Github code - https://github.com/2minuteai/a1

Performed Matrix Multiplication using Normal, Recursive, Strassen.

Did average runs - **3**

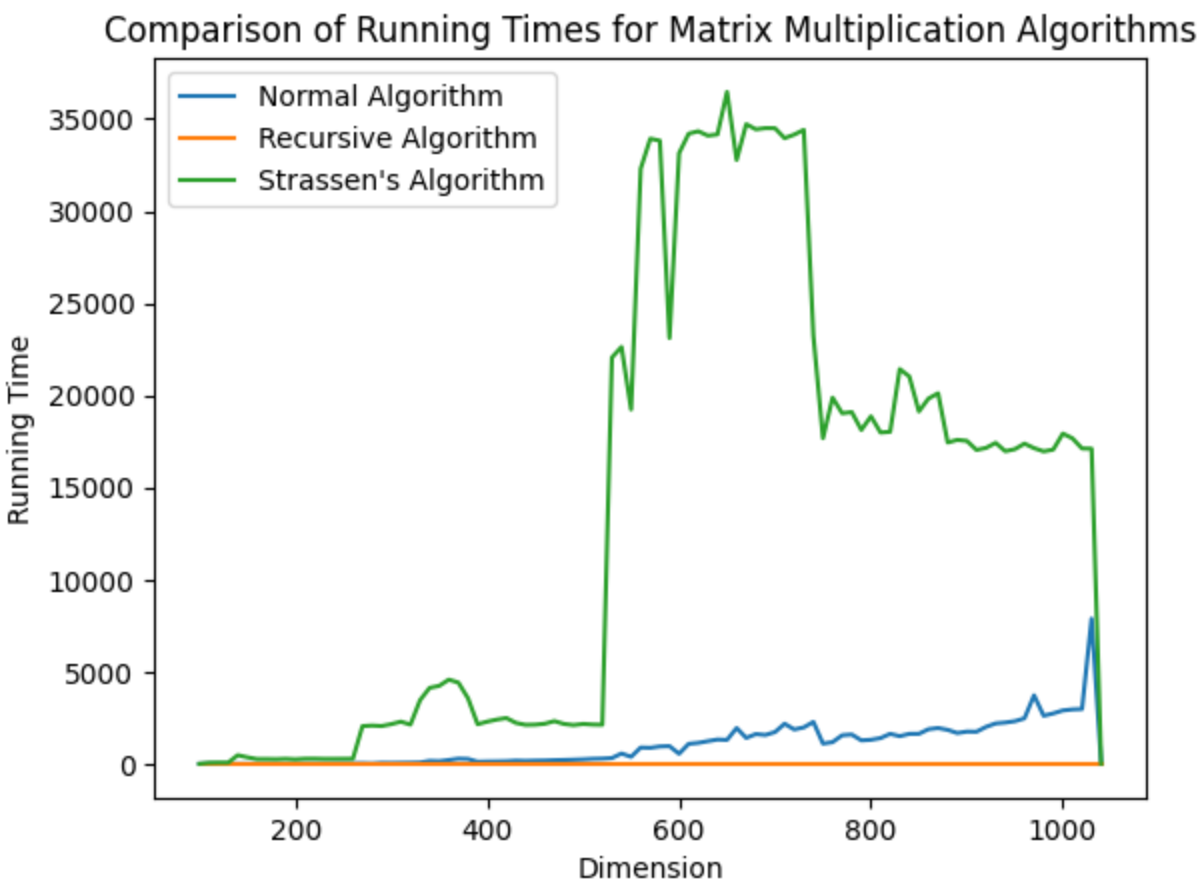
The graphs were obtained using **python** code using the csv generated from the java code.

