# Alkulukutestit

Stefan Schmid

26.10.2018

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

| Tiedekunta/Osasto — Fakultet/Sektion — Faculty | Laitos — Institution — Department |
|---|---|
| Matemaattis-luonnontieteellinen | Matematiikan ja tilastotieteen laitos |

Tekijä — Författare — Author
Stefan Schmid

Työn nimi — Arbetets titel — Title

Alkulukutestit

Oppiaine — Läroämne — Subject
Matematiikka

| Työn laji — Arbetets art — Level | Aika — Datum — Month and year | Sivumäärä — Sidoantal — Number of pages |
|---|---|---|
| Pro gradu -tutkielma | Lokakuu 2018 | 41 s. |

Tiivistelmä — Referat — Abstract

Työssä käsitellään erilaisia menetelmiä, joita voidaan käyttää lukujen jaollisuuden tai jaottomuuden testaamiseen. Nämä menetelmät voidaan luokitella heuristisiin, probabilistisiin, ja deterministisiin testeihin. Esimerkkejä työssä käsitellyistä menetelmistä ovat muun muassa Fermat'n testi, PSW-testi, Millerin ja Rabinin testi ja AKS-testi. Työssä perehdytään eri menetelmien teoreettisiin taustoihin sekä annetaan esimerkkejä niiden soveltamisesta. Lisäksi työssä käsitellään tietokoneavusteista matematiikkaa. Tärkeimpien alkulukutestien yhteydessä on esitetty, miten testin suorittava tietokoneohjelma voidaan laatia C++ -ohjelmointikielellä.

Teoriapohjaksi ei vaadita yliopistotason matematiikan opintoja, mutta lukuteorian ja algebran opinnoista on hyötyä. Aiheen kannalta tärkeää teoreettista taustaa, esimerkiksi modulaarisen aritmetiikan perusteita, on käsitelty itse työssä.

Avainsanat — Nyckelord — Keywords
Alkuluku, alkulukutesti, lukuteoria

Säilytyspaikka — Förvaringsställe — Where deposited
Kumpulan tiedekirjasto

Muita tietoja — Övriga uppgifter — Additional information

# Sisältö

# Luku 1

# Johdanto

Tämä tutkielma käsittelee alkulukujen testaamista. Alkulukutesteillä viitataan tässä yhteydessä sellaisiin menetelmiin ja algoritmeihin, joiden avulla voidaan selvittää, onko annettu luku alkuluku. Alkuluvulla tarkoitetaan sellaista luonnollista lukua, jota ei voida jakaa tasan muilla luonnollisilla luvuilla kuin itsellään ja luvulla yksi.

Alkulukutestit voidaan jaotella deterministisiin, heuristisiin ja probabilistisiin testeihin. Deterministiset testit kertovat suoraan, onko annettu luku alkuluku vai ei. Ne ovat siten luotettavin tapa testata alkulukuja, mutta ne ovat yleensä laskennallisen kompleksisuudensa vuoksi myös hitaita. Heuristisilla testeillä puolestaan viitataan sellaisiin alkulukutesteihin, joiden tulosten luotettavuudesta ei voida olla täysin varmoja. Usein näiden testien taustalla on konejktuureja, joita ei ole vielä pystytty matemaattisesti todistamaan. Monet heuristiset testit ovat kuitenkin osoittautuneet hyödyllisiksi käytännön sovellusten kannalta. Probabilistiset testit ovat astetta täsmällisempi vaihtoehto heuristisille testeille. Myös probabilististen testien antamat tulokset ovat epävarmoja, mutta ne ovat matemaattisesti eksaktimpia kuin heuristiset testit, sillä testin tuloksen oikeellisuuden todennäköisyydelle voidaan yleensä johtaa jokin alaraja.

Alkulukutestien kehittymistä on edesauttanut niiden merkitys tieto- ja viestintätekniikassa, sillä alkuluvut ovat tärkeä osa salakirjoitusmenetelmien (kryptografian) toimintaa. Erityisesti suuret alkuluvut ovat oleellisia kryptografian kannalta. Kryptografian merkitys on korostunut sitä mukaan kun sähköinen asiointi on yleistynyt, mikä on osaltaan myös kiihdyttänyt alkulukujen tutkimusta. Vaikka lukuteoriaa on tutkittu matematiikan osa-alueena jo vuosisatoja, se ei ole aina nauttinut samanlaista arvostusta kuin tänä päivänä. Ennen tieto- ja viestintätekniikan kehittymistä 1900-luvulla lukuteoria oli monen matemaatikon mielessä harrastukseen verrattava ajanviete. Sille ei nähty löytyvän hyötykäyttöä samalla tavalla kuin esimerkiksi analyysilla ja todennäköisyyslaskennalla.

Suomenkielisessä matematiikassa ei näytä olevan vakiintunutta vastinetta englanninkieliselle termille *primality* eli ”alkulukuisuus”. Tässä tutkielmassa termin *primality* vastineena käytetään useimmiten sanaa jaollisuus.

Alkuluvut ovat tänä päivänä edelleenkin laajasti tutkittu matematiikan osa-alue, ja tässä tutkielmassa käsitellään myös joitakin vastikään havaittuja alkulukuihin ja niiden testaamiseen liittyviä seikkoja. Nämä uusimmat havainnot ovat toisinaan luonteeltaan sellaisia, ettei niiden pohjalta ole juuri laadittu varsinaisia tutkimusartikkeleita, vaan tuloksia on sen sijaan koottu erilaisille alkuluvuille omistetuille Internet-sivuille. Tämän vuoksi tässä tutkielmassa on hyödynnetty tavanomaisen tiedekirjallisuuden ohella myös verkkolähteitä ja Internet-tietosanakirjoja, joita ovat muun muassa *Encyclopedia of Mathematics*, *Wikipedia* ja *Rosetta Code*.

# Luku 2

# Alkulukujen määrittely ja yksinkertaiset menetelmät

Alkuluvuille voidaan esittää useita yhtäpitäviä määritelmiä. Tavanomaisin ja yksinkertainen määritelmä on seuraava.

**Määritelmä 2.1.** Luonnollinen luku $n > 1$ on alkuluku, jos sitä ei voida jakaa tasan muilla luonnollisilla luvuilla kuin 1 ja $n$.

Lukua 1 suurempia luonnollisia lukuja, jotka eivät ole alkulukuja, kutsutaan yhdistetyiksi luvuiksi. Yksinkertaisin tapa luvun $n$ jaollisuuden selvittämiseksi on yrittää jakaa se tasan kaikilla välin $[2, n-1]$ luonnollisilla luvuilla. Menetelmänä tämä on työläs ja tehoton, mutta lähestymistapaa voidaan parantaa merkittävästi muutaman yksinkertaisen havainnon avulla.

Oletetaan, että $m$ on yhdistetty luku. Tällöin on olemassa ykköstä suuremmat luonnolliset luvut $a$ ja $b$, joille on voimassa $m = ab$. Selvästi tekijöille $a$ ja $b$ pätee, että jos $a \geq \sqrt{m}$, niin silloin $b \leq \sqrt{m}$. Toisin sanoen jos luku ilmaistaan kahden tekijänsä tulona, on toisen tekijöistä oltava pienempi tai yhtäsuuri kuin luvun neliöjuuri. Jos siis halutaan selvittää jonkin luvun $n$ jaollisuus jakokokeilulla, niin jakoa ei tarvitse suorittaa kaikilla välin $[2, n-1]$, vaan riittää käydä läpi luonnolliset luvut väliltä $[2, \sqrt{n}]$.

Lisäksi huomataan, että yhdistetyt luvut voidaan ohittaa yrittäessä jakaa alkulukuehdokasta tasan. Jokainen yhdistetty luku voidaan kirjoittaa alkulukujen tulona, joten jos luku $n$ voidaan jakaa tasan yhdistetyllä luvulla $m$, voidaan se myös jakaa tasan luvun $m$ alkulukutekijöillä. Esimerkiksi luvun 101 jaollisuutta selvitettäessä ei ole tarpeen yrittää jakaa sitä tasan luvulla 9, sillä luvulla 9 jaollisen luvun olisi myös oltava jaollinen luvulla 3.

Näiden havaintojen pohjalta on mahdollista muodostaa alkulukujen joukko rekursiivisen menetelmän avulla.

**Lause 2.2.** Luku 2 on pienin alkuluku. Luonnollinen luku $n > 2$ on alkuluku, jos sitä ei voida jakaa tasan millään välin $[2, \sqrt{n}]$ alkuluvulla.

Tämä lause on periaatteessa käyttökelpoinen esimerkiksi ohjelmoinnissa silloin, kun halutaan muodostaa jono, joka sisältää $n$ ensimmäistä alkulukua (käytännössä Erastotheneen seula osoittautuu kuitenkin tehokkaammaksi). Se ei kuitenkaan sovellu hyvin tilanteisiin, joissa halutaan selvittää yksittäisen suuren luvun jaollisuus, sillä ensin olisi selvitettävä kaikki sen neliöjuurta pienemmät alkuluvut.

Jos tavoitteena on koota lista alkuluvuista, voidaan prosessia tehostaa karsimalla osa luonnollisista luvuista pois alkulukuehdokkaiden joukosta. Ensimmäinen lähes triviaali havainto on se, että kaikki lukua 2 suuremmat alkuluvut ovat parittomia, sillä parilliset luvut ovat jaollisia luvulla 2. Alkulukuehdokkaiden joukkoa voidaan rajata vielä pienemmäksi, kun tarkastellaan luonnollisten lukujen jaollisuutta luvulla $2 \cdot 3 = 6$. Jokaista luonnollista lukua $n$ kohti on olemassa kokonaisluvut $k$ ja $r$, missä $0 \leq r \leq 5$, joilla pätee $n = 6k + r$. Nähdään, että

- Muotoa $6k$, $6k + 2$ ja $6k + 4$ olevat luvut ovat jaollisia luvulla 2.
- Muotoa $6k + 3$ olevat luvut ovat jaollisia luvulla 3.

Tämä tarkoittaa sitä, että lukujen 2 ja 3 jälkeen kaikkien alkulukujen on oltava muotoa $6k+1$ tai $6k + 5$. Näin ollen kaksi kolmasosaa luonnollisista luvuista karsiutuu pois alkulukuehdokkaiden joukosta. Tätä lähestymistapaa voidaan jatkaa edelleen tarkastelemalla luonnollisten lukujen jaollisuutta luvulla $2 \cdot 3 \cdot 5 = 30$. Muotoa $30k + r$ olevat luonnolliset luvut ovat yhdistettyjä, jos $r$ on jaollinen luvulla 2, 3 tai 5. Näin ollen lukuja 2, 3 ja 5 suurempien alkulukujen on oltava muotoa $30k + r$, missä $r$ on luku 1, 7, 11, 13, 17, 19, 23 tai 29. Mahdollisten alkulukujen joukkoon jää siis enää 8/30 kaikista luonnollisista luvuista.

## 2.1 Erastotheneen seula

Erastotheneen seula tarjoaa yksinkertaisen menetelmän etsiä kaikki alkuluvut johonkin lukuun $n$ asti. Ensimmäinen vaihe Erastotheneen menetelmää hyödyntäessä on luetella kaikki luonnolliset luvut väliltä $[2, n]$. Tämän jälkeen aloitetaan yhdistettyjen lukujen karsiminen luettelosta, lähtien liikkeelle luvun 2 monikerroista. Kun kaikki luvun 2 monikerrat on karsittu, siirrymme seuraavaan jäljellä olevaan lukuun (tässä tapauksessa lukuun 3) ja karsimme sen monikerrat. Näin jatketaan, kunnes ollaan karsittu kaikkien välin $[2, \sqrt{n}]$ luonnollisten lukujen monikerrat. Lisäksi on hyvä huomata, että karsiessa jonkin luonnollisen luvun $m$ monikertoja, voidaan karsinta aloittaa luvusta $m^2$, sillä tätä pienemmät monikerrat on jo karsittu aiemmin. Jos esimerkiksi $m = 5$, niin luvut $2 \cdot 5$ ja $3 \cdot 5$ on jo karsittu lukujen 2 ja 3 monikertojen yhteydessä, joten riittää aloittaa luvusta $5^2$.

### 2.1.1 Erastotheneen menetelmään perustuva C++ -ohjelma

Alla on esitetty C++ -koodi ohjelmalle, joka pyytää käyttäjältä luvun $n$ ja etsii kaikki alkuluvut väliltä $[2, n]$ Erastotheneen seulan avulla. Koodiin on lisätty ohjelman toimintaa selostavia kommentteja, jotka voi erottaa vihreän värin perusteella.

