Project 2A

Tianyi Mo, 875556

In order to achieve speedup using multiple cores, it is important to analysis the dependency of elements in the matrix. Starting for the upper-lest corner, each diagonal is depending on its previous diagonal.

The first method I tried is to calculate each diagonal parallelly. Although it is theoretically faster, traverse the matrix diagonally can be extremely expensive due to the memory access and cache miss (cache line size is 64 bytes but only 4 bytes is used each time).

Then I use the method to calculate diagonal of blocks parallelly, and one thread computes values sequentially within each block. This method increase the granularity and can improve the cache performance because the linear access of several continuous block of memory. In addition, it creates significantly less threads compare to the previous approach thus suffer less overhead for creating and destroying the threads. The height or width of each block is equal to the height or width of matrix divided by maximum number of thread available. Then there is maximum 24 blocks to compute parallelly and we have 24 cores, this can avoid schedule blocks in one loop.