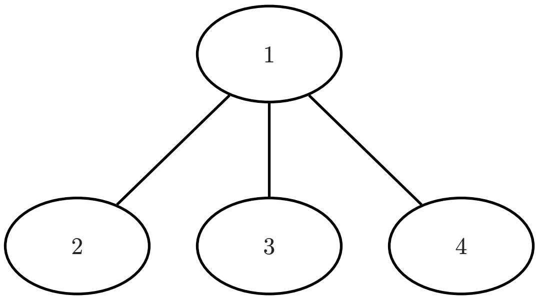
Introduction to Algorithm Engineering

Homework-2

Moida Praneeth Jain, 2022101093

Question 1



Question 2

Square Matrices

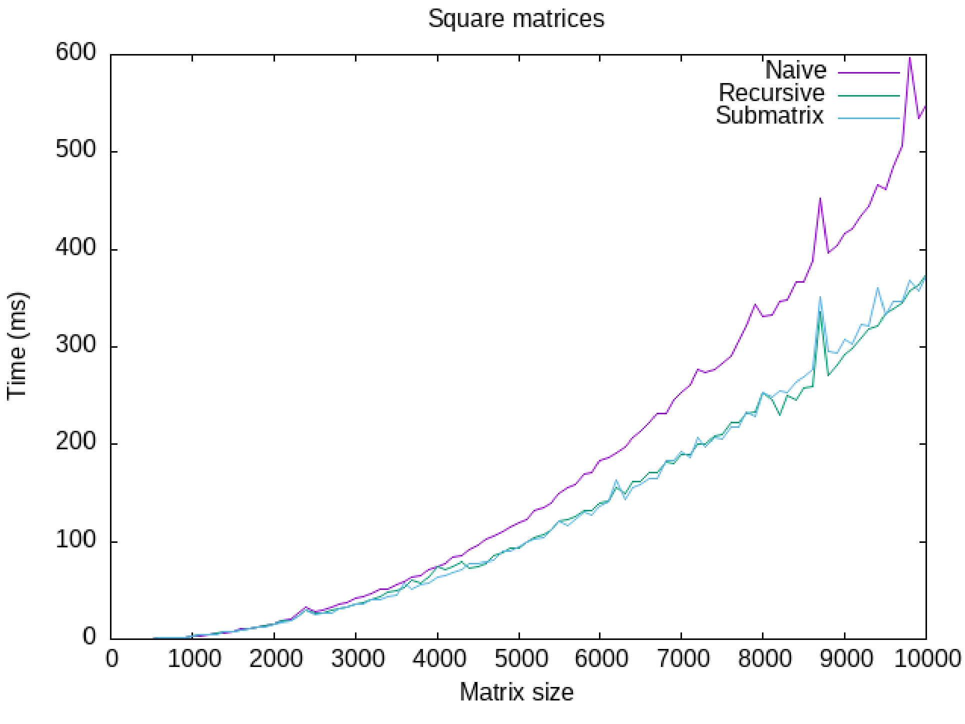


Figure 1: Variation in transpose time with size of square matrices

First, let us consider the case of square matrices. As we can observe from Figure 1, the trend is quadratic, as was noted in Homework 1. In the naive case, we get $O(N^2)$, and for the recursive and submatrix case, we get $O\left(\frac{N^2}{B}\right)$

