

Testausdokumentti

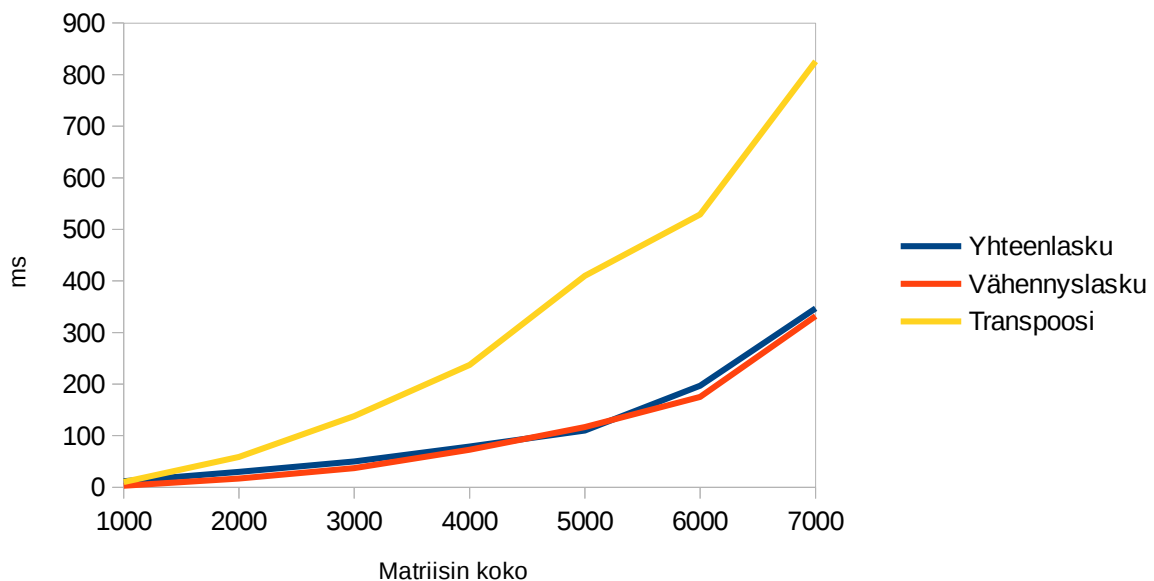
Testasin eri matriisioperaatioihin menevää aikaa eri kokoisilla syötteillä. Syöteinä olevat matriisit alustetaan satunnaisluvuilla välillä $[-100, 100]$. Käytetyt syötteet käyvät ilmi alla olevista kuvaajista. Testin voi toistaa kloonamalla projekti githubista, ja kirjoittamalla projektin juurikansiossa

```
mvn exec:java -Dexec.mainClass=com.mycompany.tiralabra_maven.performance_test.Test
```

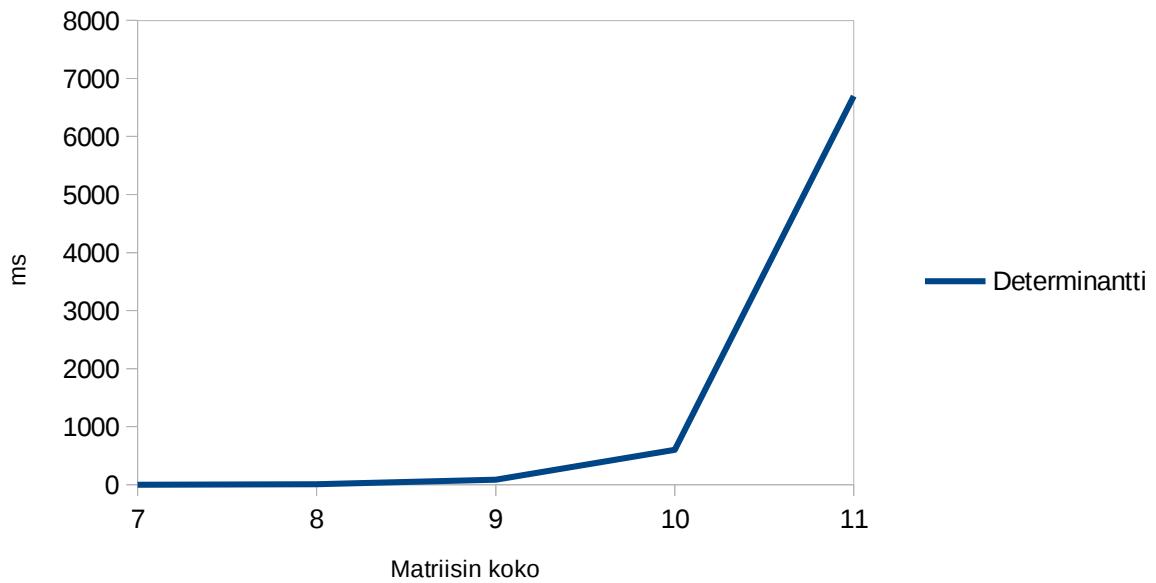
Saattaa olla, että joutuu lisäämään javan heap spacea, jotta testien ajaminen onnistuisi. Tämä onnistuu Linuxissa komennolla `export MAVEN_OPTS=-Xmx2g`.

