COMS 310

Fall 2017

Hart

Program I

Let C = AB where A, B and C are NxN matrices. For many years it was assumed that an algorithm to compute C was *O*(N3). In the late ‘60s Strassen showed that this could be improved upon, albeit not by much. The basic idea of Strassen’s algorithm is to divide each matrix into four quadrants like so:

 = 

It can be shown that

C1,1 = A1,1B1,1 + A1,2B2,1

C1,2 = A1,1B1,2 + A1,2B2,2

C2,1 = A2,1B1,1 + A2,2B2,1

C2,2 = A2,1B1,2 + A2,2B2,2

E.g, to perform the multiplication AB

AB = 

We define the following eight N/2 by N/2 matrices

A1,1 =  A1,2 =  B1,1 =  B1,2 = 

A2,1 =  A2,2 =  B2,1 =  B2,2 = 

We could perform 8 N/2 by N/2 matrix multiplications and four N/2 by N/2 matrix additions. However, doing so still results in O(N3). Strassen used a different strategy which uses only seven recursive calls. There are seven multiplications

M1 = (A1,1 + A2,2)( B1,1 + B2,2)

M2 = (A2,1 + A2,2) B1,1

M3 = A1,1 ( B1,2 – B2,2)

M4 = A2,2 (B2,1  – B1,1)

M5 = (A1,1 + A1,2) B2,2

M6 = (A2,1 – A1,1) )( B1,1 + B1,2)

M7 = (A1,2  – A2,2)(B2,1 + B2,2)

Once these multiplications are performed the final answer can be obtained with eight more additions.

C1,1  = M1 + M4 – M5 + M7

C1,2  = M3 + M5

C2,1  = M2 + M4

C2,2  = M1 – M2 + M3 + M6

It can be verified that this approach gives O(N2.807).

Your job in this programming assignment is to implement the above sketched algorithm as a divide and conquer algorithm. I.e., you are to implement Strassen’s algorithm recursively. Your algorithm should thus have the following signature:

public static int[][] Strassen(int[][] a, int[][] b)

Note that you will want to implement other methods as well and that you may assume that a and b are both of size n where n is a power of 2.

You should test your code with a variety of matrices.

This assignment is due by 11:59 p.m. on Monday September 25th 2017.