteenään parin s, k ja ratkaisee, onko k sellainen kierros, että sen hinta on korkeintaan h.

Ei-deterministinen polynominen aika

Olkoon M', M'' päätösongelma. Oletetaan, että jokaiselle $m \in M'$ on olemassa toinen merkkijono m', jota kutsutaan m:n *todistajaksi*. Sallimme myös sen, että merkkijonolla voi olla useita todistajia. Lisäksi oletamme, että m:n lyhyimmän todistajan pituus on korkeintaan joku m:n pituuden polynomi. Lisäksi oletamme, että jos $m \in M''$, m:llä ei ole todistajia.

Esimerkiksi Kauppamatkustajan ongelma on tällainen päätösongelma: Systeemin s todistaja on kiertomatka k, jonka hinta on korkeintaan h.

Sanomme, että M', M'' on ei-deterministinen polynominen päätösongelma, jos on olemassa polynomisessa ajassa toimiva tietokoneohjelma T, joka saa syötteenään parin m, m' ja ratkaisee, onko m' jonon m todistaja. Ei-deterministisen polynomisten päätösongelmien luokkaa merkitään NP. Kuten edellisen luvun viimeisessä kappaleessa totesimme, Kauppamatkustajan ongelma on ei-deterministinen polynominen päätösongelma.

Huomautamme, että jos M', M'' on ei-deterministinen polynominen päätösongelma, on olemassa tietokoneohjelma T, joka ratkaisee tämän päätösongelman, joskin epärealistisen hitaasti. T muodostetaan seuraavasti: Olkoon Q polynomi siten, että jos m on pituutta n oleva syöte, jolla on todistaja, m:llä on korkeintaan pituutta $Q(n)$ oleva todistaja. Nyt kun tietokoneohjelmalle T annetaan syöte m, se käy kaikki korkeintaan pituutta $Q(n)$ olevat merkkijonot läpi ja kokeilee jokaisen kohdalla, onko kyseessä m:n todistaja. Syötteen pituuden kasvaessa tällaisen laskennan viemä aika kasvaa kuitenkin nopeammin kuin mikään syötteen pituuden polynomi, eli tämä ei kelpaa polynomiseksi ratkaisuksi.

Sivuhuomautuksena vielä mainittakoon, että nimitys ei-deterministinen polynominen tulee siitä, että tällaiset ongelmat voidaan ratkaista polynomisessa ajassa kuvitteellisilla tietokoneohjelmilla, jotka toimivat "ei-deterministisesti" eli osaavat arvata oikein. NP-päätösongelma voidaan ratkaista ei-deterministisesti niin, että ensin arvataan oikein todistaja ja sen jälkeen tarkastetaan polynomisessa ajassa, että arvattiin oikein.

Onko $P = NP$?

Nyt saamme muotoiltua $P = NP$ -kysymyksen. Se siis kysyy, onko luokka P sama kuin luokka NP. Jos T on jonkin päätösongelman M', M'' ratkaiseva polynominen tietokoneohjelma, voidaan ajatella, että jokaisen M':uun kuuluvan syötteen todistaja on tyhjä merkkijono, joten T toimii myös ei-deterministisessä polynomisessa ajassa. Siis P sisältyy luokkaan NP. Kysymys siis kuuluu: Voidaanko jokainen ei-deterministinen polynominen päätösongelma ratkaista polynomisessa ajassa, siis ohjelmalla, joka saa syötteekseen vain alkuperäisen syötteen eikä todistajakandidaattia? Kukaan ei tiedä. Yleisesti uskotaan, että nämä kaksi luokkaa ovat eri, mutta kukaan ei osaa todistaa, että kaikkia NP-ongelmia on mahdotonta ratkaista polynomisessa ajassa.