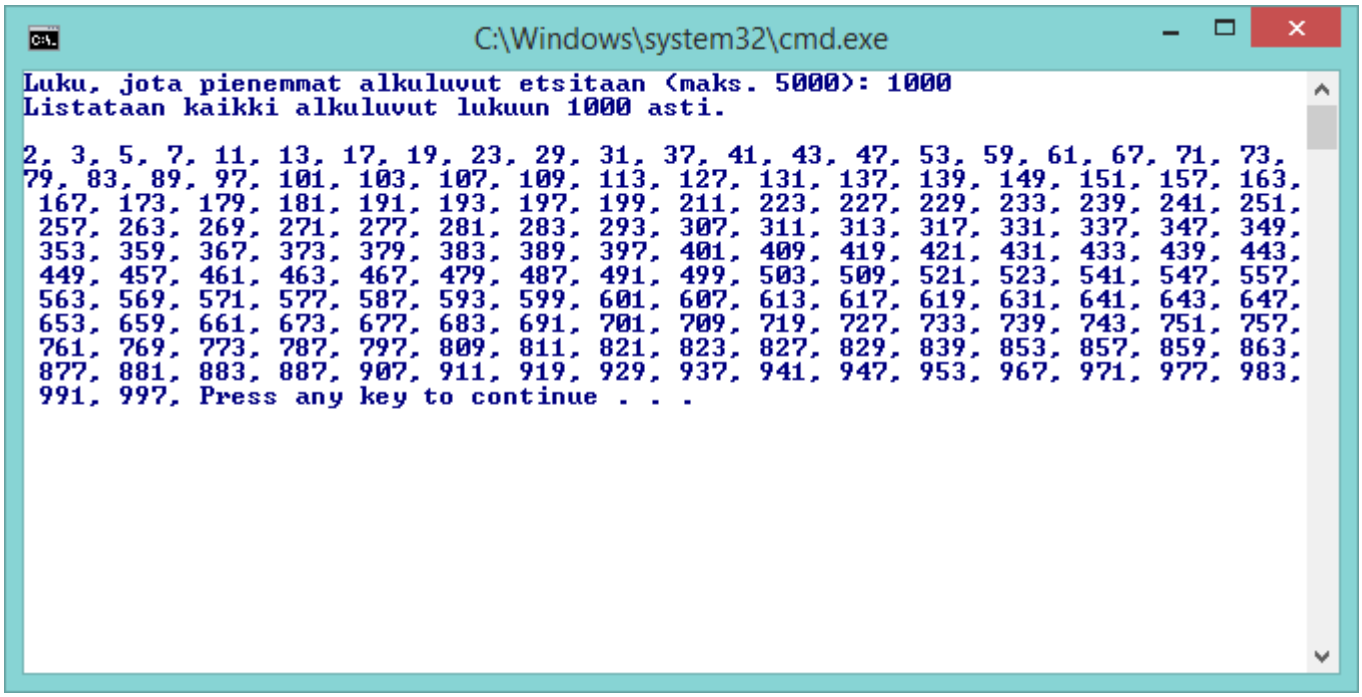
```
#include "stdafx.h"
#include <iostream>
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
   int n;
   cout << "Luku, jota pienemmat alkuluvut etsitaan (maks. 5000): ";
   cin >> n;
   cout << "Listataan kaikki alkuluvut lukuun " << n << " asti.\n\n";

   bool p[5001]; //jono, jonka jäsenet saavat arvon tosi tai epätosi
   std::fill_n(p, n, true); //asetetaan kaikille ensin arvoksi tosi
   for (int i = 2; i*i < n+1; i++)
   {
      if (p[i] == true) //p[i] on epätosi, jos i on yhdistetty luku.
      {               //Jos i on yhdistetty, myös sen monikerrat on jo käyty läpi.
         for (int k = 0; i*i + k*i < n+1; k++)
         {
            p[i*i + k*i] = false; //Merkitään luvun i monikerrat yhdistetyiksi,
         }                        //lähtien liikkeelle luvusta i^2
      }
   }

   for (int i = 2; i < n+1; i++) //Käydään lopuksi vielä läpi kaikki luvut i
   {
      if (p[i] == true) { cout << i << ", "; } //Tulostetaan alkuluvut ruudulle
   }
}
```

Ohjelma muodostaa jonon *bool* -tyyppisistä muuttujista, jotka voivat saada arvoksi tosi tai epätosi. Nämä muuttujat on indeksöity luvusta 0 eteenpäin. Jos algoritmin suoritettua muuttujan arvo on tosi, sen indeksiä vastaava luku on alkuluku. Yksinkertaisuuden vuoksi jonon pituudeksi on asetettu kiinteä luku, tässä tapauksessa 5001. Tästä syystä käyttäjän antama luku $n$ saa olla enintään 5000. Pienen lisävaivan avulla lukujonon pituus olisi mahdollista määritellä käyttäjän antaman luvun perusteella, mutta se lisäisi koodin monimutkaisuutta. Tässä

tutkielmassa esitettyjen esimerkkikoodien yhteydessä pyritään keskittymään alkulukutestien matematiikkaan, eikä niinkään ohjelmointiin liittyvien seikkoihin.




Kuva 2.1: Kuvakaappaus C++ -ohjelmasta, joka on laadittu edellä esitetyn koodin avulla. Ohjelma pyytää käyttäjältä luvun $n$ ja luettelee sen jälkeen kaikki alkuluvut väliltä $[2, n]$.

# Luku 3

# Peruslauseita

Tässä luvussa käsitellään kaksi tämän tutkielman kannalta keskeistä lukuteorian tulosta: Fermat'n pieni lause ja Lagrangen lause. Näiden avulla esitetään todistus Wilsonin lauselle, jota voidaan käyttää muun muassa alkulukutestinä. Aloitetaan Fermat'n pienestä lauseesta, joka on yksi tärkeimpiä lukuteorian tuloksia alkulukujen testaamisen kannalta.

## 3.1 Fermat'n pieni lause

**Lause 3.1.** Olkoon $p$ alkuluku ja $a$ kokonaisluku. Tällöin $a^p - a$ on luvun $p$ monikerta, eli

$$a^p \equiv a \pmod{p}.$$

Olettaen, että $p$ ei ole luvun $a$ tekijä, voidaan relaatio kirjoittaa yhtäpitävässä muodossa

$$a^{p-1} \equiv 1 \pmod{p}.$$

Jos $p$ on luvun $a$ tekijä, niin $p$ jakaa luvut $a^n$ kaikilla $n \in \mathbb{N}$, jolloin $a^{p-1} \equiv 0 \pmod{p}$.

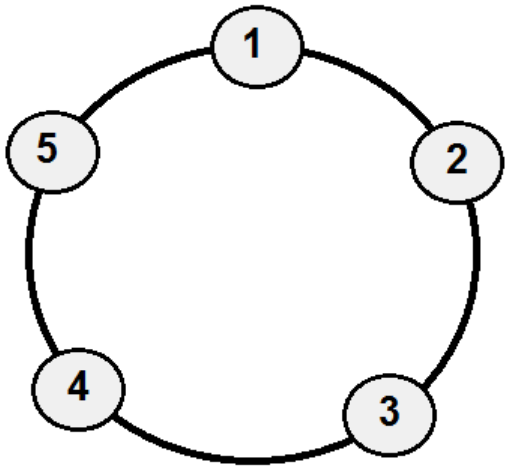
*Todistus.* Olkoon $a$ positiivinen kokonaisluku ja $p$ alkuluku. Määritellään joukko $K$, joka sisältää kaikki luonnolliset luvut väliltä $[1, a]$, eli $K = \{1, 2, ..., a\}$. Tutkitaan joukkoa $K^p$, jonka alkiot ovat $p$-ulotteisen koordinaatiston pisteitä $(n_1, n_2, ..., n_p)$. Joukon $K^p$ alkioiden lukumäärä on $a^p$. Joukossa $K^p$ on $a$ sellaista alkiota, joiden jokainen koordinaatti on sama. Muodostetaan joukko $X$, joka ei sisällä näitä muotoa $(n, n, ..., n)$ olevia alkioita, mutta sisältää kaikki muut joukon $K^p$ alkiot. Täsmällisemmin joukon $X$ määritelmä on siis seuraava:

$$X = \{k \in K^p \mid k \neq (n, n, ..., n) \text{ kaikilla } n \in [1, a]\}.$$

Muodostamamme joukon $X$ alkioiden lukumäärä on $a^p - a$. Seuraavaksi pyrimme osoittamaan, että joukon $X$ alkiot voidaan ryhmitellä siten, että jokaisessa ryhmässä on tasan $p$ alkiota.

Tämän perusteella $a^p - a$ olisi luvun $p$ monikerta, mikä todistaisi lauseen. Tätä ryhmittelyä varten otamme käyttöön käsitteen *helminauha*.

Tässä yhteydessä helminauha kuvaa sitä, millaisen ketjun joukon $X$ alkio muodostaa, jos sen ensimmäinen ja viimeinen alkio yhdistetään toisiinsa. Esimerkiksi alkiot $(1, 2, 3, 4, 5)$, $(2, 3, 4, 5, 1)$, $(3, 4, 5, 1, 2)$, $(4, 5, 1, 2, 3)$ ja $(5, 1, 2, 3, 4)$ muodostavat samanlaiset helminauhat. Samanlaisen helminauhan muodostavien alkioiden lukumäärä on yhtä suuri kuin helminauhassa toistuvan jakson pituus. Esimerkiksi alkion $(1, 1, 2, 1, 1, 2)$ helminauhassa toistuu jakso $(1 - 1 - 2)$, jonka pituus on kolme. Näin ollen saman helminauhan muodostaa vain kaksi muuta alkiota, $(1, 2, 1, 1, 2, 1)$ ja $(2, 1, 1, 2, 1, 1)$. Huomataan, että helminauhan pituuden on oltava siinä toistuvan jakson monikerta, eli jakson pituus jakaa helminauhan pituuden tasan. Helminauhan pituuden ollessa jokin alkuluku $p$, on jakson pituuden oltava joko $p$ tai 1.




Kuva 3.1: Havainnollistus helminauhasta matemaattisena käsitteenä. Helminauhassa ei ole ensimmäistä tai viimeistä koordinaattia, joten alkiot $(1, 2, 3, 4, 5)$, $(2, 3, 4, 5, 1)$, $(3, 4, 5, 1, 2)$, $(4, 5, 1, 2, 3)$ ja $(5, 1, 2, 3, 4)$ muodostavat samanlaiset helminauhat.

Palataan joukon $X$ tarkasteluun. Ryhmitellään joukon alkiot siten, että saman helminauhan muodostavat alkiot kuuluvat samaan ryhmään. Jokaisessa ryhmässä ryhmän alkioiden lukumäärä on sama kuin helminauhan jakson pituus. Koska joukon $X$ alkiot ovat $p$-ulotteisen koordinaatiston pisteitä, on niiden muodostamien helminauhojen pituus alkuluku $p$. Helminauhassa esiintyvän jakson pituus jakaa alkuluvun $p$ tasan, joten jakson pituuden on oltava 1 tai $p$. Jakson pituus 1 tarkoittaisi sitä, että vektorin jokainen koordinaatti olisi sama. Joukon $X$ määritelmässä kuitenkin hylättiin ne joukon $K^p$ alkiot, joiden jokainen koordinaatti on sama. Näin ollen joukon $X$ alkioiden muodostamien helminauhojen pituus on $p$. Aiemmin havaittiin, että samanlaisen helminauhan muodostavien alkioiden lukumäärä on sama kuin nauhan jakson pituus. Näin ollen jokaisen ryhmän alkioiden lukumäärä on $p$.

Olemme siis ryhmitelleet joukon $X$ alkiot siten, että jokainen ryhmä sisältää $p$ alkiota. Joukon $X$ alkioiden lukumäärä oli $a^p - a$, mikä on edellisen perustella luvun $p$ monikerta. Tämä todistaa väitteen $a^p \equiv a \pmod{p}$. □

*Huomautus* 3.2. Esitetyn todistuksen nojalla Fermat'n pieni lause on voimassa positiivisille kokonaisluvuille $a$, mikä on riittävä tulos tässä tutkielmassa esiintyvien sovellusten kannalta. Yleisesti Fermat'n pieni lause pätee kuitenkin kaikille kokonaisluvuille $a$, ja se on Eulerin lauseen erikoistapaus.

## 3.2 Lagrangen lause alkuluvuille

Seuraavaksi käsitellään Lagrangen lause alkuluvuille. Tämä lause osoittautuu hyödylliseksi, kun perustelemme Wilsonin testin oikeellisuuden.

**Lause 3.3.** Olkoon $p$ alkuluku ja $f(x) = a_n x^n + a_{n-1}x^{n-1} + ... + a_1 x + a_0$ kokonaislukukertoiminen polynomikuvaus asteluvulla $n$. Nyt kuvauksen $f$ termien kertoimet $a_0, a_1, ..., a_n$ ovat siis kokonaislukuja. Tällöin kuvaukselle $f$ pätee jompi kumpi seuraavista:

- Kaikki kuvauksen kertoimet ovat jaollisia luvulla $p$.
  Toisin sanoen: relaatio $a_k \equiv 0 \pmod{p}$ on voimassa kaikilla $k \in [0, n]$.
- Kuvausta $f$ kohti on enintään sen asteluvun $n$ verran sellaisia muuttujan $x$ arvoja, jotka eivät poikkea toisistaan luvun $p$ monikerran verran, ja joilla kuvaus $f$ saa luvulla $p$ jaollisen arvon.
  Toisin sanoen: Yhtälöllä $f(x) \equiv 0 \pmod{p}$ on enintään $n$ ratkaisua, joilla on eri jakojäännös luvulla $p$ jaettaessa.

*Todistus.* Oletetaan, että $p$ on alkuluku. Määritellään polynomikuvaus
$f(x) \in \mathbb{Z}[x], f(x) = a_n x^n + a_{n-1}x^{n-1} + ... + a_1 x + a_0$.
Määritellään kokonaisluvut $b_n, b_{n-1}, ..., b_0$ siten, että kaikilla $k \in [0, n]$ :
$b_k < p$ ja $b_k \equiv a_k \pmod{p}$. Määritellään sitten kuvaus
$g(x) \in (\mathbb{Z}/p)[x], g(x) = b_n x^n + b_{n-1}x^{n-1} + ... + b_1 x + b_0$.
Kuvaus $p$ on siis polynomifunktio, jonka termien kertoimet vastaavat kuvauksen $f$ termien kerrointen jakojäännöksiä luvulla $p$ jaettaessa.

Jos $g(x) = 0$, niin kuvauksen $g$ jokaisen termin kerroin on nolla. Näin ollen kuvauksen $f$ jokainen termi on jaollinen alkuluvulla $p$. Tämä oli ensimmäinen kuvaukselle $f$ tarjottu vaihtoehto Lagrangen lauseessa.

Oletetaan sitten, että $g(x)$ ei ole identtisesti nolla. Jos $a_n$ on jaollinen luvulla $p$, niin $b_n = 0$. Tällöin polynomin $g(x)$ asteluku on pienempi kuin polynomin $f$ asteluku. Muutoin polynomeilla on sama asteluku $n$. Nyt voimme tehdä seuraavat havainnot:

- Syklinen ryhmä $\mathbb{Z}/p$ on kunta. Jos siis polynomikuvauksen $g$ asteluku on enintään $n$, niin sillä on enintään $n$ nollakohtaa. Nollakohdilla on eri jakojäännökset luvulla $p$ jaettaessa.
- $f(m) \equiv 0 \pmod{p}$ jos ja vain jos $g(m) = 0$.