Row Matrices

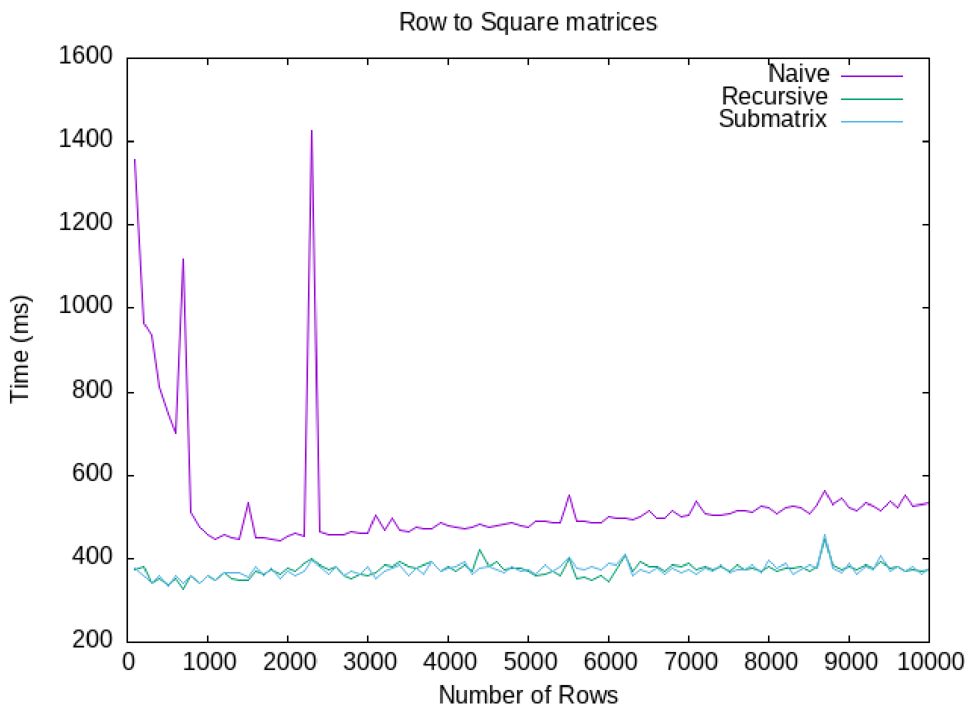


Figure 2: Variation in transpose time with rows of matrix. (1e8 matrix entries)

We fix the total number of elements in the matrix to `1e8`, and increase the number of rows to `1e4` while decreasing the number of columns from `1e8`, i.e, the plot represents the transpose time as the graph transitions from a row matrix to a square matrix.

From Figure 2 we observe that the times are approximately constant. This is because in these matrices, each row has more elements than the cache can fit, thus the cache hit/miss rate remains constant throughout.

Column Matrices

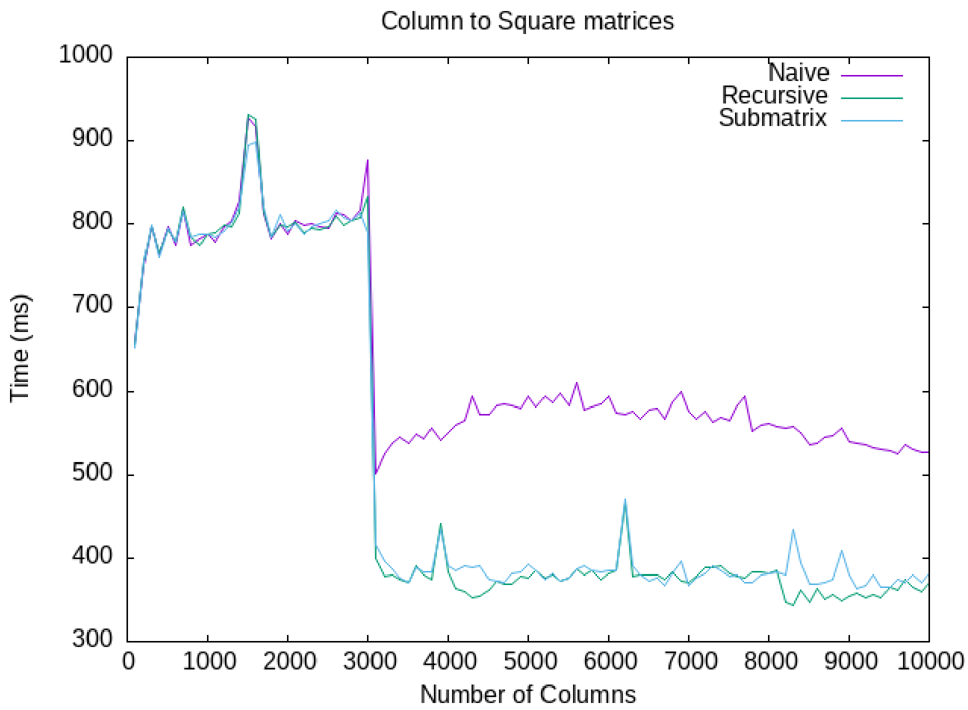


Figure 3: Variation in transpose time with columns of matrix. (1e8 matrix entries)

We fix the total number of elements in the matrix to `1e8`, and increase the number of columns to `1e4` while decreasing the number of rows from `1e8`, i.e, the plot represents the transpose time as the graph transitions from a column matrix to a square matrix.

The key point to note from Figure 3 is that these times are much higher than the row matrices. This is because column matrices are much worse for cache access than row matrices. At around `3000` columns, the times go back down. This is where the number of elements in each row (the number of columns) is large enough for the cache to get filled up.