Testit on toteutettu pakkauksessa `performance_test`. Satunnaisen vaihtelun vaikutuksen minimoimiseksi operaatioiden suoritusaajat lasketaan viiden suorituskerran keskiarvona. Käytin Strassenin algoritmia rekursion lopettamisen rajana 64:tä, eli kun syötteenä on 64×64 matriisi, tai sitä pienempi matriisi, niin kertominen suoritetaan perinteisellä matriisikertomisella. Optimaalinen raja saattaa riippua siitä, missä systeemissä ohjelma ajetaan. Raja säädetään `MatrixMath` luokan vakiolla `RECURSION_ENDPOINT`.

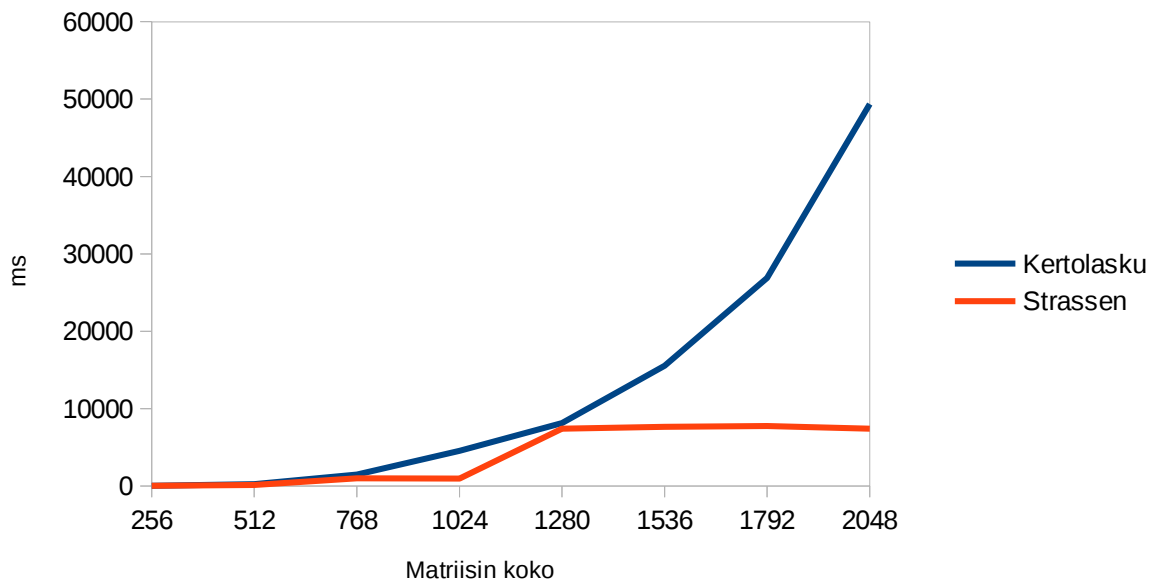
Testien tulokset tallennetaan `performanceTest.txt` nimiseen tekstitiedostoon projektin juureen.



Kuvaaja 1. Yhteen- ja vähennyslaskun sekä transpoosin suoritusaajat millisekunneissa eri arvoilla n , kun syötteenä on erikokoisia $n \times n$ matriiseja.



Kuvaaja 2. Determinantin suoritusajat eri arvoilla n , kun syötteenä on erikokoisia $n \times n$ matriiseja.



Kuvaaja 3. Perinteisen kertolaskun ja Strassenin algoritmin suoritusajat eri arvoilla n , kun syötteenä on erikokoisia $n \times n$ matriiseja.