

Käyttöohje

Lasse Lybeck, 013748498

Aineopintojen harjoitustyö: Tietorakenteet ja algoritmit

Ohjelman käynnistys

Valmis ohjelma löytyy kansioista Program. Ohjelma on jar-muodossa ja sen toimii komentorivin kautta. Ohjelman saa käynnistettyä komentoriviltä komennoilla

```
cd <kansion Program sijainti>
java -jar Matrix.jar
```

Päävalikko

Kun ohjelma on käynnistetty, aukeaa ohjelman päävalikko. Päävalikosta saa valittua minkä aliohjelman haluaa ajaa. Tämä onnistuu kirjoittamalla sen ohjelman numeron, jonka haluaa käynnistää. Ohjelman saa suljettua numerolla 0.

Matriisilaskin

Matriisilaskimessa voi matriiseilla suorittaa niiden yleisimpiä operaatioita. Laskimessa voi lisäksi tallentaa muuttujiin matriiseja ja käyttää näitä laskennassa. Ohjelman käynnistyessä muuttujat ovat tyhjennetty, ja ne myös häviävät muistista ohjelman suljettua. Kesken ohjelman voi muuttujat tyhjentää komennolla 'clear', ja kullakin hetkellä muistissa olevat muuttujat voi tarkistaa komennolla 'vars'. Ohjelmasta voi poistua komennoilla 'quit' ja 'exit'.

Muuttujiin tallentaminen tapahtuu kirjoittamalla komentoriville muuttujan nimeksi halutun merkkijonon ja tämän jälkeen '=' ja halutun komennon. Esimerkiksi komento

```
mat2 = rand(3)
```

tallentaa muuttujaan 'mat2' satunnaisen 3x3 matriisin. Muuttujan nimi saa sisältää ainoastaan kirjaimia ja numeroita, ja sen tulee alkaa kirjaimella. Jos komentoa ei tallenna mihinkään muuttujaan, tulee se muistiin tilapäiseen *ans* muuttujaan. Muuttujat ovat tyypiltään matriiseja (Matrix), lukuja (Number) tai LU-hajotelmia (LU). Vain kahden saman tyyppisen muuttujan välillä voi toteuttaa laskutoimituksia.

Komennot on toteutettu funktiomuodossa. Tämä tarkoittaa, että esimerkiksi kahden matriisin *m1* ja *m2* summan saa komennolla

```
add(m1, m2)
```

Operaattoreita (+, *, jne.) ei voi käyttää laskutoimituksiin. Myöskään useampaa komentoa ei voi tehdä samalla rivillä. Jos esimerkiksi halutaan kolmen matriisin *m1*, *m2* ja *m3* summa, tulee tämä toteuttaa esimerkiksi seuraavasti:

```
temp = add(m1, m2)
add(temp, m3)
```

Joissakin komennoissa argumenttina voi antaa luvun (esimerkiksi skalaarilla kertominen). Tällöin luvun tulee olla Javan double-arvon hyväksymässä muodossa. Luku voi siis alkaa + tai - merkillä (mutta ei tarvitse), jonka jälkeen seuraa numeroita ja desimaaliosia erotettuna pisteellä. Loppuun voi

lisätä vielä kymmenpotenssi osan erotettuna e tai E symbolilla. Esimerkiksi $-3.14e+12$ on hyväksyttävä luku. Lisäksi luvun voi kirjoittaa murtolukuna, eli kahden hyväksyttävän luvun jakona. Esimerkiksi $-3/13$ on siis hyväksyttävä luku.

Komennon jälkeen tulostuksen voi poistaa komennon loppuun sijoitetulla puolipisteellä (;). Jos puolipistettä ei lisää, tulostuu komennon tulos komentoriville.

Matriisilaskimen operaatiot löytyvät dokumentin lopusta.

Tiedoston ajaja

Tiedoston ajajalla (file runner) voi ajaa tiedostoon kootut komennot kerralla. Komennot toimivat samalla tavalla kuin matriisilaskimessa.

Ajettavat tiedostot tulee tallentaa kansioon 'files', joka sijaitsee samassa kansiossa ohjelman kanssa. Tiedostojen tulee olla .txt muodossa, ja jokaisella riville tulee olla korkeintaan yksi komento. Rivit, jotka ovat tyhjiä tai alkavat #-merkillä sivuutetaan. Rivin saa siis kommentoitua pois aloittamalla rivin #-merkillä.

Komennolla 'files' tulostaa ohjelma listan ajettavista tiedostoista, eli niistä tiedostoista, jotka sijaitsevat files-kansion jossakin alikansiossa. Tiedoston voi ajaa kirjoittamalla tiedoston polun (kuten 'files'-komento sen antaa) komentoriville ja painamalla enter-näppäintä.

Yhtälön ratkaisija

Yhtälön ratkaisijassa (equation solver) voi ratkaista $N \times N$ lineaarisen yhtälöryhmän, eli yhtälön $Ax=b$, missä A on $N \times N$ matriisi ja b $N \times 1$ pystyvektori.