Näin ollen on olemassa enintään $n$ kongruenssirelaation $f(m) \equiv 0 \pmod{p}$ toteuttavaa muuttujan $m$ arvoa, joilla on eri jakojäännös luvulla $p$ jaettaessa. Tämä oli toinen kuvaukselle $f$ tarjottu vaihtoehto Lagrangen lauseessa.

□

## 3.3 Wilsonin testi

Wilsonin lause tarjoaa yksinkertaisen testin luvun jaollisuuden selvittämiselle. Lauseen mukaan kertoma $(n-1)!$ on yhden pienempi kuin jokin luvun $n$ monikerta, jos ja vain jos $n$ on alkuluku tai luku 1.

**Lause 3.4.** Luonnollinen luku $n \geq 2$ on alkuluku, jos ja vain jos

$$(n-1)! \equiv -1 \pmod{n}.$$

*Todistus.* On osoitettava, että alkuluvut toteuttavat Wilsonin lauseen relaation, ja että relaation toteuttava luku on alkuluku.

Osoitetaan ensin, että kongruenssirelaation toteuttava ykköstä suurempi luku on alkuluku. Olkoon $n \geq 2$ luonnollinen luku, jolle pätee $(n-1)! \equiv -1 \pmod{n}$.
Tehdään vastaoletus, että $n$ on yhdistetty luku. Tällöin $n \geq 4$, ja se voidaan jakaa tasan jollakin alkuluvulla $a$, missä $2 \geq a \geq n-2$. Oletuksen mukaan $n$ jakaa luvun $(n-1)!+1$. Koska $a$ on luvun $n$ tekijä, myös se jakaa luvun $(n-1)!+1$. Toisaalta, koska $a < n$, on luvun $a$ oltava jokin luvuista $n-2$, $n-3$ , ... , 2. Luku $a$ siis esiintyy tekijänä kertomassa $(n-1)!$, joten se jakaa luvun $(n-1)!$.

On siis osoitettu, että $a$ jakaa sekä luvun $(n-1)!+1$ että luvun $(n-1)!$, joten se jakaa luvun 1. Tämä on ristiriita, joten $n$ ei voi olla yhdistetty luku.

Nyt on vielä osoitettava, että kaikki alkuluvut toteuttavat kongruenssirelaation. Oletetaan siis, että $p$ on alkuluku. Todistuksen idea on seuraava: muodostamme polynomikuvauksen $h(x)$, jonka vakiotermi on $(p-1)!+1$, ja joka saa luvulla $p$ jaollisen arvon astelukuaan suuremmalla

määrällä muuttujan $x$ arvoja. Tällöin funktio $h$ ei toteuta Lagrangen lauseen toista ehtoa, joten sen on toteutettava lauseen ensimmäinen ehto, jonka mukaan sen termien kertoimet ovat jaollisia luvulla $p$. Tästä seuraa, että kuvauksen $h$ vakiotermi $(p-1)!+1$ on jaollinen luvulla $p$, mikä on yhtäpitävä muoto Wilsonin lauseelle.

Selvästi luku 2 toteuttaa relaation: $(2-1)! = 1 \equiv -1 \pmod 2$. Käsitellään sitten lukua 2 suuremaat, parittomat alkuluvut. Olkoon $p \geq 3$ alkuluku. Määritellään avuksi kuvaukset $f$ ja $g$ seuraavasti

$$f(x) := x^{p-1} - 1,$$
$$g(x) := (x-1)(x-2)...(x-(p-1)).$$

Molemmilla kuvauksilla on sama asteluku $p-1$ ja sama ensimmäinen termi $x^{p-1}$. Fermat'n pienen lauseen mukaan $a^{p-1} - 1 \equiv 0 \pmod p$, jos $a$ ei ole jaollinen luvulla $p$. Erityisesti jos $a < p$, niin $a$ ei voi olla jaollinen luvulla $p$, joten Fermat'n lause on voimassa. Fermat'n lauseen nojalla kongruenssirelaatio $f(x) = x^{p-1} - 1 \equiv 0 \pmod p$ on voimassa muuttujan $x$ arvoilla $1, 2, ..., p-1$.

Polynomilla $g$ on $p-1$ nollakohtaa: $g(x) = 0$ muuttujan $x$ arvoilla $1, 2, 3, ..., p-1$. Näin ollen myös myös kuvaukselle $g$ pätee, että kongruenssirelaatio $g(x) \equiv 0 \pmod p$ on voimassa muuttujan $x$ arvoilla $1, 2, ..., p-1$. Muodostetaan vielä kuvaus $h(x) = g(x) - f(x)$. Nyt kuvaukselle $h$ pätevät seuraavat havainnot:

- Kuvausten $f$ ja $g$ vakiotermit ovat $-1$ ja $(p-1)!$, joten kuvauksen $h = g - f$ vakiotermi on $(p-1)!+1$.
- Relaatiot $f(x) \equiv 0 \pmod p$ ja $g(x) \equiv 0 \pmod p$ ovat voimassa muuttujan $x$ arvoilla $1, 2, ..., p-1$. Näin ollen relaatio $h(x) \equiv 0 \pmod p$ on voimassa muuttujan $x$ arvoilla $1, 2, ..., p-1$. Kuvauksella $h$ on siis vähintään $p-1$ epäkongruenttia nollakohtaa modulo $p$.
- Kuitenkin kuvauksen $h$ asteluku on korkeintaan $p-2$, sillä kuvauksissa $f$ ja $g$ esiintyvä ensimmäinen termi $x^{p-1}$ kumoutuu erotuksessa $g-f$. Näin ollen Lagrangen lauseen nojalla kuvauksella $h$ on korkeintaan $p-2$ epäkongruenttia nollakohtaa modulo $p$.

Kuvauksen $g$ on siten oltava identtisesti nolla modulo $p$, eli $f(x) \equiv 0 \pmod p$, joten myös kuvauksen $h$ vakiotermin on oltava nolla modulo $p$, eli $(p-1)!+1 \equiv 0 \pmod p$.
Tämä todistaa alkuperäisen väitteen $(p-1)! \equiv -1 \pmod p$. □

Wilsonin lause on yksinkertainen esimerkki hitaasta deterministisestä testistä. Käytännössä se ei ole kovinkaan hyödyllinen alkulukujen testaamisen näkökulmasta. Tämä johtuu siitä, että

suuren alkulukuehdokkaan $n$ tapauksessa kertoman $(n-1)!$ laskeminen vaatii paljon laskutehoa. Jopa jakokokeilu osoittautuu usein tehokkaammaksi menetelmäksi. Esimerkiksi luvun 401 jaollisuuden selvittäminen Wilsonin lauseen avulla vaatisi laskun 400! modulo 401 suorittamista. Jakokokeilussa sen sijaan riittää tarkistaa, onko luku 401 jaollinen luvulla $2, 3, 5, 7, 11, 13, 17$ tai 19. Wilsonin lauseen kaltaiset alkulukuja koskevat lukuteorian tulokset ovat kuitenkin olleet tärkeitä tehokkaampien menetelmien kehittymisen kannalta.

# Luku 4

# Heuristiset testit

Heuristiset testit poikkeavat aiemmin käsitellyistä menetelmistä siten, että niiden avulla ei voida saavuttaa täydellistä varmuutta tarkasteltavan luvun jaollisuudesta. Heuristisilla testeillä on silti etunsa. Ne ovat usein tehokkaita, mikä mahdollistaa suurten lukujen jaollisuuden tutkimisen verrattain vähäisellä laskuteholla.

## 4.1 Fermat'n testi

Edellisessä luvussa käsiteltiin Fermat'n pieni lause, jonka mukaan alkuluvulle $p$ ja kokonaisluvulle $a$ pätee $a^{p-1} \equiv 1 \pmod{p}$, olettaen että $a$ ei ole luvun $p$ monikerta. Useimmiten tämä kongruenssiyhtälö ei ole voimassa, jos $p$ on yhdistetty luku. Tämän perusteella Fermat'n pientä lausetta voidaan käyttää heuristisena testinä luvun $p$ jaollisuuden selvittämiseksi seuraavasti:

- Valitaan jokin kokonaisluku $a > 1$, joka ei ole luvun $p$ monikerta
- Selvitetään luvun $a^{p-1}$ jakojäännös, kun jakajana on luku $p$.
- Jos jakojäännös on 1, luku $p$ on luultavasti alkuluku. Muilla jakojäännöksillä $p$ on yhdistetty luku.

Käytännössä luku $a$ kannattaa valita väliltä $[2,\ p-1]$, koska kaikilla $n \in \mathbb{N}$ muotoa $np+1$ olevat luvun $a$ arvot toteuttavat kongruenssiyhtälön triviaalisti: jos $a \equiv 1 \pmod{p}$, niin $a^{p-1} \equiv 1 \pmod{p}$. Seuraavassa esimerkissä lukujen 211 ja 221 jaollisuutta on tutkittu Fermat'n testin avulla.

**Esimerkki 4.1.** Tutkitaan luvun $p = 211$ jaollisuutta. Valitaan ensin $a = 17$.
Huomataan, että $a^{p-1} = 17^{210} \equiv 1 \pmod{211}$.
Toistetaan testi kantaluvun $a$ arvolla 95.

Nyt $a^{p-1} = 95^{210} \equiv 1 \pmod{211}$.
Kokeillaan vielä kantaluvun $a$ arvoa 250.
Nyt $a^{p-1} = 250^{210} \equiv 1 \pmod{211}$.
Jakojäännös näyttäisi olevan aina 1, joten Fermat'n testin perusteella luku 211 on todennäköisesti alkuluku.

Tutkitaan sitten lukua $p = 221$. Valitaan ensin $a = 38$.
Tällöin $a^{p-1} = 38^{220} \equiv 1 \pmod{221}$.
Kokeillaan sitten kantaluvun $a$ arvolla 78.
Tällöin $a^{p-1} = 78^{220} \equiv 13 \pmod{221}$.

Toisella kerralla jakojäännökseksi saatiin 13, joten Fermat'n pieni lause ei ole voimassa. Luvun 221 on siten oltava yhdistetty luku. Tämä esimerkki osoittaa myös sen, että Fermat'n testin tulos ei aina ole luotettava, sillä ensimmäisellä yrityksellä luku 221 läpäisi testin kantaluvulla 38.

Fermat'n testi luokitellaan välillä myös probabilistiseksi testiksi. Todennäköisyys, että Fermat'n testi ilmoittaa yhdistetyn luvun olevan luultavasti alkuluku, riippuu valitusta kantaluvusta $a$ ja siitä, miltä väliltä alkulukukandidaatti on valittu. Esimerkiksi kantaluvulla $a = 2$ testin läpäisee hieman alle 22000 yhdistettyä lukua väliltä $[1,\ 2,5 \times 10^{10}]$. Lukua $2,5 \times 10^{10}$ pienempien alkulukujen lukumäärä on noin $1,1 \times 10^9$. Näin ollen kantaluvulla $a = 2$ testin läpäisevien yhdistettyjen lukujen suhde alkulukuihin on noin $0,005\,\%$ välillä $[1,\ 2,5 \times 10^{10}]$.

Eräs Fermat'n testin merkittävä heikkous on se, että on olemassa ääretön määrä yhdistettyjä lukuja, jotka läpäisevät testin kaikilla kantaluvuilla $a$. Näistä niin sanotuista Carmichaelin luvuista pienin on 561.

### 4.1.1 Modulaariaritmetiikkaa

Monet tässä tutkielmassa käsiteltävät alkulukutestit ovat käytännöllisiä vain, jos niitä sovellettaessa hyödynnetään myös sopivia modulaarisen aritmetiikan tuloksia. Tämän voi havaita jo Fermat'n testin yhteydessä esitellystä esimerkistä, jossa luvun 221 jaollisuuden selvittämiseksi oli tiedettävä jakojäännös, kun luku $78^{220}$ jaetaan luvulla 221. Laskun $78^{220}$ suorittaminen osoittautuu ongelmalliseksi, sillä useimmat laskimet ja ohjelmointimenetelmät eivät mahdollista näin suurten lukujen käsittelyä. Tämänkaltaisissa tilanteissa suurten lukujen selvittäminen voidaan kuitenkin ohittaa hyödyntämällä seuraavaa tulosta:

**Lemma 4.2.** Olkoot $a, b, k$ ja $n \neq 0$ luonnollisia lukuja.
Jos $a \equiv b \pmod{n}$, niin $ka \equiv kb \pmod{n}$.

*Todistus.* Oletetaan, että $a \equiv b \pmod{n}$. Tällöin on olemassa kokonaisluku $c$, jolle $b = a + cn$.
Näin ollen $kb = ka + kcn$, ja selvästi $ka \equiv ka + kcn \pmod{n}$. □

On siis sallittua kertoa kongruenssirelaatio puolitain luonnollisella luvulla. Tätä tulosta voidaan hyödyntää jakojäännösten selvittämiseksi, kun jaettavana on suuri potenssimuodossa esitetty luku.

**Esimerkki 4.3.** Selvitetään jakojäännös, kun luku $8^4$ jaetaan luvulla 5. Hyödynnetään lemmaa 4.2 seuraavasti:

$8 \equiv 3 \pmod 5$ || kerrotaan relaatio puolittain luvulla 8.
$8^2 \equiv 8 \cdot 3 \equiv 24 \equiv 4 \pmod 5$
$8^3 \equiv 32 \equiv 2 \pmod 5$
$8^4 \equiv 16 \equiv 1 \pmod 5$.

Kun luku $8^4$ jaetaan luvulla 5, on jakojäännös siis 1.

Tätä menetelmää hyödyntämällä voidaan välttyä niiltä haasteilta, jotka liittyvät suurten lukujen käsittelyyn. Lähestymistapaa voidaan lisäksi tehostaa vielä siten, että myös suoritettavien laskujen määrä on pienempi.

