

Tiralabran viikkoraportit

Viikko 0

Tällä viikolla pääsin ratkaisuun harjoitustyöni aiheesta ja aloin päästä perille siitä mitä kaikkea siihen tulee sisältymään. Etsiäni tietoa aiheesta ja sen toteutustavoista tutustuin ensimmäistä kertaa matriisihajotelmiin ja opin niitten toiminnasta ja tehokkuudesta ja siitä kuinka voisin hyödyntää niitä työssäni. Uuden oppimisen lisäksi palautin mieleen myös vanhoja ja osin unohtuneita asioita tiran kurssilta ja selvitin millaisia tietorakenteita ja algoritmeja minun kannattaa toteuttaa työssäni.

Hiukan jäi tällä viikolla vielä epäselväksi se, mitkä ovat kurssilla matriisilaskimelle asetetut tila- ja aikavaativuudet. Alustavasti suunnittelin kuitenkin aikavaativuudeksi vähintään $O(n^3)$ ja jos aikaa jää, voisin yrittää pyrkiä $O(n^{2.8074})$ tehokkuuteen. Epäselväksi jäi myös se, mitä työstä on löydyttävä päästäkseen läpi ja mitä kaikkea on oltava vitosen saavuttamiseksi.

Ohjelman suunnitteluvaihe alkaa olla valmis ja on aika siirtyä itse toteutukseen. Seuraavan palautuksen jälkeen pitäisi olla valmiina sellainen ohjelma, joka osaa laskea kahden matriisin summan ja erotuksen, laskea kahden matriisin tulon ja toteuttaa vakioilla kertomisen. Matriisin

vastaanottaminen syötteenä ja tallennus voidaan toteuttaa jo nyt jos aikaa jää.

Viikko 1

Tällä viikolla opin (uudelleen) käyttämään mavenia ja kehityin JUnit- testien toteutuksessa ja suunnittelussa.

Täysin uutta opittua asiaa oli matriisien ja niiden laskutoimituksia laskevan koodin toteuttaminen.

Matriisikertolaskun kohdalla jouduin hetken miettimään kuinka toteuttaa metodi toimimaan $O(n^3)$ ajassa, kunnes tajusin että sen pystyy toteuttamaan melko suoraan kertolaskun matemaattisen kaavan mukaan. Tämä helpotti selvästi työtä ja säästi paljon aikaa. Opin myös vihdoin sujuvasti taulukon läpikäynnin for-loopin avulla, joka on aikaisemmin tuottanut jonkin verran ongelmia mm. läpikäyntisuunnan kanssa.

Epäselväksi on jäänyt mm. miten voin varmistaa 100% testikattavuuden ja mitä tarkoittaa testidokumentaation ohje "ohjelman toiminnan empiirisen testauksen tulosten esittäminen graafisessa muodossa". Minua askarruttaa myös se, voinko tallentaa matriisit valmiin tietorakenteen avulla (esim hashtable) vai olisiko suotavampaa toteuttaa sellainen itse. Matriisien (pysyvämpi) tallennus ei ole harjoitustyön olennainen osio, mutta hyödyllinen sellainen.

Tällä viikolla ohjelma on edistynyt paremmin kuin olin olettanut saavani aikaiseksi, tosin matriisin tallentaminen syötteestä jäi toteuttamatta ja se olisi ollut kiva saada mukaan tähän palautukseen. Matriisilaskimen peruslaskutoiminnot on kumminkin toteutettu, joten projektin ydin alkaa olla vaaditusti kasassa. Tällä hetkellä matriisikertolasku on $O(n^3)$ ajassa, mutta tavoitteena on kehittää sitä tehokkaammaksi tulevina viikkoina.

Seuraavaksi toteutetaan matriisin transpoosi ja aletaan kehittämään LU- hajotelmaa. Viikon päästä olisi siis hyvä jos laskin osaisi määrittää ainakin determinantin ja ehkä käänteismatriisin. Käyttöliittymää kehitetään jos jää aikaa.

Viikko 2

Tällä viikolla opin toteuttamaan LU-hajoitelman, kuinka se toimii matriiseissa ja miksi sekä miten sitä voidaan hyödyntää mm. determinantin laskemisessa. Olen edistynyt myös koodin kirjoittamisessa suoraan matemaattisesta kaavasta ilman pseudokoodia.

Tältä viikolta useampi asia jäi hieman epäselväksi. Sain algoritmin Doolittle toimimaan niin että se tulostaa oikean

hajoitelman wikipedian esimerkkimatriisille, mutta en ole täysin varma siitä että se tulostaa oikeanlaisen kaikille matriiseille, enkä tiedä miten metodia voisi testata yleisenä tapauksena kaikille hajautusmatriiseille. Lisäksi muuttaessani matriisit liukulukumatriiseiksi, luokan Normimatriisi metodin skalaaritulo testauksessa on ongelmia. Testit heittävät "exepcted 0.0, but was -0.0". Kokeilin testimetodia assertArrayEquals deltan kanssa, mutta netbeans ei hyväksynyt delta kaksiulotteisen doubletaulukon kanssa. Deltan käyttäminen liukuluvun vertailussa kuitenkin onnistui. Muitakin pieniä epäselvyyksiä on, mutta tulen niistä sitten juttelemaan pajaan.

