

Viikkoraportti 3

Lasse Lybeck, 013748498

Aineopintojen harjoitustyö: Tietorakenteet ja algoritmit

Tällä viikolla olen työstänyt Strassenin algoritmia matriisien kertolaskulle. Olin jo valmiiksi tehnyt testit tarkistamaan, että Strassenin algoritmilla saatu tulos on sama kuin normaalin kertolaskun, joten metodi tarvitsi vain toteuttaa. Lisäksi työstin hieman käyttöliittymää, lisäämällä sinne kertolaskun vertailun. Vertailu tuli tarpeelliseksi kun Strassenin algoritmi tuli valmiiksi, näin kertolakuihin menevää aikaa pääsi helposti vertailemaan.

Kertolaskujen vertailusta huomasi kuitenkin nopeasti, että normaali kertolasku toimi erittäin paljon Strassenin algoritmia nopeammin. Toteutin ensin Strassenin suoraan käyttäen Matrix-luokkaa, mutta muutin sen myöhemmin käyttämään suoraan `double[][]`-tyyppiä. Näin pääsin eroon turhista tarkistuksista uusia matriiseja luodessa. Tämä ei kuitenkaan vaikuttanut merkittävästi kertolaskun nopeuteen, vaan se jäi edelleen paljon normaalia kertolaskua hitaammaksi.

Luulen, että Strassenin toteuttaminen javalla tekee siitä niin hitaan kuin se nyt vaikuttaa olevan. Algoritmissa luodaan ja tuhotaan suuria määriä matriiseja, ja tämä ei javassa ole kovinkaan nopeaa. Yhtenä ongelmana saattaa olla, että java alustaa kaikki matriisin alkiot ensin nolliksi kun matriisit luodaan kopioiksi toisista matriiseista. Tämän jälkeen alkiot muutetaan niille kuuluviksi arvoiksi, joten alkiot joudutaan alustamaan kahteen kertaan. Kun jokaisessa rekursiivisessa kutsussa luodaan yhteensä 13 uutta matriisia, tarkoittaa tämä että jo 8x8 matriisien kertolaskussa yli 1400 double-muuttujaa alustetaan turhaa nolliksi, ja 64x64 matriisien kohdalla luku on jo yli 100 000. Lisäksi kaikkien näiden matriisien luominen ja hylkääminen aktivoi roskienkerääjää erittäin paljon, joka taas vie resursseja itse algoritmilta.

Olisi kiinnostavaa tietää kuinka vastaava algoritmi toimisi nopeammissa kielissä, kuten C++:ssa tai C:ssä. Jos periodin loppupuolella löytyy aikaa, voisin kääntää saman algoritmin esim. C:hen, ja katsoa onko tästä apua tehokkuuden saavuttamisessa. Näin voisi saada selvyyttä sille, onko algoritmi vain niin paljon hitaampi pienille matriiseille kuin vaikuttaa, vai onko vika javassa tai minun toteutuksessani.

Testeissä vaikuttaa siltä, että Strassenin algoritmi kuitenkin saavuttaa merkittävästi normaalia kertolaskua. Kun matriisit ovat suhteellisen pieniä (korkeintaan 512x512) kestää Strassenin kertolasku yli sata kertaa niin kauan kuin normaali kertolasku, mutta kun matriisit ovat kokoa 1024x1024 kestää Strassen enää 17 kertaa niin kauan kuin normaali kertolasku (normaali 11s, Strassen 211s), ja 2048x2048 enää 10 kertaa niin kauan (normaali 2.3min, Strassen 24min). Tätä voisi vertailla myöhemmin kurssin aikana, ja katsoa kuinka suuria matriisien tulisi olla, että tästä Strassenin toteutuksesta olisi oikeasti hyötyä.

Ajattelin nyt jättää Strassenin syrjään seuraavaksi viikoksi, ja palata siihen myöhemmin jos aikaa riittää. Suunnitelmissa olisi seuraavaksi LU-hajotelman ja determinantin laskennan toteuttaminen ja testaaminen. Jos ensi viikolla riittää aikaa voisin myös jatkaa käyttöliittymän tekemistä, että matriiseilla pystyisi myös pian laskemaan.