**Esimerkki 4.4.** Selvitetään jakojäännös, kun luku $13^{37}$ jaetaan luvulla 11. Toistuvasti neliöimällä ja hyödyntämällä aiempien laskujen tuloksia voidaan välttyä tarpeelta suorittaa 36 kertolaskua:

$13 \equiv 2 \pmod{11}$
$13^2 \equiv 4 \pmod{11}$
$13^4 = 13^2 \cdot 13^2 \equiv 16 \equiv 5 \pmod{11}$
$13^8 \equiv 25 \equiv 4 \pmod{11}$
$13^{16} \equiv 16 \equiv 5 \pmod{11}$
$13^{32} \equiv 25 \equiv 4 \pmod{11}$
$13^{36} = 13^{32} \cdot 13^4 \equiv 4 \cdot 5 \equiv 9 \pmod{11}$
$13^{37} = 13^{36} \cdot 13 \equiv 9 \cdot 2 \equiv 7 \pmod{11}$.

Kun luku $13^{37}$ jaetaan luvulla 11, on jakojäännös siis 7.

Alla on esitetty vielä toinen modulaarisen aritmetiikan perustulos, koskien jäännösluokkia ja summia.

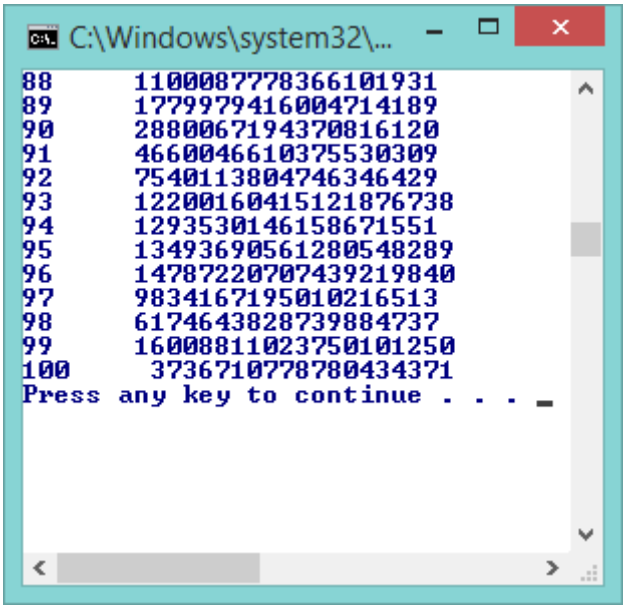
**Lemma 4.5.** Olkoot $a, b$ ja $n \neq 0$ luonnollisia lukuja.
Jos $a \equiv c \pmod n$ ja $b \equiv d \pmod n$, niin
$a + b \equiv c + d \pmod n$.

*Todistus.* Oletetaan, että $a \equiv c \pmod{n}$ ja $b \equiv d \pmod{n}$.
Tällöin on olemassa kokonaisluvut $k_1$ ja $k_2$, joille $a = c + k_1 \cdot n$ ja $b = d + k_2 \cdot n$.
Näin ollen $a + b = c + d + n(k_1 + k_2) \equiv c + d \pmod{n}$. □

Lemma 4.4 osoittautuu hyödylliseksi erityisesti Fibonaccin lukujen jakojäännösten tarkastelun yhteydessä. Fibonaccin jonon ensimmäinen ja toinen jäsen ovat luku 1. Muut jäsenet saadan kahden edeltävän jäsenen summana.

**Esimerkki 4.6.** Selvitetään Fibonaccin jonon 8. jäsenen jakojäännös modulo 5.
Jonon ensimmäiset kahdeksan jäsentä ovat $1, 1, 2, 3, 5, 8, 13$ ja $21$.
Koska kahdeksas jäsen on tunnettu, voidaan suoraan laskea $21 \equiv 1 \pmod{5}$.
Vaihtoehtoisesti voidaan hyödyntää lemmaa 4.4. Muodostetaan Fibonaccin jono modulo 5, jossa siis jonon jäsen on kahden edellisen luvun summan jakojäännös modulo 5. Tällöin jonon kahdeksan ensimmäistä jäsentä ovat $1, 1, 2, 3, 0, 3, 3$ ja $1$.

Molemmat menetelmät johtavat luonnollisesti samaan tulokseen: kahdeksannen jäsenen jakojäännös luvulla 5 jaettaessa on 1. Jälkimmäinen menetelmä osoittautuu hyödylliseksi suurten Fibonaccin lukujen jakojäännösten tarkastelussa, sillä sen avulla voidaan sivuuttaa suurten lukujen käsittelyyn littyvät haasteet.




Kuva 4.1: Kuvakaappaus Fibonaccin lukuja listaavasta ohjelmasta. Ohjelmoinnissa on käytetty *unsigned int64* -datatyyppiä, eli 64-bittisiä ei-negatiivisa kokonaislukuja. Jonon 94. jäsen ylittää arvon $2^{64} - 1$, joten ohjelman ilmoittamat luvut muuttuvat virheellisiksi.

### 4.1.2 Fermat'n testin suorittava ohjelma

Fermat'n testin suorittava tietokoneohjelma on helposti toteutettavissa testin yksinkertaisuuden vuoksi. Alla esitetty koodi on testin suorittavan C++ -ohjelmointikielellä laaditun sovelluksen koodi. Koodin logiikassa olennaista on edellä esitetyn lemman 4.2 hyödyntäminen toistuvasti *while* –silmukan sisällä.

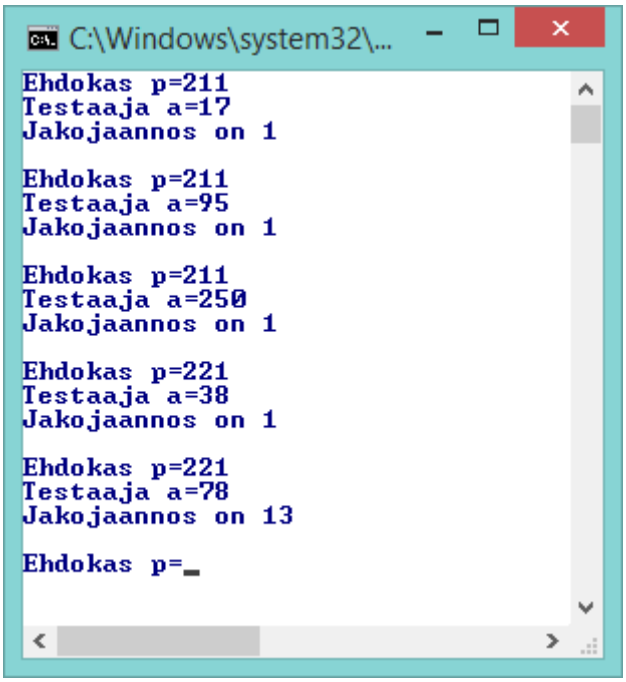
```
#include "stdafx.h"
#include "iostream"
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
   int p=0; //alkulukuehdokas
   int a=0; //fermat testaaja
   int e=0; //juokseva eksponentti
   int r=0; //jakojäännös

      cout << "Ehdokas p=";
      cin >> p;
      cout << "Testaaja a=";
      cin >> a;
      e=1;
      r=1;

      while (e < p)
      {
         r = r*a%p;
         e++;
      }
      cout << "Jakojaannos on " << r << "\n\n";
}
```

Koodi pyytää käyttäjää antamaan alkulukuehdokkaan $p$ ja kantaluvuksi testaajan $a$. Jakojäännöksen ilmoittava muuttuja $r$ asetetaan aluksi arvoon 1. Tämän jälkeen toistetaan $p-1$ kertaa prosessi, jossa muuttujan $r$ uudeksi arvoksi asetetaan jakojäännös, joka on saatu jakamalla luku $r \cdot a$ luvulla $p$. Lemman 4.2 nojalla muuttujan $r$ lopullinen arvo on tällöin sama kuin luvun $a^{p-1}$ jakojäännös modulo $p$.

Tätä esimerkkiä varten koodi on pyritty esittämään mahdollisimman yksinkertaisessa ja helposti luettavassa muodossa. Ohjelman toimintaa on mahdollista huomattavasti tehostaa hyödyntämällä esimerkissä 4.4 esitettyä toistetun neliöinnin menetelmää, mikä vähentäisi suoritettavien kertolaskujen määrää erityisesti suurten lukujen $p$ kohdalla.




Kuva 4.2: Kuvakaappaus Fermat'n testin suorittavasta C++ -ohjelmasta. Kuvassa ohjelmalla on selvitetty esimerkkiä 4.1 varten tarvitut jakojäännökset.

## 4.2 Fibonaccin testi

Fibonaccin lukuja voidaan käyttää apuna alkulukutesteissä. Erityisesti seuraavat Fibonaccin lukuja koskevat tulokset osoittautuvat hyödyllisiksi.

**Lause 4.7.** Alkuluvulla $p$ on seuraavat ominaisuudet.

- Luku $p$ jakaa Fibonaccin luvun $F_{p-1}$ tasan, jos $p \equiv \pm 1 \pmod 5$.
- Luku $p$ jakaa Fibonaccin luvun $F_{p+1}$ tasan, jos $p \equiv \pm 2 \pmod 5$.

**Seuraus 4.8.** Luonnollinen luku $p$ on yhdistetty luku, jos se toteuttaa yhden seuraavista ehdoista.

- $F_{p-1} \not\equiv 0 \pmod p$ ja $p \equiv \pm 1 \pmod 5$.
- $F_{p+1} \not\equiv 0 \pmod p$ ja $p \equiv \pm 2 \pmod 5$.

Kuten Fermat'n testin tapauksessa, myös yhdistetty luku voi toteuttaa lauseen 4.7 ominaisuudet. Useimmiten näin ei ole, eli ominaisuudet toteuttava luku on luultavasti alkuluku, mutta yleisesti Fibonaccin testin läpäiseminen ei yksinään anna riittävää varmuutta luvun jaottomuudesta. Useamman heuristisen testin yhdistelmät saattavat kuitenkin olla hyvin tehokkaita, esimerkkinä tästä on seuraavassa alaluvussa käsitelty PSW-testi. Seuraavassa esimerkissä on tutkittu kahta luonnollista lukua Fibonaccin testin avulla.

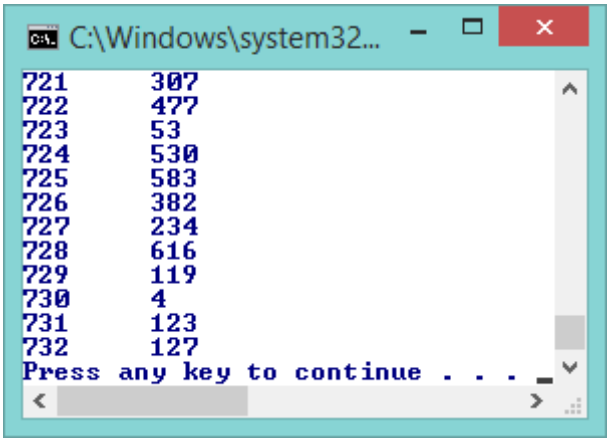
**Esimerkki 4.9.** Tutkitaan lukujen 143 ja 731 jaollisuutta.

Selvästi $143 \equiv -2 \pmod 5$, joten selvitetään $F_{144}$.

$F_{144} = 555565404224292694404015791808$, ja $\dfrac{F_{144}}{143} = 3885072756813235625202907635,021...$

Jako $\dfrac{F_{144}}{143}$ ei mene tasan, joten 143 on yhdistetty luku.

Tutkitaan sitten lukua 731. Fibonaccin luvun $F_{730}$ selvittäminen olisi haastavaa, joten hyödynnetään edellisen luvun esimerkkiä 4.5 ja muodostetaan Fibonaccin lukujono modulo 731, jossa siis seuraava jonon jäsen saadaan jakojäännöksenä, kun kahden edellisen luvun summa on jaettu luvulla 731. Saadaan siis jono
$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 256, 135, 391, ...$




Kuva 4.3: Kuvakaappaus C++ -ohjelmasta, jolla muodostettiin Fibonaccin jono modulo 731. Kuvasta nähdään, että jonon 730. jäsen on luku 4.

Jonon 730. jäsen on luku $4 \neq 0$, joten jako $\dfrac{F_{730}}{731}$ ei mene tasan. Täten 731 on yhdistetty luku.

## 4.3 PSW-testi

Yhdysvaltalaiset matemaatikot Carl Pomerance, John Selfridge ja Samuel Wagstaff ovat esittäneet konjektuurin, jota hyödynnetään yleisesti alkulukujen testaamisessa. Oletetaan, että $p$ on pariton luku, jolle pätee $p \equiv \pm 2 \pmod 5$. PSW-konjektuurin mukaan $p$ on alkuluku, jos se toteuttaa seuraavat ehdot:

- $2^{p-1} \equiv 1 \pmod p$ ja
- $f_{p+1} \equiv 0 \pmod p$.

Tässä $f_n$ on $n$:s Fibonaccin luku. Konjektuurin ensimmäinen ehto vaatii Fermat'n testin suorittamisen kantaluvulla 2. Toinen ehto perustuu lauseeseen 4.7, jonka mukaan alkuluku $p$ jakaa luvun $f_{p+1}$, jos $p \equiv \pm 2 \pmod 5$. Käytännössä PSW-testi on siis Fermat'n testin ja Fibonaccin testin yhdistelmä. Seuraavassa esimerkissä on tutkittu luvun 52633 jaollisuutta PSW-testin avulla.

**Esimerkki 4.10.** Tutkitaan lukua 52633.
Fermat'n testin suorittavan ohjelman avulla nähdään, että $2^{52632} \equiv 1 \pmod{52633}$.
Fibonaccin testillä puolestaan selviää, että $f_{52634} \equiv 15285 \pmod{52633}$.
Luku 52633 siis läpäisee Fermat'n testin kantaluvulla 2, mutta ei Fibonaccin testiä. Näin ollen se on yhdistetty luku. Itse asiassa kyseessä on Carmichaelin luku, eli se läpäisee Fermat'n testin testaajaksi valitun kantaluvun $a$ arvosta riippumatta. Oikeaa tietoa luvun 52633 jaollisuudesta ei siten olisi saatu pelkän Fermat'n testin avulla.

