

Viikkoraportti 2

Lasse Lybeck, 013748498

Aineopintojen harjoitustyö: Tietorakenteet ja algoritmit

Tällä viikolla olen saanut harjoitustyön tekemisen kunnolla käyntiin. Luokkia harjoitustyöhön ei vaikuta tulevan montaa, vaikka koodia kyllä varmasti tulee riittävästi.

Loin luokan Matrix ohjelman ydinluokaksi. Tämä on siis luokka, joka mallintaa matriisia. Syy siihen, etten käytä suoraan primitiivisiä taulukoita on, että näin voin olla varma, ettei taulukossa ole esim. eri pituisia rivejä, eli että taulukko on aina varmasti matriisi. Lisäksi luokka pystyy nyt sisältämään lisää tietoa, kuten rivien ja sarakkeiden määrän.

Rakenteen selkeyttämiseksi loin Operations-luokan, joka sisältää suurimman osan matriisioperaatioista. Näin aivan kaikki koodi ei ole suljettuna Matrix-luokkaan, vaan tämä kirjastoluokka sisältää algoritmit operaatioiden takana. Matrix-luokassa myös osa tiedoista, kuten matriisin data, on protected suojattua. Näin Operations-luokka pääsee näihin suoraan käsiksi, mutta jokin käyttäjä ulkopuolelta ei pääse sorkkimaan näitä tärkeitä tietoja. Operations-luokka sisältää oleellisesti samat metodit kuin Matrix-luokka (tosin staattisina), joten kun kutsutaan jotakin operaatiota matriisille, kutsuu matriisi edelleen samaa metodia operaatiokirjastosta.

Itse algoritmit eivät vielä tällä viikolla ole olleet vaikeita, olen saanut valmiiksi suurimman osan helpoimmista operaatioista, kuten matriisien yhteenlaskun ja matriisien kertolaskun koulualgoritmin. Olen kuitenkin luonut pohjat myös muutamalle muulle operaatiolle, jotka olen päättänyt toteuttaa.

Olen myös tehnyt JUnit-testejä kaikille jo valmiina oleville operaatioille, ja valmiiksi ainakin matriisien kertolaskun toiselle toteutettavalle algoritmillemme. Yritän noudattaa pitkälti TDD (Test Driven Development) tyyliä koodin luomisessa, eli luon jollekin operaatiolle ensin testit, sen jälkeen toteutan operaation, ja kun testit menevät läpi siirryn seuraavaan. Ainakin vaikeimmissa operaatioissa tämä on mielestäni helpoin (tai nopein) tapa saada algoritmit toimiviksi.

Olen myös hieman aloittanut käyttöliittymän tekemistä ohjelmalleni. Tarkoituksena olisi tehdä matlab-tyylinen käyttöliittymä, jossa voidaan suorittaa tärkeimmät matriisi-operaatiot. Olen saanut ratkaistua miten säilön käyttäjän tallentamat muuttujat muistiin. Tein tämän käyttämällä HashMap-rakennetta, jossa muuttujan nimi String-muodossa liitetään johonkin matriisiin. Lisäksi olen luonut käyttöliittymään new-komennon, jonka avulla uuden muuttujan voi luoda. Tämän enempää en ole kuitenkaan vielä tehnyt käyttöliittymää, vaan haluan mieluummin ensin keskittyä itse matriisioperaatioiden toteuttamiseen ja testaamiseen.

Seuraavaksi askeleeksi olen ajatellut haastavampien matriisioperaatioiden toteuttamista. Olen jo hieman tutustunut matriisien kertolaskun Strassenin algoritmiin, ja kirjoittanut testit tälle. Seuraavan viikon aikana tarkoitukseni olisi saada ainakin tämä algoritmi valmiiksi. Testauksen olen toteuttanut niin, että ensin testaan, että naiivi kertolaskualgoritmini toimii kattavilla testeillä, ja tämän jälkeen testaan antaako Strassenin algoritmi saman vastauksen. Tietysti tämä tarkoittaa, että en saa tietää toimiiko Strassenin algoritmi oikein, jos naiivissa kertolaskussa on jotain häikkää, mutta koska se on melko helppo toteuttaa onnistuneesti (ja olen sen jo tehnyt), on tämä lähestymistapa mielestäni järkevä.

Suunnitelmissani on myös toteuttaa kertolasku, joka hyödyntää sekä Strassenin että naiivia algoritmia isojen matriisien kertolaskuihin. Kun olen saanut Strassenin algoritmin suoritettua, aion testata onko se oikeasti (tai mistä kooista ylöspäin se on) tehokkaampi kuin naiivi algoritmi, ja tämän mukaan toteuttaa kertolaskun niin, että tarpeeksi suurille matriiseille sovelletaan aluksi Strassenin algoritmia, ja kun se on pilkkonut matriisin tarpeeksi pieniin osiin, suoritetaan kertolasku loppuun naiivilla algoritmilla. En

ole kuitenkin varma onko tämä järkevä lähestymistapa, mutta olen kiinnostunut näiden algoritmien tehoeroista, joten siitä saa ainakin jonkinlaisen dokumentaation aikaan, vaikka Strassenin algoritmi osoittautuisi (ainakin minun toteuttamana) tehottomaksi.