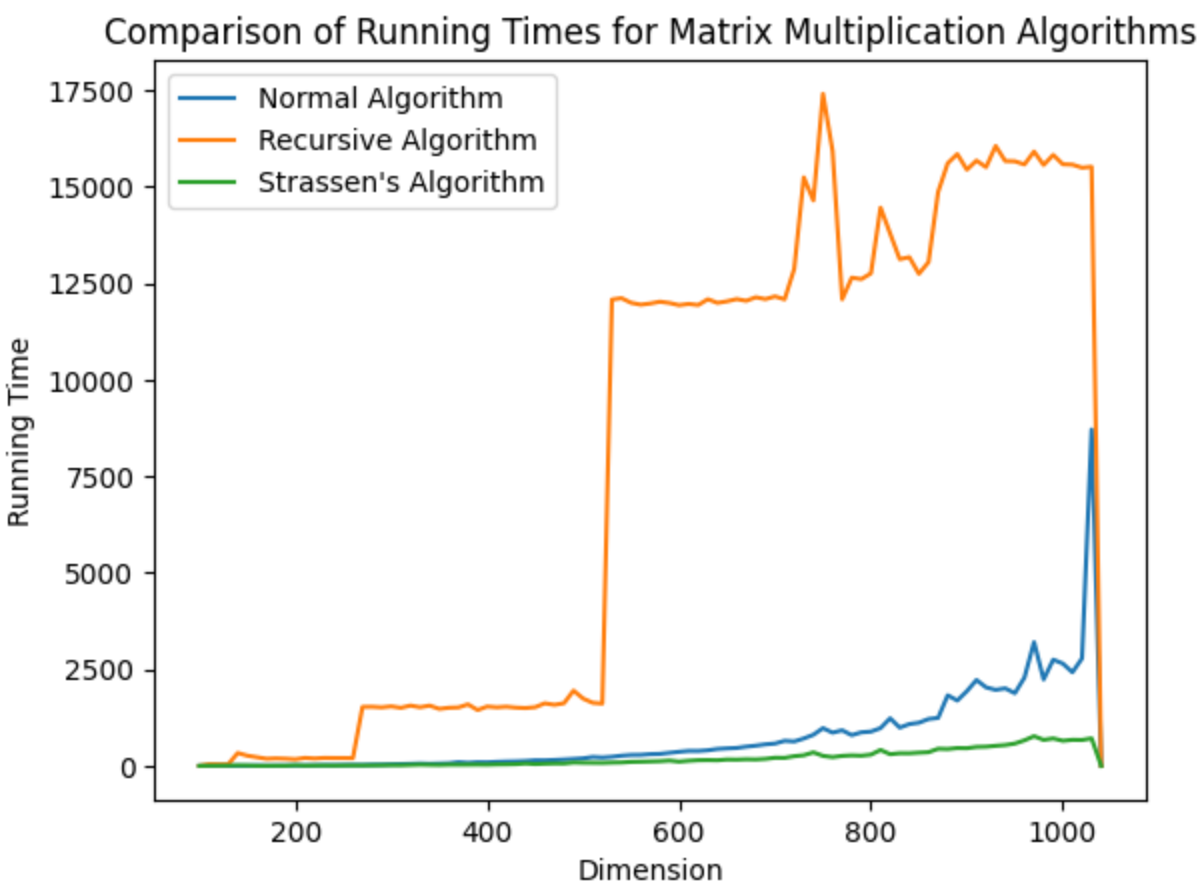
First I generated the csv file using the average times in milliseconds for each of the three algorithms.
Using the below code -