Lukua 5 suuremmat alkuluvut päättyvät numeroon 1, 3, 7 tai 9. PSW-testi soveltuu ainoastaan numeroihin 3 tai 7 päättyvien lukujen testaamiseen ehdon $p \equiv \pm 2 \pmod 5$ vuoksi. Mikäli käsiteltävänä on jokin lukujoukko $A$, jossa numeroihin 1, 3, 5 ja 7 päättyviä lukuja esiintyy yhtä usein, voidaan PSW-testillä siis testata vain puolet mahdollisista alkuluvuista. Käytännön sovelluksissa tämä ei välttämättä ole ongelma, mikäli kaikkia joukon $A$ alkulukuja ei tarvita. Yleisesti esimerkiksi kryptografiassa tavoitteena on vain löytää mahdollisimman tehokkaasti *joitakin* suuria alkulukuja, joiden varaan salaustekniikka voidaan rakentaa. Kaikkien tietyn välin alkulukujen selvittäminen on harvoin tavoitteena, jolloin tarve karsia alkulukuehdokkaita testin rajoitteiden vuoksi ei välttämättä ole ongelma.

PSW-konjektuuria ei ole vielä pystytty todistamaan. Toisaalta yhtäkään testin läpäisevää yhdistettyä lukua ei myöskään ole löydetty. Pomerance, Selfridge ja Wagstaff ovat tarjonneet yhteensä 620 dollarin rahapalkkion vastaesimerkistä.

# Luku 5

# Probabilistiset testit

Edellisessä luvussa käsitellyt heuristiset testit eivät anna täyttä varmuutta tarkasteltavan luvun jaollisuudesta. Fermat'n testin ja Fibonaccin testin yhteydessä havaittiin, että testin läpäisevä luku saattaa olla yhdistetty luku. PSW-testin tapauksessa testin läpäisevää yhdistettyä lukua ei ole löydetty, mutta toisaalta sen oikeellisuutta ei niin ikään voida matemaattisesti todistaa. Probabilistiset testit eivät myöskään anna täydellistä varmuutta luvun jaollisuudesta. Nämä testit ovat kuitenkin siinä mielessä täsmällisempiä, että testin tuloksen oikeudellisuuden todennäköisyys, tai vähintäänkin todennäköisyyden alaraja, on yleensä täsmällisemmin määriteltävissä.

Probabilistinen testi on lähinnä hyödyllinen siinä tapauksessa, jos testi voidaan suorittaa useamman kerran putkeen käyttämällä eri testilukuja ja jos peräkkäisten testien voidaan osoittaa olevan toisistaan rippumattomia tapahtumia todennäköisyyden suhteen. Usein probabilistisen testin toiminta nojaa seuraavaan ideaan:

- Valitaan testaajaksi jokin kokonaisluku $a$.
- Selvitetään, toteuttavatko alkulukuehdokas $p$ ja testaaja $a$ jonkin valitusta testistä riippuvan ehdon.
- Jos ehto ei toteudu, $p$ on yhdistetty luku, ja testi voidaan lopettaa. Jos ehto toteutuu, luvun $p$ todennäköisyys olla yhdistetty luku on korkeintaan $P$.
- Toistetaan testi varioimalla testaajaa $a$ kunnes saavutetaan haluttu varmuus. Jos luku $p$ läpäisee testin $n$ kertaa, on sen todennäköisyys olla yhdistetty luku korkeintaan $P^n$.

Esimerkkejä tämän kaltaisista testeistä ovat muun muassa Millerin ja Rabinin testi sekä Solovayn ja Strassenin testi.

## 5.1 Millerin ja Rabinin testi

Millerin ja Rabinin alkulukutesti perustuu alun perin Gary L. Millerin laatimaan deterministiseen testiin. Alkuperäinen testi oletti todistamattoman yleistetyn Riemannin hypoteesin pitävän paikkansa, joten sen oikeellisuudesta ei ole täyttä varmuutta. Michael O. Rabin laati Millerin testin pohjalta probabilistisen testin, jonka oikeellisuus ei ole riippuvainen todistamattomista tuloksista. Millerin ja Rabinin testi toimii seuraavan algoritmin mukaisesti.

- Olkoon $p > 2$ testattava alkulukuehdokas. Valitaan testaajaksi luonnollinen luku $a < p$.
- Etsitään pariton luonnollinen luku $d$, jolle $2^s d = p - 1$, missä $s \in \mathbb{N}$.
  (Huomaa, että luku $p - 1$ on parillinen ehdon $p > 2$ vuoksi. Lähtemällä liikkeelle luvusta $p - 1$ ja jakamalla toistuvasti luvulla 2 päädytään väistämättä parittomaan lukuun, joka on haluttuu luku $d$. Toisin sanoen, $d$ on luvusta 2 poikkeavien luvun $p - 1$ alkulukutekijöiden tulo. Luku $s$ puolestaan kertoo, kuinka monta kertaa luku 2 esiintyy luvun $p - 1$ alkulukuhajotelmassa.)
- Jos $a^d \not\equiv 1 \pmod{p}$ ja $a^{2^r d} \not\equiv -1 \pmod{p}$ kaikilla kokonaisluvuilla $r \in [0, s - 1]$, niin $p$ on yhdistetty luku. Muutoin $n$ saattaa olla alkuluku.
- Mikäli alkulukuehdokas $p$ läpäisi testin, voidaan tuloksen luotettavuutta parantaa toistamalla testi useilla testaajan $a$ arvoilla. Usein testaajan $a$ arvot pyritään valitsemaan satunnaisesti.

Millerin ja Rabinin testistä voidaan joissakin tapauksissa suorittaa deterministinen versio toistamalla testi tietyillä testaajan $a$ arvoilla. Se, mitkä testaajan $a$ arvot on käytävä läpi, riippuu alkulukuehdokkaan $p$ suuruudesta. Pomerance, Selfridge, Wagstaff ja Jaeschke ovat todentaneet, että jos $p < 4\,759\,123\,141$, niin riittää käydä läpi testaajan $a$ arvot $2, 7$ ja $61$. Tämä versio testistä on riittävä 32-bittiseille kokonaisluvuille, sillä $2^{32} < 4\,759\,123\,141$. Vastaavasti jos $p < 2^{64}$, niin riittää käydä läpi testaajan $a$ arvot $2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31$ ja $37$. Seuraavassa esimerkissä Millerin ja Rabinin testin avulla on selvitetty, onko 3277 alkuluku.

**Esimerkki 5.1.** Tutkitaan lukua $p = 3277$. Alkuluvut eivät toteuta Millerin ja Rabinin testin ehtoja. Jos 3277 täyttää testin ehdot, on luvun oltava yhdistetty luku.
Haluamme kirjoittaa luvun $p - 1$ muodossa $2^s d$, missä $d$ on pariton:
$p - 1 = 3276 = 2 \cdot 1638 = 2 \cdot 2 \cdot 819$. Siis $d = 819$ ja $s = 2$.
Valitaan testaajaksi $a = 2$.
Tarkistetaan ensin, täyttyykö ehto $a^d \not\equiv 1 \pmod{p}$:
$a^d = 2^{819} \equiv 128 \neq 1 \pmod{3277}$, joten ensimmäinen ehto täyttyy.

Tarkistetaan sitten, täyttyykö ehto $a^{2^r d} \not\equiv -1 \pmod{p}$ kaikilla $r \in [0, s-1]$.
On siis käytävä läpi luvun $r$ arvot 0 ja 1.
Tapaus $r = 0$ tarkistettiin käytännössä jo ensimmäisen ehdon kohdalla:
$a^{2^r d} = 2^{2^0 819} = 2^{819} \equiv 128 \neq -1 \pmod{3277}$.
Käydään vielä läpi tapaus $r = 1$. Tällöin
$a^{2^r d} = 2^{2^1 \cdot 819} = 2^{1638} \equiv 3276 \equiv -1 \pmod{3277}$.

Luku $p = 3277$ ei täytä Millerin ja Rabinin testin ehtoja, kun testaaajana käytettiin lukua $a = 2$. Näin ollen $p$ saattaa olla alkuluku. Toistetaan testi vielä testaajalla $a = 3$.

Tarkistetaan ensimmäinen ehto. Kuten aiemmin havaittiin, samassa yhteydessä selviää myös toinen ehto arvolla $r = 0$, koska $2^0 = 1$:
$a^d = a^{2^r d} = 3^{819} \equiv 2564 \neq \pm 1 \pmod{3277}$.
Toinen ehto luvun $r$ arvolla 1:
$a^{2^r d} = 3^{1638} \equiv 434 \neq -1 \pmod{3277}$.

Käyttämällä testaajan $a$ arvoa 3 Millerin ja Rabinin testin ehdot ovat siis voimassa luvulle 3277. Näin ollen 3277 on yhdistetty luku.

### 5.1.1 Millerin ja Rabinin testin suorittava ohjelma

Alla on esitetty koodi Millerin ja Rabinin testin suorittavalle C++ -ohjelmalle.

---

```
#include "stdafx.h"
#include "iostream"
#include <iomanip>
#include <string>
using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{

   int p=0, a=0, s=0, d=0, r=1; // Millerin ja Rabinin testin parametrit
   int q=1, e=0;                // Avustavia muuttujia laskua (a^d mod p) varten
   bool yhd = true;             // Kertoo, onko kyseessä yhdistetty luku

   cout << "Ehdokas p=";
   cin >> p;
   cout << "Testaaja a=";
   cin >> a;
   d = p-1;                 // Asetetaan d ensin arvoon p-1.

   while (d%2==0)           // Selvitetään, kuinka monta kertaa luku 2
   {                        // esiintyy luvun (p-1) alkulukuhajotelmassa.
      d = d/2;
      s++;                  // Saadaan parametrien s ja d oikeat arvot.
   }

   while (e < d)            // Muuttuja e on juokseva eksponentti.
   {
      q = q*a % p;          // Silmukan loputtua q = a^d (mod p).
      e++;
   }


   if (q==1 || q==p-1)      // "Jos q=1 tai jos q=-1"
   {                        // Testataan ehto 1, ja ehto 2 arvolla r=0.
      yhd = false;
   }
```

```
   while (yhd == true && r < s) // Ehto 2 parametrin r muilla arvoilla.
   {                            // Testaaminen voidaan lopettaa, jos ehto ei täyty.
      r++;                      // Kun siirrytään seuraavaan parametrin r arvoon,
      q = q*q % p;              // lauseke a^((2^r)d) neliöityy.

      if (q == p-1)             // Jos q = p-1 (mod p), niin ehto 2 ei toteudu.
      {                         // Tällöin testaus voidaan lopettaa, ja
         yhd = false;           // p saattaa olla alkuluku.
      }
   }

   if (yhd == true) { cout << p << " on yhdistetty luku.\n\n"; }
   else { cout << p << " saattaa olla alkuluku.\n\n"; }



   return 0;
}
```

### 5.1.2 Millerin ja Rabinin testauksen taustaa

Kuten aiemmin käsitellyn Fermat'n testin tapauksessa, myös Millerin ja Rabinin testin lähtökohtana toimii Fermat'n pieni lause. Tarvitsemme myös seuraavaa lemmaa.

**Lemma 5.2.** Olkoon $p$ alkuluku.
Jos $x^2 \equiv 1 \pmod{p}$, niin $x \equiv 1 \pmod{p}$ tai $x \equiv -1 \pmod{p}$.

*Todistus.* Oletetaan, että $p$ on alkuluku ja $x^2 \equiv 1 \pmod{p}$. Tällöin
$x^2 - 1 \equiv 0 \pmod{p}$, ja edelleen
$(x+1)(x-1) \equiv 0 \pmod{p}$. Toisin sanoen $p$ jakaa lukujen $(x+1)$ ja $(x-1)$ tulon. Koska $p$ on alkuluku, se jakaa näin ollen joko luvun $x-1$ tai luvun $x+1$, mikä on yhtäpitävä alkuperäisen lemman kanssa.

□

Palataan Fermat'n pieneen lauseeseen. Lauseen mukaan alkuluvulle $p$ on voimassa

$$a^{p-1} \equiv 1 \pmod{p}$$

kaikilla kokonaisluvuilla $a$. Oletetaan, että $p > 2$ ja siten pariton. Tällöin luku $p-1$ on parillinen, ja se voidaan kirjoittaa muodossa $2^s d$, missä $d$ on pariton. Tässä luku $s$ kertoo, kuinka monta

kertaa luku 2 esiintyy luvun $p-1$ alkulukuhajotelmassa, ja $d$ on luvun $p-1$ parittomien alkulukutekijöiden tulo. Nyt Fermat'n pieni lause voidaan kirjoittaa muodossa

$$a^{2^s d} \equiv 1 \pmod{p}.$$

Tästä seuraa lemman 5.2 nojalla, että

$$a^{2^{s-1} d} \equiv 1 \pmod{p} \qquad \text{tai} \qquad a^{2^{s-1} d} \equiv -1 \pmod{p}.$$

Jos yllä esitetyistä relaatioista ensimmäinen on voimassa, seuraa lemman 5.2 nojalla

$$a^{2^{s-2} d} \equiv 1 \pmod{p} \qquad \text{tai} \qquad a^{2^{s-2} d} \equiv -1 \pmod{p}.$$

Näin voidaan jatkaa kunnes päädytään lukuun $a^d$. Fermat'n pienei lause ja lemma 5.2 johtavat siis seuraavaan tulokseen.

**Lause 5.3.** Olkoon $p > 2$ alkuluku, ja olkoot $d$ ja $s$ luonnollisia lukuja, joille $p-1 = 2^s d$, missä $d$ on pariton. Tällöin
$a^d \equiv 1 \pmod{p}$ tai
$a^{2^r d} \equiv -1 \pmod{p}$ jollakin kokonaisluvulla $r \in [0, s-1]$.