Ohjelman tärkeimmät metodit ja ydinalgoritmit ovat nyt toteutettuina, tosin matriisitulo on vielä tarkoitus muuttaa Strassen algoritmiksi. Ohjelma muuttaa nyt $n \times n$ -matriisit LU-matriiseiksi Doolittlen avulla ja erottaa hajautetut matriisit erillisiksi L- ja U -matriiseiksi. Hajotelmaa hyödyntävistä metodeista on toteutettu determinantti ja tämän lisäksi normimatriiseille on toteutettu matriisin transpoosi.

Seuraavaksi on tarkoituksena korjata/parantaa LU – hajoitelmaa ja varmistaa sen toimivuus kaikissa tapauksissa. Lisäksi tarkoituksena olisi toteuttaa käänteismatriisifunktio ja

ryhtyä suunnittelemaan Strassen algoritmia. Mikäli nämä asiat saadaan kuntoon, voidaan toteuttaa matriisin tallennus syötteestä.

Viikko 3

Tällä viikolla opin laskemaan käänteismatriisin lu-hajoitelman avulla ja algoritmin joka toteuttaa kyseisen laskuoperaation. Opin myös lisää testien suunnittelusta ja JUnitin ominaisuuksista (esim. deltan hyödyntäminen), sekä paikantamaan koodin ongelmakohtia debugin avulla (taito jossa olen yleensä surkea).

LU-hajotelmassa tai sitä hyödyntävissä metodeissa on jonkinlainen virhe (todennäköisesti U-matriisin kanssa), jonka vuoksi tulostamattomat arvot ovat okei kahta-kolmea solmua lukuunottamatta. Olen käynyt koodia läpi rivi riviltä eikä vikaa meinaa löytyä, joten tulen todennäköisesti pajaan ratkaisemaan ongelmaa. Lisäksi ohjelmassani on liian vähän testejä, mutta en millään keksi millaisia tapauksia voisin matriiseilla testata, joten ehdotuksia otetaan vastaan.

Ohjelman tuloksissa esiintyvien puutteiden vuoksi itse koodi on edistynyt tällä viikolla paljon vähemmän kuin oli tarkoitus,

ja tuntuu että en puutteiden vuoksi en pääse jatkamaan eteenpäin ennenkuin ne on korjattu. Lisäksi uusintakokeisiin valmistautuminen ja sairastelu ovat hidastaneet projektin edistymistä, mutta toivon että pääsen pian takaisin vauhtiin.

Seuraavaksi on luvassa ohjelman nykyisten toimintojen korjaamista toimiviksi, testien ja kestävyysden laatimista ja dokumenttien kirjoittelua. Mahdollisesti myös käyttöliittymä/Strassen algoritmi.

Viikko 4

Tällä viikolla sain korjattua LU-operaatioissa esiintyneet ongelmat, selkeytettyä koodia ja ohjelman rakennetta, sekä opin arvioimaan toisen kirjoittamaa koodia. Parin viikon doolittlen kanssa tuskailun jälkeen tajusin syöttää erään testimatriisin Wolfram alphaan ja selvisi että kyseinen matriisi olikin LPU-matriisi eikä LU-matriisi. Toisin sanoen Doolittle on toiminut kokoajan oikein, olin vain yrittänyt testata kelpaamattomilla syötteillä. Myöskään kertolasku $LU = A$ ei toiminut, koska apumuuttujan tyyppi oli virheellisesti int eikä double ja matriisit sisälsivät desimaalilukuja. Nyt LU-metodit toimivat hyvin käänteismatriisia lukuunottamatta, mutta tässä

vaiheessa se ei ole kovin suuri katastrofi.

Dokumenttien ohjeissa on muutama kohta jotka ovat jääneet hieman hämäräksi. Miten suorituskky ja O-analyysivertailu toteutetaan? Tarkoittaako tämä nyt sitä että ohjelman todellinen suoritusaika on sille annetun O-analyysiarvion mukainen? Tarkoittaako testausdokumentin "ohjelman toiminnan empiirisen testauksen tulosten esittäminen graafisessa muodossa." esim. palkkidiagrammia käsin tehdyistä testisuorituksista? Millainen sen kuuluisi olla? Haluaisin myös tietää, millaisia apumetodeja/laskufunktioita LU-luokkaan kannattaisi ehkä lisätä.

Tällä viikolla ohjelmaan ei ole kertynyt paljoa uutta sisältöä, sillä olen keskittynyt vanhojen metodien korjaamiseen, rakenteen ja nimien refactorointiin sekä testien kirjoittamiseen ja selkeyttämiseen. Viikonloppuna koitan saada myös jonkinlaista käyttöliittymää kasaan.

Seuraavassa palautuksessa pitäisi olla toimiva käyttöliittymä ja matriisien vastaanottaminen syötteenä, sekä toivottavasti toimiva käänteismatriisi. Mikäli se ei tunnu korjaantuvan, alan toteuttamaan Strassen algoritmia ja jätän käänteismatriisin korjauksen viimeiseksi asiaksi.