Ohjelma kysyy ensin lukua N , joka siis on yhtälöiden ja tuntemattomien määrä. Tämän jälkeen ohjelma pyytää syöttämään rivi kerrallaan yhtälön vakiot. Nämä syötetään numeroina ja erotetaan välilyönneillä tai pilkuilla (.). Yhdellä rivillä tulee olla $N+1$ lukua, joista N ensimmäistä ovat muuttujien kertoimet ja viimeinen on oikean puolen vakio.

Esimerkiksi, jos halutaan ratkaista yhtälöryhmä

$$3x + 2y - z = 1$$

$$2x - 2y + 4z = -2$$

$$-x + y/2 - z = 0$$

menee ohjelman toiminta seuraavasti:

Input N, zero or empty line exits:

```
>> 3
```

Input coefficients in equation 1 separated with comma or space.

```
>> 3, 2, -1, 1
```

Input coefficients in equation 2 separated with comma or space.

```
>> 2, -2, 4, -2
```

Input coefficients in equation 3 separated with comma or space.

```
>> -1, .5, -1, 0
```

Equation $Ax = b$:

$A =$

| | | |
|---------|---------|---------|
| 3,0000 | 2,0000 | -1,0000 |
| 2,0000 | -2,0000 | 4,0000 |
| -1,0000 | 0,5000 | -1,0000 |

$b =$

| |
|---------|
| 1,0000 |
| -2,0000 |
| 0,0000 |

$x =$

| |
|---------|
| 1,0000 |
| -2,0000 |
| -2,0000 |

Vastaukseksi saadaan siis

$x = 1, y = -2, z = -2.$

Matriisilaskimen operaatiot

Parse(expr)

Komennolla voi syöttää matriisin laskimeen. `expr` on merkkijono, joka sisältää matriisin seuraavassa muodossa: Kaikki alkiot on syötetty numeroina (murtoluvut ovat hyväksytyjä), saman rivin alkiot on erotettu toisistaan pilkuilla tai välilyönneillä, ja rivit toisistaan puolipisteillä. Jokaisella rivillä tulee olla yhtä monta alkia.

Esimerkki:

```
>> a = parse(1, 2, 3; 5, 4, 3; 2/3, -5 3.14)
```

$a =$

| | | |
|--------|---------|--------|
| 1.0000 | 2.0000 | 3.0000 |
| 5.0000 | 4.0000 | 3.0000 |
| 0.6667 | -5.0000 | 3.1400 |

zeros(n), zeros(m,p)

Palauttaa $n \times n$ tai $m \times p$ matriisin, jonka kaikki alkiot ovat nollia.

ones(n), ones(m,p)

Palauttaa $n \times n$ tai $m \times p$ matriisin, jonka kaikki alkiot ovat ykkösiä.

eye(n), eye(m,p)

Palauttaa $n \times n$ tai $m \times p$ yksikkömatriisin.

rand(n), rand(m,p)

Palauttaa $n \times n$ tai $m \times p$ matriisin, jonka alkiot ovat satunnaisia väliltä $[0, 1[$.

randi(imax, n), randi(imax, m, p), randi(imin, imax, m, p)

Palauttaa $n \times n$ tai $m \times p$ matriisin, jonka alkiot ovat satunnaisia kokonaislukuja. Kahdessa ensimmäisessä komennossa luvut ovat väliltä $[0, imax]$, kolmannessa väliltä $[imin, imax]$.

add(m1, m2)

Palauttaa matriisien $m1$ ja $m2$ summan.

sub(m1, m2)

Palauttaa matriisien $m1$ ja $m2$ erotuksen ($m1 - m2$).

mul(m1, m2)

Palauttaa matriisien $m1$ ja $m2$ tulon. Tulo lasketaan käyttämällä naiivia matriisien kertolaskun algoritmia, ja on siis identtinen komennon `mulN` kanssa.

mulN(m1, m2)

Palauttaa matriisien $m1$ ja $m2$ tulon. Tulo lasketaan käyttämällä naiivia matriisien kertolaskun algoritmia.

mulS(m1, m2)

Palauttaa matriisien $m1$ ja $m2$ tulon käyttäen Strassenin algoritmia.

pow(m, e)

Korottaa matriisin m korotettuna potenssiin e (m^e). Exponentin tulee olla kokonaisluku ja vähintään -1 .

transpose(m)

Palauttaa matriisin m transpoosin.

det(m)

Palauttaa matriisin m determinantin.

inv(m)

Palauttaa matriisin m käänteismatriisin, jos sellainen on olemassa. Jos käänteismatriisia ei ole olemassa funktio ilmoittaa tästä varoituksella.

LU(m)

Palauttaa matriisin m LU-hajotelman. Hajotelman matriisit saa käyttämällä kutsuja `getL` ja `getU`.

getL(lu)

Palauttaa LU-hajotelman lu matriisin L.

getU(lu)

Palauttaa LU-hajotelman lu matriisin U.