Millerin ja Rabinin testissä siis pyritään tunnistamaan yhdistetty luku testaamalla lauseen 5.3 kontrapositio.

### 5.1.3 Probabilistiset testit ja todennäköisyys

Kuten aiemmin mainittiin, Millerin ja Rabinin testin avulla on mahdollista tunnistaa kaikki yhdistetyt luvut, sillä jokaista yhdistettyä lukua kohti vähintään 3/4 mahdollisista testaajan $a$ arvoista osoittaa, että kyseessä on yhdistetty luku. Jos testi suoritetaan $n$ kertaa valitsemalla jokaista testiä varten uusi satunnainen testaajan $a$ arvo, yhdistetyn luvun todennäköisyys läpäistä testaus on enintään $\frac{1}{4^n}$.

On syytä huomata, että tämä todennäköisyys ei yksinään vielä kerro paljoa. Oleellista on tietää myös, millä tiheydellä alkulukuja esiintyy testattavan alkulukuehdokkaan ympäristössä. Tätä on havainnollistettu seuraavassa esimerkissä.

**Esimerkki 5.4.** Oletetaan, että joukon $A \subset \mathbb{N}$ luvuista $1/5$ on alkulukuja. Oletetaan yksinkertaisuuden vuoksi myös, että jokainen joukon $A$ yhdistetty luku havaitaan tasan $3/4$ testaajan $a$ arvolla. Valitaan alkulukuehdokkaaksi $p$ jokin satunnainen joukon $A$ alkio ja suoritetaan Millerin ja Rabinin testi yhden kerran satunnaisella testaajan $a$ arvolla. Tällöin:

- Todennäköisyys, että joukosta $A$ valittiin alkuluku on $1/5$.
- Todennäköisyys, että valittiin yhdistetty luku ja sellainen testaajan arvo, joka ei sitä havaitse, on $4/5 \cdot 1/4 = 1/5$.
- Todennäköisyys, että valittiin yhdistetty luku ja sellainen testaajan arvo, jolla tämä käy ilmi, on $3/5$.

Oletetaan sitten, että luku $p$ läpäisee testin. Yllä esitetyistä kolmesta vaihtoehdosta viimeinen ei siis toteutunut, mutta kahden ensimmäisen tapahtuman todennäköisyydet ovat edelleen keskenään yhtä suuria. Näin ollen luvun $p$ todennäköisyys olla yhdistetty luku on $1/2$.

Tämä esimerkki kuvaa hyvin sitä, miksi käytännön sovellusten kannalta emme useinkaan ole kiinnostuneita siitä, millä todennäköisyydellä yhdistetty luku läpäisee testin. Tärkeämpi kysymys on seuraava: millä todennäköisyydellä testin läpäissyt luku on yhdistetty luku? Yleisesti tätä varten voidaan hyödyntää Bayesin lakia ehdollisille todennäköisyyksille. Jos tapahtuman $A$ todennäköisyys on $P(A)$, ja tapahtuman $B$ todennäköisyys on $P(B) > 0$, niin

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}.$$

Tässä $P(A|B)$ on tapahtuman $A$ todennäköisyys sillä ehdolla, että tapahtuma $B$ on käynyt toteen ja vastaavasti $P(B|A)$ on tapahtuman $B$ todennäköisyys ehdolla $A$. Edellisen esimerkin tapauksessa Bayesin lakia olisi voitu soveltaa seuraavasti:

Olkoon $P(A) = 4/5$ on todennäköisyys, jolla ehdokas $p$ on yhdistetty luku.
Olkoon $P(B) = 2/5$ on todennäköisyys, jolla ehdokas $p$ läpäisee testin.
Todennäköisyys, jolla $p$ läpäisee testin jos se on yhdistetty luku, on siten $P(B|A) = 1/4$.
Näin ollen testin läpäisseen ehdokkaan $p$ todennäköisyys olla yhdistetty luku on

$$P(A|B) = \frac{1/4 \cdot 4/5}{2/5} = \frac{1/5}{2/5} = 1/2.$$

# Luku 6

# Erityistä muotoa olevien lukujen testaus

Aiemmissa luvuissa käsiteltyihin alkulukutesteihin ei juuri liity rajoitteita sen suhteen, mitä alkulukuehdokkaita niiden avulla voidaan testata. Poikkeuksena tästä oli PSW-testi, joka soveltuu vain numeroihin 3 ja 7 päättyvien lukujen testaukseen. Näiden lisäksi on myös laadittu sellaisia testejä, jotka soveltuvat vain tiettyä erityistä muotoa olevien lukujen testaukseen. Näistä testeistä tässä luvussa käsitellään Lucasin ja Lehmerin testi sekä Pépinin testi.

## 6.1 Lucasin ja Lehmerin testi Mersennen luvuille

Lucasin ja Lehmerin testi on Mersennen lukujen testaamista varten laadittu alkulukutesti. Testi on deterministinen, eli se antaa varman tiedon siitä, onko testattava luku alkuluku vai yhdistetty luku. Se on erityisen tehokas suurten alkulukujen etsinnässä. Lucasin ja Lehmerin testiä hyödyntävän *Great Internet Mersenne Prime Search* –projektin avulla on löydetty enemmistö suurimmista tunnetuista alkuluvuista. Joulukuussa 2017 GIMPS -projektin avulla luku $2^{77\,232\,917} - 1$ osoitettiin alkuluvuksi, ja se on tällä hetkellä suurin tunnettu alkuluku.

### 6.1.1 Mersennen luvuista

Mersennen luvuilla tarkoitetaan tässä yhteydessä kaikkia luonnollisia lukuja, jotka ovat muotoa $2^n - 1$ jollakin $n \in \mathbb{N}$. (Joissakin yhteyksissä Mersennen luvuilla viitataan ainoastaan sellaisiin muotoa $2^n - 1$ oleviin lukuihin, jotka ovat alkulukuja. Tässä tutkielmassa varaamme näille alkuluvuille nimityksen Mersennen alkuluvut.)

Aluksi on hyvä huomata, että muotoa $2^n - 1$ oleva luku voi olla alkuluku vain, jos $n$ on alkuluku. Tämä käy ilmi seuraavan lauseen todistuksesta.

**Lause 6.1.** Olkoon $n \in \mathbb{N}$ yhdistetty luku. Tällöin $2^n - 1$ on yhdistetty luku.

*Todistus.* Todistusta varten johdamme ensin hyödyllisen aputuloksen. Olkoot $a$ ja $b$ reaalilukuja. Pyritään selvittämään lausekkeiden $(2^a - 1)$ ja $(2^{a(b-1)} + 2^{a(b-2)} + 2^{a(b-3)} + ... + 2^a + 1)$ tulo. Nähdään, että

$$(2^a - 1)\,(2^{a(b-1)} + 2^{a(b-2)} + 2^{a(b-3)} + ... + 2^a + 1)$$

$$= 2^{a+ab-a} - 2^{ab-a} + 2^{a+ab-2a} - 2^{ab-2a} + 2^{a+ab-3a} - 2^{ab-3a} + ... + 2^{a+a} - 2^a + 2^a - 1$$

$$= 2^{ab} - 2^{ab-a} + 2^{ab-a} - 2^{ab-2a} + 2^{ab-2a} - 2^{ab-3a} + ... + 2^{2a} - 2^a + 2^a - 1$$

$$= 2^{ab} \cancel{- 2^{ab-a} + 2^{ab-a}} \cancel{- 2^{ab-2a} + 2^{ab-2a}} \cancel{- 2^{ab-3a}} + ... \cancel{+ 2^{2a}} \cancel{- 2^a + 2^a} - 1$$

$$= 2^{ab} - 1.$$

Jatketaan varsinaista todistusta olettamalla, että $n$ on yhdistetty luku. On siis olemassa luonnolliset luvut $a, b \in [2, n-1]$, joilla $n = ab$. Nyt edellisen tarkastelun nojalla

$$2^n - 1 = 2^{ab} - 1 = (2^a - 1)(2^{a(b-1)} + 2^{a(b-2)} + 2^{a(b-3)} + ... + 2^a + 1).$$

On siis osoitettu, että $2^n - 1$ on luvun $2^a - 1$ monikerta. Lisäksi koska $a \geq 2$, niin $2^a - 1 > 1$.

Luku $2^n - 1$ on siis ykköstä suuremman luvun monikerta, joten se ei ole alkuluku. Tämä todistaa lauseen. □

*Huomautus* 6.2. Koska tässä luvussa käsitellään Mersennen lukuja, esitettiin lauseen 6.1 todistus muotoa $2^n - 1$ oleville luvuille. Yleisemmin pätee, että jos $n \in \mathbb{N}$ on yhdistetty luku, niin $k^n - 1$ on yhdistetty luku kaikilla ykköstä suuremmilla kokonaisluvuilla $k$. Tälle väitteelle voidaan antaa todistus, joka on käytännössä identtinen lauseen 6.1 todistuksen kanssa, sillä lauseelle 6.1 esitetty todistus etenee täsmälleen samalla tavalla kantaluvusta $k$ riippumatta.

### 6.1.2 Lucasin ja Lehmerin testi

Lucasin ja Lehmerin testi Mersennen luvuille noudattaa seuraavaa algoritmia:

- Olkoon $M_p = 2^p - 1$ testattava Mersennen luku.
- Määritellään rekursiivinen lukujono $(s_i)$ siten, että
  $s_0 = 4$ ja $s_{i+1} = {s_i}^2 - 2$ kaikilla $i \geq 0$.
- Nyt $M_p$ on alkuluku jos ja vain jos $s_{p-2} \equiv 0 \pmod{M_p}$.

*Huomautuksia* 6.3. Seuraavista havainnoista on usein apua, kun testi pyritään suorittaa mahdollisimman tehokkaasti.

- Lauseen 6.1 nojalla $M_p = 2^p - 1$ voi olla alkuluku vain, jos $p$ on alkuluku. Tästä syystä testin alkuun voidaan asettaa lisäehto, että testiä jatketaan vain, jos $p$ on alkuluku. Luku $p$ on eksponentiaalisesti pienempi kuin $M_p$, joten sen jaollisuus voidaan useimmiten selvittää verrattain tehokkaasti käyttämällä jotakin yksinkertaista menetelmää, kuten jakokokeilua.
- Luvun 2 potensseja voidaan yleisesti käsitellä tehokkaasti standardilla binäärijärjestelmään perustuvalla tietokoneella. Luvun $2^p$ binääriesitys on $p + 1$ -bittinen luku, jossa ensimmäinen bitti on 1, ja jälkimmäiset $p$ bittiä ovat 0, esimerkiksi $2^5 = 100000_2$. Luvun $2^p - 1$ binääriesityksessä puolestaan on $p$ bittiä, jotka ovat kaikki ykkösiä, esimerkiksi $2^5 - 1 = 11111_2$. Useimmista ohjelmointikielistä löytyy bittisiirto–operaatio, jolla jokainen bitti voidaan siirtää yhden askeleen verran vasemmalle tai oikealle. Bittien siirto vasemmalle vastaa luvulla 2 kertomista, esim $01001101_2 \cdot 2 = 10011010_2$. Esimerkiksi lukuun $M_p = 2^p - 1$ päästään lähtemällä liikkeelle luvusta 1, suorittamalla bittien siirto vasemmalle $p$ kertaa, ja vähentämällä saadusta tuloksesta lopuksi 1.
- Lukujonon $(s_i)$ termit kasvavat nopeasti hyvin suuriin arvoihin. Näiden lukujen selvittäminen ei kuitenkaan ole tarpeellista, jos hyödynnetään jo aiempien testien yhteydessä käytettyjä modulaarisen aritmetiikan tuloksia. Jos $s_i \equiv n \pmod{M_p}$, niin $s_{i+1} = {s_i}^2 - 2 \equiv n^2 - 2 \pmod{M_p}$. Suurten lukujen käsittelyltä voidaan jälleen välttyä käsittelemällä ainoastaan niiden jakojäännöksiä.

### 6.1.3 Lucasin ja Lehmerin testin suorittava ohjelma

Alla on esitetty mahdollisimman yksinkertainen versio Lucasin ja Lehmerin testin suorittavasta C++ -ohjelmasta.