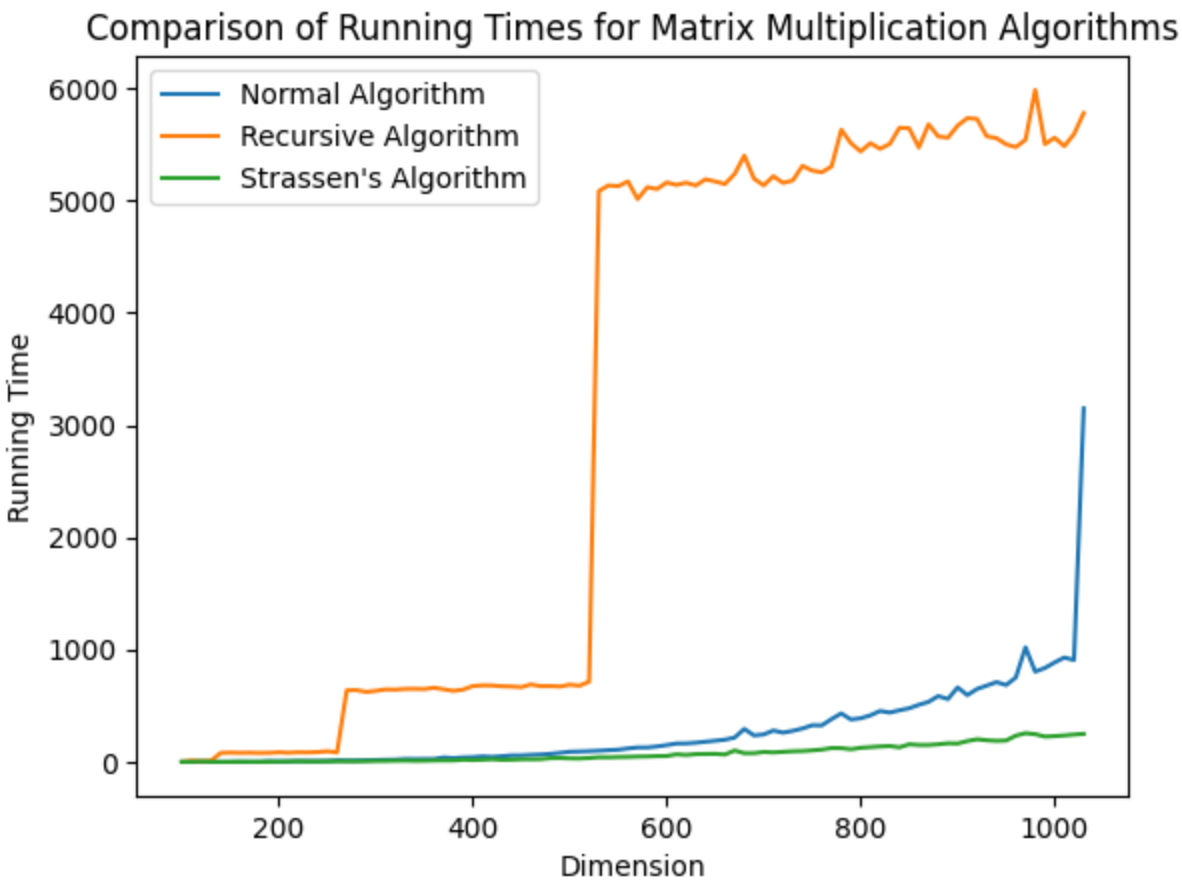
```
normalTimes[(z - start) / 10] = normalTime / runs;
recursiveTimes[(z - start) / 10] = recursiveTime / runs;
strassenTimes[(z - start) / 10] = strassenTime / runs;
writer.append("Dimension,Normal Time,Recursive Time,Strassen Time\n");
        for (int i = 0; i < normalTimes.length; i++) {
           writer.append(String.valueOf(i * 10 + 100));
```

The dimensions range from 100 to 1024 → beyond that takes more than an hour.
The matrices were randomly generated at each dimension.
The graph using multiple tries were different for some reason, so I changed the random integer from 0 to 5 instead of 0 to 20 and the running time came significantly down.

The graphs attached in order of decreasing running time after modifications.


Comparison of Running Times for Matrix Multiplication Algorithms
Normal Algorithm
Recursive Algorithm
Strassen's Algorithm
Running Time
Dimension
0
5000
10000
15000
20000
25000
30000
35000
200
400
600
800
1000


Comparison of Running Times for Matrix Multiplication Algorithms
Normal Algorithm
Recursive Algorithm
Strassen's Algorithm
Running Time
Dimension
17500
15000
12500
10000
7500
5000
2500
0
200
400
600
800
1000


Comparison of Running Times for Matrix Multiplication Algorithms
Normal Algorithm
Recursive Algorithm
Strassen's Algorithm
Running Time
Dimension
0
1000
2000
3000
4000
5000
6000
200
400
600
800
1000