Shawn Zhu  
ECE759  
Assignment 2

Task 1c: Scaling Analysis

A graph with numbers and lines

Description automatically generated

Task 3F:

In mmul1, the loops iterate in the order (i, j, k). This results in relatively cache-friendly access when traversing matrix A, since it is stored in row-major order. The problem here is that matrix B is accessed column wise because every step of k results in a n size jump.

In mmul2, j is now the inner most iterator which means that matrix B is accessed more efficiently in row-major order as well. This means that both matrices are accessed in row-major order leading to the most efficient performance of the 4 options.

In mmul3, the loops iterate the least efficiently. The outermost loop now iterates over j, meaning the algorithm effectively computes column-by-column for the output matrix C. This can lead to inefficient cache usage for both matrices A and B, as neither is traversed in an order that matches the way the matrices are stored in memory.

In mmul4, the difference is the way matrices are stored in memory. So the order of traversal is the same as in mmul1, resulting in very similar times.