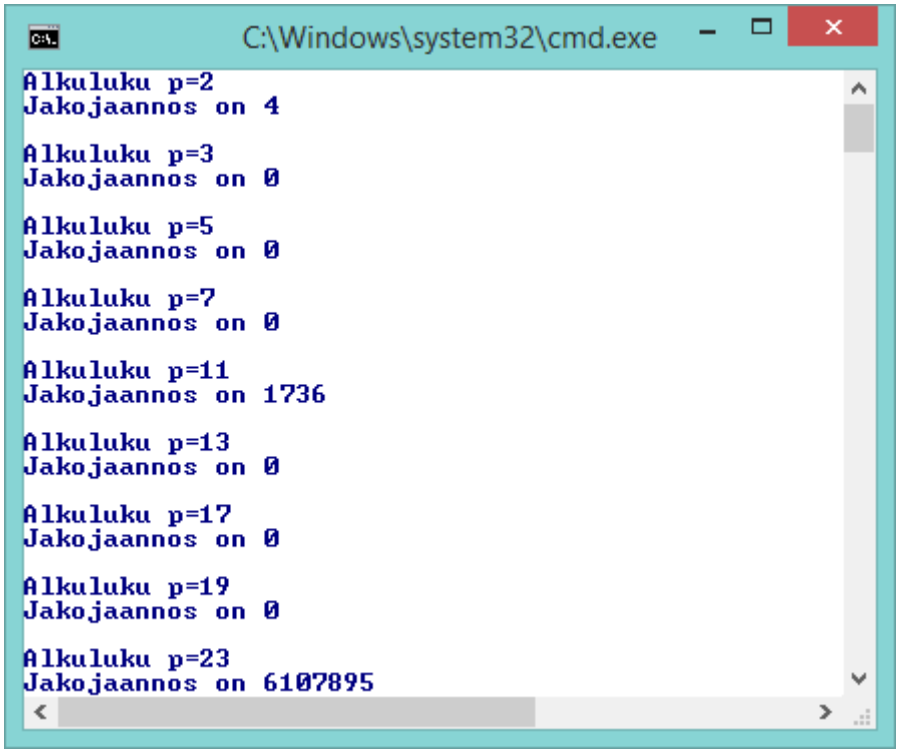
```
#include "stdafx.h"
#include "iostream"
using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
   while(true)
   {
   __int64 s=4;
   int p;
   cout << "Alkuluku p=";
   cin >> p;
   int M = (1 << p)-1; //Lähdetään luvusta 1, suoritetaan bittisiirto
                       //vasemmalle p kertaa, ja vähennetään 1, saadaan M=2^p-1.
   for (int a = 1; a < p-1; a++)
   {
      s = ((s*s)-2) %M;
   }
   cout << "Jakojaannos on " << s << "\n\n";

   }
   return 0;
}
```

On hyvä huomata, että jos $M$ on määritelty $n$–bittiseksi luvuksi ja sen arvo on riittävän lähellä $n$–bittisen luvun ylärajaa, niin myös $s$ voi saada *for*–silmukan aikana arvon, joka on lähellä $p$ –bittisen luvun ylärajaa. Tällöin seuraavalla kierroksella tehtävä laskutoimitus $s \cdot s$ voi ylittää kyseisen rajan, joten $s$ on hyvä määritellä $2n$–bittiseksi luvuksi. Yksinkertaisuutensa vuoksi esitetty koodi soveltuu vain pienten Mersennen lukujen tarkasteluun, sillä se toimii virheettömästi vain kun $p < 32$. Yleisimpiä ohjelmointikieliä varten on kuitenkin löydettävissä työkaluja suurten lukujen käsittelyä varten. Esimerkiksi *GMP*–ohjelmakirjastosta (*GNU Multiple Precision Arithmetic Library*) on olemassa versioita monelle ohjelmointikielelle, mukaan lukien C++ -kielelle. GMP mahdollistaa käytännössä kuinka tahansa suurten lukujen käsittelyn, ainoa rajoittava tekijä on käytettävissä oleva keskusmuisti. Erityisesti suuria Mersennen

lukuja käsiteltäessä on ehdottomasti kannattavaa tehostaa ohjelman toimintaa tarkastamalla ensin, onko $p$ alkuluku.




Kuva 6.1: Kuvakaappaus C++ -ohjelmasta, joka saatuaan käyttäjältä luvun $p$ suorittaa Lucasin ja Lehmerin testin Mersennen luvulle $M_p = 2^p - 1$. Testatuista luvuista $M_{11}$ ja $M_{23}$ ovat yhdistettyjä lukuja, muut ovat alkulukuja. Lucasin ja Lehmerin testissä oletetaan, että $p > 2$, ja tämän vuoksi testatessa lukua $M_2$ saadaan virheellisesti tulos, jonka mukaan kyseessä olisi yhdistetty luku.

Ohjelman toimintaa on lisäksi mahdollista tehostaa seuraavan mielenkiintoisen havainnon avulla: kokonaisluvuille $n$ ja $p$ on voimassa

$$n \equiv \left( \lfloor \frac{n}{2^p} \rfloor + (n \bmod 2^p) \right) \pmod{2^p - 1}.$$

Tässä $\lfloor \frac{n}{2^p} \rfloor$ on lattiafunktio, joka antaa suurimman kokonaisluvun, joka on lukua $\frac{n}{2^p}$ pienempi. Hieman epätyylikästä merkintää $(n \bmod 2^p)$ on käytetty tässä yksinkertaisuuden vuoksi. Sillä tarkoitetaan jakojäännöstä, joka saadan jaettaessa luku $n$ luvulla $2^p$.

Tämä tulos on erityisen hyödyllinen, kun käytössä on standardi binäärijärjestelmällä laskeva tietokone. Laskuoperaatiossa ($n \bmod 2^p$) luvusta $n$ jaetaan pois ($p+1$ –bittisen) luvun $2^p$ monikerrat, jolloin jäljelle jää luvun $n$ viimeiset $p$ bittiä. Luvun $n/2^p$ kokonaislukuosa saadaan puolestaan karsimalla viimeiset $p$ bittia, ja suorittamalla jäljellä olevien bittien siirto oikealle $p$ kertaa. Seuraavat esimerkit havainnollistavat, miten tätä tulosta voidaan hyödyntää silloin, kun laskujärjestelmän kantaluku on 2.

**Esimerkki 6.4.** Selvitetään luvun 3374 jakojäännös, kun jakajana on luku $2^6 - 1$.
Edellä esitetyn tuloksen mukaan

$$3374 \equiv \Big( \lfloor \frac{3374}{2^6} \rfloor + (3374 \bmod 2^6) \Big) \pmod{2^6 - 1}.$$

Luvun 3374 binääriesitys on $110100101110_2$.
Jakojäännös ($n \bmod 2^6$) on tämän 6 viimeistä bittiä, eli ($n \bmod 2^6$) $= 101110_2$.
Lattiafunktio antaa alkupäästä jäljelle jäävät bitit, eli $\lfloor \frac{3374}{2^6} \rfloor = 110100_2$.
Siis $3374 \bmod 2^6 - 1$
$= 110100101110_2 \bmod 2^6 - 1$
$= (110100_2 + 101110_2) \bmod 2^6 - 1$
$= 1100010_2 \bmod 2^6 - 1$ || käytetään samaa menetelmää toisen kerran
$= (100010_2 + 1_2) \bmod 2^6 - 1$
$= 100011_2 \bmod 2^6 - 1$ || $2^6 - 1 = 111111_2 > 100011_2$
$= 100011_2 = 35$.

Kun luku 3374 jaetaan luvulla $2^6 - 1$, on jakojäännös siis 35.

## 6.2 Pépinin testi Fermat'n luvuille

### 6.2.1 Fermat'n luvuista

Fermat'n luvuiksi kutsutaan niitä lukuja $F_n$, jotka ovat muotoa $F_n = 2^{2^n} + 1$, missä $n \geq 0$ on kokonaisluku. Tästä määritelmästä seuraa, että Fermat'n lukujen muodostama jono kasvaa hyvin nopeasti: ensimmäiset kuusi Fermat'n lukua ovat 3, 5, 17, 257, 65 537 ja 4 294 967 297. Lukujen suuruuden vuoksi myös niiden käsittely on haastavaa ja aikaavievää. Edellä esitetyistä kuudesta Fermat'n luvusta viisi ensimmäistä on alkulukuja, ja ne ovat myös ainoat tunnetut Fermat'n alkuluvut, sillä lukua $F_4 = 65\,537$ suurempia Fermat'n alkulukuja ei ole löydetty. Sen sijaan lähes 300 Fermat'n lukua on osoitettu yhdistetyiksi luvuiksi.

### 6.2.2 Pépinin testi

Pépinin testi on deterministinen testi Fermat'n luvuille. Testissä tarkistetaan, toteutuuko ehto

$$3^{\frac{F_n+1}{2}} \equiv -1 \pmod{F_n}.$$

Jos ehto toteutuu, $F_n$ on alkuluku. Muutoin $F_n$ on yhdistetty luku.

Laskettaessa luvun $3^{\frac{F_n-1}{2}}$ jakojäännöstä on hyvä soveltaa toistetun neliöinnin menetelmää, joka esiteltiin aiemmin esimerkissä 4.4. Alla on tarkasteltu lukua $F_3$ Pépinin testin avulla.

**Esimerkki 6.5.** Selvitetään, onko Fermat'n luku $F_3 = 2^{2^3} + 1 = 257$ alkuluku.

Nyt $\frac{F_3 - 1}{2} = 128$, joten on selvitettävä $3^{128} \bmod 257$.

Käytetään toistetun neliöinnin menetelmää:

$3^2 \equiv 9 \pmod{257}$
$3^4 \equiv 81 \pmod{257}$
$3^8 \equiv 6561 \equiv 136 \pmod{257}$
$3^{16} \equiv 18496 \equiv 249 \pmod{257}$
$3^{32} \equiv 62001 \equiv 64 \pmod{257}$
$3^{64} \equiv 4096 \equiv 241 \pmod{257}$
$3^{128} \equiv 58081 \equiv 256 \equiv -1 \pmod{257}$.

Luku $F_3$ toteuttaa Pépinin testin ehdon, joten se on alkuluku.

# Luku 7

# Nopeat deterministiset testit

Nopeat deterministiset testit ovat verrattain tuore löytö matematiikassa, sillä ensimmäiset tämänkaltaiset testit kehitettiin vasta 2000-luvulla. Nopealla deterministisellä testillä viitataan tässä yhteydessä sellaiseen testiin, joka toteuttaa seuraavat neljä ehtoa:

- Testi on *deterministinen*, eli se antaa varman tiedon luvun jaollisuudesta. (Vertaa probabilistisiin testeihin, kuten Millerin ja Rabinin testiin.)
- Testi on *nopea*, eli algoritmin suoritusaikaa rajoittaa testattavasta luvusta $n$ riippuva polynomi $P(n)$. (Vertaa esimerkiksi Wilsonin testiin, jossa testin suoritusaika riippuu eksponentiaalisesti testattavasta luvusta).
- Testi on *yleinen*, eli sen avulla voidaan testata minkä tahansa luonnollisen luvun $n$ jaollisuus. (Useat nopeat testit soveltuvat vain tiettyä muotoa oleville luvuille. Tästä esimerkkinä Lucasin ja Lehmerin testi Mersennen luvuille.)
- Testi on *ehdoton*, eli sen oikeellisuus ei riipu todistamattomista hypoteeseista tai konjektuureista. (Vertaa PSW-testiin, joka olettaa todistamattoman PSW-konjektuurin oikeellisuuden.)

Ensimmäisenä yllä mainitut ehdot täyttävän testin kehittivät intialaiset matemaatikot ja tietojenkäsittelytieteilijät Manindra Agrawal, Neeraj Kayal ja Nitin Saxena vuonna 2002.

## 7.1 AKS-testin taustaa

Seuraava lause on Fermat'n pienen lauseen yleistys polynomeille.

**Lause 7.1.** Olkoot $n \geq 2$ ja $a$ luonnollisia lukuja, joiden suurin yhteinen tekijä on 1. Luku $n$ on alkuluku, jos ja vain jos kongruenssirelaatio

$$(x+a)^n \equiv (x^n + a) \pmod{n}$$

on voimassa.

On syytä huomata, että kyseessä on kahden polynomin välinen relaatio, jossa $x$ tulisi ymmärtää formaalina symbolina, eikä niinkään reaalilukuarvoja saavana muuttujana. Formaalissa algebrassa kaksi polynomia ovat määritelmän mukaan yhtä suuret silloin, jos samoilla samoilla objektin $x$ potensseilla on samat kertoimet. Toisin sanoen, polynomeille

$$p_1 = a_0 + a_1 x + a_2 x^2 + ... + a_m x^m \quad \text{ja} \quad p_2 = b_0 + b_1 x + b_2 x^2 + ... + b_m x^m$$

pätee, että

$$p_1 = p_2 \quad \Leftrightarrow \quad a_k = b_k \quad \text{kaikilla } k \in [0, m].$$

Vastaavasti polynomit ovat kongruentteja modulo $n$, jos samojen objektin $x$ potenssien kertoimet ovat kongruentteja modulo $n$, eli

$$p_1 \equiv p_2 \pmod{n} \quad \Leftrightarrow \quad a_k \equiv b_k \pmod{n} \quad \text{kaikilla } k \in [0, m].$$

Seuraavassa esimerkissä on tutkittu lukuja 5 ja 6 käyttäen lausetta 7.1 alkulukutestinä.

**Esimerkki 7.2.** Selvitetään lausetta 7.1 hyödyntäen, onko luku 5 alkuluku. Valitaan testaajaksi $a$ luku 2, jolloin ehto $syt(5, a) = 1$ täyttyy. Binomilauseen (tai Pascalin kolmion) avulla saadaan

$$(x+2)^5 = x^5 + 5x^4 2 + 10x^3 2^2 + 10x^2 2^3 + 5x 2^4 + 2^5 \equiv x^5 + 2^5 \pmod{5}.$$

Nyt siis lauseen 7.1 perusteella 5 on alkuluku, jos relaatio $x^5 + 2^5 \equiv x^5 + 2 \pmod{5}$ on voimassa. Selvästi termien $x^5$ kertoimet ovat kongruentteja, sillä

$$1 \equiv 1 \pmod{5}.$$

Lisäksi termien $x^0$ kertoimet ovat kongruentteja, sillä

$$2^5 = 32 \equiv 2 \pmod{5}.$$

Näin ollen polynomit $(x+a)^5$ ja $x^5 + a$ ovat kongruentteja modulo 5, joten luku 5 on alkuluku.

Tutkitaan sitten lukua 6. Valitaan testaajaksi $a$ luku 5, jolloin $syt(6, a) = 1$. On selvitettävä, päteekö kongruenssirelaatio

$$(x+5)^6 = x^6 + 6x^5 5 + 15x^4 5^2 + 20x^3 5^3 + 15x^2 5^4 + 6x5^5 + 5^6 \equiv x^6 + 5 \pmod 6.$$

Tutkitaan polynomien neljännen asteen termien kertoimia. Vasemmalla puolella kerroin on $15 \cdot 5^2$, oikealla puolella se on 0. Huomataan, että

$$15 \cdot 5^2 = 375 \equiv 3 \not\equiv 0 \pmod 6.$$

Polynomien neljännen asteen termien kertoimet eivät siis ole kongruentteja modulo 6, joten luku 6 ei ole alkuluku.

*Huomautus* 7.3. Ensimmäisessä esimerkissä esiintynyt kongruenssirelaatio $2^5 \equiv 2 \pmod 5$ pätee selvästi jo Fermat'n lauseen nojalla, koska 5 on alkuluku. Fermat'n lauseen käyttö kyseisessä esimerkissä olisi kuitenkin verrattavissa kehäpäätelmään, sillä luvun 5 jaottomuutta oltiin vasta selvittämässä.

