Lab 4: Strassen method for matrix multiplication

Professor: Ronaldo Menezes

TA: Ivan Bogun

Department of Computer Science
Florida Institute of Technology

September 15, 2014

1 Problem statement

Standard way for matrix multiplication takes $O(n^3)$, which is pretty expensive. In 1969 Strassen was the first to show that matrix multiplication is in fact $o(n^3)$ by presenting a method whose asymptotic complexity is $O(n^{\log 7})$. In this lab you are asked to implement it using recursion.

1.1 Strassen method

Assume we are to calculate the product $C = A \cdot B$ where $A, B \in \mathbf{R}^{2^n \times 2^n}$ for some $n > 0$, then we can rewrite the multiplication as a product of blocks of size $2^{n-1} \times 2^{n-1}$

$$\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \cdot \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

Step 1 Create matrices $A_{11}, ..., B_{11}, ..., C_{11}$. Each matrix should have a size of 2^{n-1}.

Step 2 Calculate matrices $S_1, ..., S_{10}$ which are linear combinations of matrices from the step 1. Matrices $S_1, ..., S_{10}$ are given by

$$\begin{aligned}
S_1 &= B_{12} - B_{22}, \\
S_2 &= A_{11} + A_{12}, \\
S_3 &= A_{21} + A_{22}, \\
S_4 &= B_{21} - B_{11}, \\
S_5 &= A_{11} + A_{22}, \\
S_6 &= B_{11} + B_{22}, \\
S_7 &= A_{12} - A_{22}, \\
S_8 &= B_{21} + B_{22}, \\
S_9 &= A_{11} - A_{21}, \\
S_{10} &= B_{11} + B_{12}.
\end{aligned}$$

Step 3 Use matrices calculated at step 2 to recursively calculate matrix products $P_1, ..., P_7$ defined as:

$$\begin{aligned}
P_1 &= A_{11} \cdot S_1, \\
P_2 &= S_2 \cdot B_{22}, \\
P_3 &= S_3 \cdot B_{11}, \\
P_4 &= A_{22} \cdot S_4, \\
P_5 &= S_5 \cdot S_6, \\
P_6 &= S_7 \cdot S_8, \\
P_7 &= S_9 \cdot S_{10}.
\end{aligned}$$

Step 4 Compute matrices $C_{11}, C_{12}, C_{21}, C_{22}$ from matrices P_i:

$$\begin{aligned}
C_{11} &= P_5 + P_4 - P_2 + P_6, \\
C_{12} &= P_1 + P_2, \\
C_{21} &= P_3 + P_4, \\
C_{22} &= P_5 + P_1 - P_3 - P_7.
\end{aligned}$$

2 Implementation

Implement the class *Matrix.java* [1].

```
public class Matrix {

        private double[][] elements;                    // elements of the matrix
        private int n;                                                  // size


        public Matrix(double[][] elements_) {
                // constructor
        }

        public Matrix multiplyStrassen(Matrix b) {
        // implement Strassen method for matrix multiplication. This function
            should be recursive.
        }

        public Matrix multiply(Matrix b) {
        //      implement regular matrix multiplication method (hint: you might
            want to use it for testing)
        }

        public boolean equals(Matrix b) {
        // check if matrices are equal. Compare elements up to certain
            precision, say 1e-6, e.g.
        // abs(this.elements[i][j]-b.elements[i][j])<1e-6
        }

        public String toString() {
        // return string representation of the matrix
        }

        public Matrix add(Matrix b) {
        // addition
```

[1] You can copy code for this file from: Matrix.java

```
    }

    public Matrix subtract(Matrix b) {
    // subtraction
    }

}
```

3 Sample input-output

Create the file *Driver.java* [2] whose modified version will be used for testing. You can always assume that the input will be a square matrices whose size is a factor of 2.

3.1 Input

```
public class Driver {

    public static void main(String[] args) {

        // create some double
        double[][] el_1={{1,3,4,5},{2,4,3,5},{1,3,4,5},{2,4,3,5}};
        double[][] el_2={{1,3,4,5},{2,4,1,5},{1,5,4,5},{2,4,-3,3}};

        Matrix m1 = new Matrix(el_1);
        Matrix m2 = new Matrix(el_2);

        System.out.println(m1);
        Matrix productStrassen=m1.multiplyStrassen(m2);
        Matrix productRegular=m1.multiply(m2);

        System.out.println("Are matrices the same?
            "+productStrassen.equals(productRegular));
        System.out.println(productStrassen);

    }

}
```

3.2 Output

```
Matrix of the size [4,4]
1.0 3.0 4.0 5.0
2.0 4.0 3.0 5.0
```

[2] Driver.java

```
1.0 3.0 4.0 5.0
2.0 4.0 3.0 5.0

Are matrices the same? true
Matrix of the size [4,4]
21.0 55.0 8.0 55.0
23.0 57.0 9.0 60.0
21.0 55.0 8.0 55.0
23.0 57.0 9.0 60.0
```

4 Grade breakdown

basis	grade
Implementation	(60)
Strassen multiplication	40
regular multiplication	5
other	15
Comments	(20)
General	10
Javadocs	10
Overall	(20)
Compiled	5
Style	5
Runtime	10
Total	100