Sen verran kuitenkin tiedetään, että jos Kauppamatkustajan ongelma saataisiin ratkaistua polynomisessa ajassa, ratkaisusta osattaisiin muokata minkä tahansa NP-päätösongelman polynominen ratkaisu. Tällaisia NP-päätösongelmia, joiden polynominen ratkaisu ratkaisisi $P = NP$ -kysymyksen kertaheitolla on muitakin. Esimerkkinä tällaisesta mainittakoon seuraava:

Esimerkki 3. *Miinaharava-pelin tilanteella tarkoitamme mielivaltaisen kokoista Miinaharava-pelin tilannetta, jossa osa ruuduista on avattu ja osa avaamatta, ja kussakin avatussa ruudussa on numero, joka voi olla myös nolla. Lisäksi tilanteessa on asetettu lippuja osaan niistä ruuduista, joissa on miina. On myös mahdollista, että yhtään lippua ei ole asetettu. Oletamme kuitenkin, että liput on asetettu oikein, eli jokaisessa sellaisessa ruudussa, jossa on lippu, on myös miina.*

Nyt päätösongelmamme on seuraava: On annettu Miinaharava-pelin tilanne. Onko olemassa (muiden kuin liputettujen) miinojen sellaisia sijainteja, että ne sopivat annettuun tilanteeseen?

Lopuksi

NP ei suinkaan ole vaikeimpien mahdollisten päätösongelmien luokka. On olemassa päätösongelmia, jotka ovat ratkaistavissa tietokoneella[2], mutta jotka ovat niin vaikeita, että ne eivät kuulu edes luokkaan NP. Päätösongelmille on määritelty paljon muitakin luokkia kuin

[2] Joskin käytännön kannalta epärealistisen hitaasti.

P ja NP, ja $P = NP$ -kysymyksen lisäksi myös paljon muita luokkia koskevia kysymyksiä on vastaamatta. $P = NP$ -kysymys on pelkästään näistä kuuluisin.

On olemassa myös päätösongelmia, joita mikään tietokoneohjelma ei ratkaise. Tällaisiin tutustuimme kirjoitelmassa Korppi [1].

Pähkinöitä

1. Olkoon $f\colon \mathbb{N} \to \mathbb{N}$ funktio, jolle on olemassa polynomi Q ja luonnollinen luku n_0 siten, että $f(n) \leq Q(n)$ aina, kun $n \geq n_0$. Osoita, että on olemassa polynomi Q' siten, että $f(n) \leq Q'(n)$ kaikilla $n \in \mathbb{N}$. (Tämä tehtävä osoittaa, että tässä kirjoitelmassa annettu polynomisen aikavaatimuksen määritelmä on yhtäpitävä kirjallisuudesta yleisemmin löytyvän määritelmän kanssa.)
2. Tutkitaan Esimerkissä 3 esitettyä Miinaharavaa koskevaa päätösongelmaa. Osoita, että kyseinen ongelma kuuluu luokkaan NP.
3. Osoita, että jos Esimerkissä 3 esitetty Miinaharavaa koskeva päätösongelma saataisiin ratkaistua polynomisessa ajassa, saataisiin polynomisessa ajassa ratkaistua myös seuraava päätösongelma: On annettu Miinaharava-pelin tilanne ja suljettu ruutu r. Onko varmaa, että ruudussa r ei ole miinaa?
4. Onko $P = NP$?

Viitteet

[1] Korppi, Tuomas, *Mitä ei voida laskea?*, Solmu 1/2013

Verkko-Solmun oppimateriaalit

Osoitteesta `http://solmu.math.helsinki.fi/oppimateriaalit.html` löytyvät oppimateriaalit:

Kilpailumatematiikan opas (Matti Lehtinen)

Geometrian perusteita (Matti Lehtinen)

Geometria (K. Väisälä)

Lukualueiden laajentamisesta (Tuomas Korppi)

Jaksolliset desimaaliesitykset algebrallisesta näkökulmasta (Jaska Poranen ja Pentti Haukkanen)

Algebra (Tauno Metsänkylä ja Marjatta Näätänen)

Algebra (K. Väisälä)

Matemaattista fysiikkaa lukiolaiselle 1: Mekaniikkaa (Markku Halmetoja ja Jorma Merikoski)

Matemaattista fysiikkaa lukiolaiselle 2: Sähköoppia (Markku Halmetoja ja Jorma Merikoski)

Lukuteorian helmiä lukiolaisille (Jukka Pihko)

Matematiikan peruskäsitteiden historia (Erkki Luoma-aho)

Matematiikan historia (Matti Lehtinen)

Reaalianalyysiä englanniksi (William Trench)