Lausetta 7.1 voidaan siis käyttää alkulukutestinä, mutta sen käyttö on kuitenkin hidasta. Polynomin $(x-a)^n$ kerrointen selvittäminen ja niiden jaollisuuden tarkastaminen johtavat aikavaatimukseen, joka riippuu eksponentiaalisesti alkulukuehdokkaasta $n$. Lauseen pohjalta voidaan kuitenkin johtaa ehto, joka voidaan tarkastaa polynomisessa ajassa.

Lauseen 7.1 kongruenssirelaatio on polynomirenkaan $\mathbb{Z}_n[x]$ relaatio. Tämän sijaan kongruenssirelaatiota voidaan tarkastella jossakin polynomirenkaan $\mathbb{Z}_n[X]$ jäännösluokkarenkaassa $\mathbb{Z}_n[x]/(P(x))$, missa $P(x)$ on jokin polynomi. Tämä asettaa ylärajan tarkasteltavan polynomin asteluvulle. AKS-testissä kongruenssirelaatio tarkistetaan renkaassa $\mathbb{Z}_n[x]/(x^r - 1)$: Lauseesta 7.1 seuraa, että kongruenssirelaatio

$$(x+a)^n \equiv (x^n + a) \pmod{x^r - 1,\ n}$$

on voimassa alkuluvuille. Selvennyksen vuoksi: yllä esitetty ehto on yhtäpitävä sen kanssa, että on olemasssa polynomit $P$ ja $Q$, joille

$$(x+a)^n - (x^n + a) = n \cdot P + (x^r - 1) \cdot Q.$$

Tässä vaiheessa ongelmaksi muodostuu kuitenkin se, että myös jotkut yhdistetyt luvut saattavat toteuttaa yllä esitetyn yhtälön joillakin luvuilla $a$ ja $r$. Agrawal, Kayal ja Saxena kuitenkin osoittavat, että on mahdollista valita sopiva luku $r$, ja tarkistaa yhtälö tietyillä testaajan $a$ arvoilla niin, että menetelmä antaa varman tiedon luvun $n$ jaollisuudesta polynomisessa ajassa.

AKS-testin algoritmi ja sen perustelu sivuutetaan tässä vaiheessa, sillä se on huomattavasti aiemmin käsiteltyjä menetelmiä monimutkaisempi. Agrawalin, Kayalin ja Saxenan alkuperäinen artikkeli "PRIMES is in P" on löydettävissä Indian Institute of Technology Kanpurin verkkosivuilta (https://www.cse.iitk.ac.in/users/manindra/algebra/primality_v6.pdf). AKS-testin algoritmi on esitelty artikkelin sivulla 3.

### 7.1.1 AKS-testin merkityksestä

AKS-testi on matematiikan ja tietojenkäsittelytieteen teorian kannalta merkittävä löytö, sillä se osoittaa, että minkä tahansa luvun jaollisuuden selvittämiseen tarvittava aika on annetusta luvusta riippuvan polynomin rajoittama. Käytännön sovellusten kannalta testi ei kuitenkaan ole hyödyllinen, sillä se on muita vaihtoehtoja huomattavasti hitaampi. Agrawalin, Kayalin ja Saxenan alkuperäisen algoritmin aikavaatimukselle saadaan ylärajaksi $O((\log_2 n)^{12})$. Tässä "iso $O$ notaatio" eli asymptoottinen suoritusaika kuvaa sitä, kuinka monta askelta algoritmin täytyy enintään suorittaa. Kuriositeettina tätä voidaan verrata tutkielman alussa esitetyn Erastotheneen seulan algoritmin aikavaatimukseen, jonka ylärajaksi saadaan $O(n \log_2 \log_2 n)$. Tehdään nopea heuristinen vertailu näiden välillä.

Tutkitaan kuvauksen $f : [1, \infty[ \to \mathbb{R}, f(x) = x \log_2 \log_2 x - (\log_2 n)^{12}$ merkkiä.
Tarkastelun yksinkertaistamiseksi merkitään $x = 2^y$, jolloin $f(x) = 2^y \log_2 y - y^{12}$.
Pikaisen numeerisen tarkastelun avulla nähdään, että
$f(x) < 0$ kun $y \leq 71$ ja $f(x) > 0$ kun $y \geq 72$.

Huomataan siis, että ainakin kaikilla $n \leq 2^{71}$ Erastotheneen seulalla on pienempi aikavaatimuksen maksimi kuin AKS-testillä. On siis hyvinkin mahdollista, että tällaisen luvun $n$ jaollisuuden selvittäminen AKS-testillä vaatisi enemmän aikaa kuin kaikkien korkeintaan luvun $n$ suuruisten alkulukujen tunnistaminen Erastotheneen seulalla.

Vuonna 2005 Carl Pomerance ja Hendrik Lenstra esittivät muokatun version testistä, jonka suoritusajan maksimi on $O((\log_2 n)^6)$. Vaikka tämä on merkittävä parannus alkuperäiseen testiin verrattuna, se on edelleen muita vaihtoehtoja hitaampi käytännön sovellusten kannalta.

# Luku 8

# Yhteenveto

Tässä tutkielmassa on pyritty mahdollisimman kattavasti esittämään keinoja alkulukujen testaamiseksi. Yksinkertaisin tapa selvittää annetun luvun $n$ jaollisuus on kokeilla, voidaanko se jakaa tasan jollakin välin $[2, \sqrt{n}]$ luonnollisella luvulla. Tämä menetelmä on kuitenkin hyvin työläs. Erastotheneen seula puolestaan on kätevä työkalu, jos halutaan selvittää kaikki alkuluvut johonkin luonnolliseen lukuun $m$ asti, mutta myös tämä algoritmi on hidas suurilla luvuilla $m$. Yksinkertaisena deterministisenä testinä voidaan käyttää Wilsonin lausetta, jonka mukaan luonnolliselle luvulle $n > 1$ pätee $(n-1)! \equiv -1 \pmod{n}$ jos ja vain jos $n$ on alkuluku. Koska tämä vaatii $n$ modulaarista kertolaskua, muuttuu tämäkin menetelmä työlääksi testattavan luvun ollessa suuri.

Edellä kuvatut menetelmät ovat deterministisiä, eli ne antavat varman tiedon siitä, onko tutkittava luku alkuluku. Esimerkkejä heuristisista testeistä puolestaan ovat Fermat'n testi, Fibonaccin testi ja PSW-testi. Fermat'n testi perustuu Fermat'n pieneen lauseeseen, jonka mukaan alkuluvulle $p$ ja kokonaisluvulle $a$ on voimassa kongruenssirelaatio $a^p \equiv a \pmod{p}$. Mikäli testattava luku $p$ ei noudata Fermat'n lausetta jollakin testaajan $a$ arvolla, tiedämme, että $p$ on yhdistetty luku. Relaatio saattaa kuitenkin olla voimassa myös yhdistetyn luvun tapauksessa, joten testin läpäisevä luku ei välttämättä ole alkuluku. Toinen vastaavanlainen testi on Fibonaccin testi. Se hyödyntää tietoa, että numeroon 1 tai 9 päättyvä alkuluku $p$ jakaa Fibonaccin luvun $F_{p-1}$ tasan, kun taas numeroon 3 tai 7 päättyvä alkuluku $p$ jakaa Fibonaccin luvun $F_{p+1}$ tasan. Fermat'n testin ja Fibonaccin testin yhdistelmänä saadaan PSW-testi. PSW-testissä numeroon 3 tai 7 päättyvää lukua $p$ testataan ensin Fermat'n testillä valitsemalla kantaluvuksi $a = 2$, ja mikäli luku $p$ läpäisee Fermat'n testin, suoritetaan vielä Fibonaccin testi. PSW-testi on esimerkki sellaisesta heuristisesta testistä, jolle yhtäkään testin läpäisevää yhdistettyä lukua ei ole löydetty.

Probabilistiset testit ovat menetelminä matemaattisesti täsmällisempiä kuin heuristiset testit. Probabilistinen testi antaa jonkin todistettavissa olevan rajan sille todennäköisyydelle, että yhdistettyä lukua ei tunnisteta testin avulla. Esimerkkejä probabilistisista testeistä ovat

Solovayn ja Strassenin testi sekä tässä tutkielmassa käsitelty Millerin ja Rabinin testi. Millerin ja Rabinin testissä alkulukuehdokkaalle $p > 2$ etsitään pariton luku $d$, jolle $2^s d = p - 1$ jollakin $s \in \mathbb{N}$. Jos ehdot

$$a^d \not\equiv 1 \pmod{p} \quad \text{ja} \quad a^{2^r d} \not\equiv -1 \pmod{p}$$

täyttyvät kaikilla kokonaisluvuilla $r \in [0, s-1]$, niin $p$ on yhdistetty luku. Jos jokin ehdoista ei täyty, $p$ saattaa olla alkuluku. Jokaista paritonta yhdistettyä lukua kohti vähintään $3/4$ testaajan $a$ arvoista on sellaisia, että ne toteuttavat testin ehdot, jolloin yhdistetty luku voidaan tunnistaa. Testin luotettavuus on sitä parempi, mitä useammalla testaajan $a$ arvolla se suoritetaan. Jos halutaan selvittää, millä todennäköisyydellä testin läpäissyt luku on yhdistetty luku, on yleensä turvauduttava Bayesin lakiin. Tällöin tarvitaan jotakin tietoa siitä, millä tiheydellä alkulukuja esiintyy luvun $p$ ympäristössä.

Jotkut alkulukutestit soveltuvat vain tiettyä muotoa olevien lukujen testaamiseen. Esimerkiksi aiemmin mainittu PSW-testi soveltuu vain sellaisten lukujen testaamiseen, jotka päättyvät numeroon 3 tai 7. Muita esimerkkejä erityistä muota oleville luvuille laadituista testeistä ovat Lucasin ja Lehmerin testi Mersennen luvuille (muotoa $M_p = 2^p - 1$ olevat luvut) sekä Pépinin testi Fermat'n luvuille (muotoa $F_n = 2^{2^n} + 1$ olevat luvut). Lucasin ja Lehmerin testissä määritellään rekursiivinen lukujono $(s_i)$ asettamalla $s_0 = 4$ ja $s_{i+1} = {s_i}^2 - 2$ kaikilla $i \geq 0$. Testissä tarkistetaan, toteutuuko ehto

$$s_{p-2} \equiv 0 \pmod{M_p}.$$

Ehdon toteutuessa $M_p$ on alkuluku, muutoin se on yhdistetty luku. Testi on siis deterministinen. Pépinin testissä puolestaan tarkastetaan, toteutuuko ehto

$$3^{\frac{F_n+1}{2}} \equiv -1 \pmod{F_n}.$$

Myös Pépinin testi on deterministinen, eli ehto toteutuu alkuluvuilla ja ainoastaan alkuluvuilla.

Nopeat deterministiset testit puolestaan ovat verrattain uusi löytö matematiikassa. AKS-testi on ensimmäinen kaikille luvuille soveltuva, polynomisessa ajassa suoritettava ja todistamattomista konjektuureista riippumaton deterministinen alkulukutesti. Se laadittiin vuonna 2002. Kyseinen testi on kuitenkin hyvin hidas käytännön alkulukutestaamisen kannalta. Onkin hyvä huomata, että nopean deterministisen testauksen yhteydessä sana "nopea" viittaa ainoastaan siihen, että algoritmin suoritusajan maksimi voidaan ilmaista testattavan luvun $n$ polynomina $P(n)$, eli luvun $n$ potenssina tai potenssien summana. Oleellisesti "nopeus" tarkoittaa siis tässä yhteydessä sitä, että kasvu ei ole eksponentiaalista, eli luku $n$ ei itse esiinny eksponenttina suoritusajan maksimissa. Käytännön sovellusten kannalta nopeatkin deterministiset testit ovat hyvin hitaita, sillä niiden suoritusajan maksimi riippuu usein jostakin suuresta luvun $n$ potenssista.

# Kirjallisuutta


[1] Häsä, J., Rämö, J. 2012. Johdatus abstraktiin algebraan.

[2] Crandall, R., Pomerance, C. 2005. Prime Numbers: A Computational Perspective (2nd ed.).

[3] Apostol, T. 1976. Introduction to analytic number theory.

[4] Long, Calvin T. 1972. Elementary Introduction to Number Theory.

[5] Albert, A. 2015. Modern Higher Algebra.

[6] Dietzfelbinger, M. 2004. Primality Testing in Polynomial Time. From Randomized Algorithms to "PRIMES Is in P".

[7] Cormen, T., Leiserson, C., Rivest, R., Stein, C. 2001. Introduction to Algorithms, Third Edition.

[8] Holopainen I., Mirka, A. 2015. Mitta ja Integraali, luentomoniste, Helsingin yliopisto.

[9] http://primes.utm.edu/howmany.html
luettu 13.10.2018

[10] https://www.dcode.fr/modular-exponentiation-calculus
luettu 13.10.2018

[11] https://www.encyclopediaofmath.org/index.php/Wilson_theorem
luettu 13.10.2018

[12] http://rosettacode.org/wiki/Lucas-Lehmer_test#C.2B.2B
luettu 13.10.2018

[13] https://artofproblemsolving.com/wiki/index.php?title=Wilson's_Theorem
luettu 13.10.2018

[14] http://mathworld.wolfram.com/FibonacciPrime.html
luettu 13.10.2018

[15] https://en.wikipedia.org/wiki/Primality_test
luettu 13.10.2018

[16] https://en.wikipedia.org/wiki/Lucas–Lehmer_primality_test
luettu 13.10.2018

[17] https://en.wikipedia.org/wiki/Proofs_of_Fermat's_little_theorem
luettu 13.10.2018

[18] Agrawal, M., Kayal, N., Saxena, N. 2004. PRIMES is in P.
Lähde: https://www.cse.iitk.ac.in/users/manindra/algebra/primality_v6.pdf
luettu 13.10.2018

[19] Bernstein, Daniel J. 2003. Proving Primality After Agrawal-Kayal-Saxena.
Lähde: https://cr.yp.to/papers/aks.pdf
luettu 13.10.2018