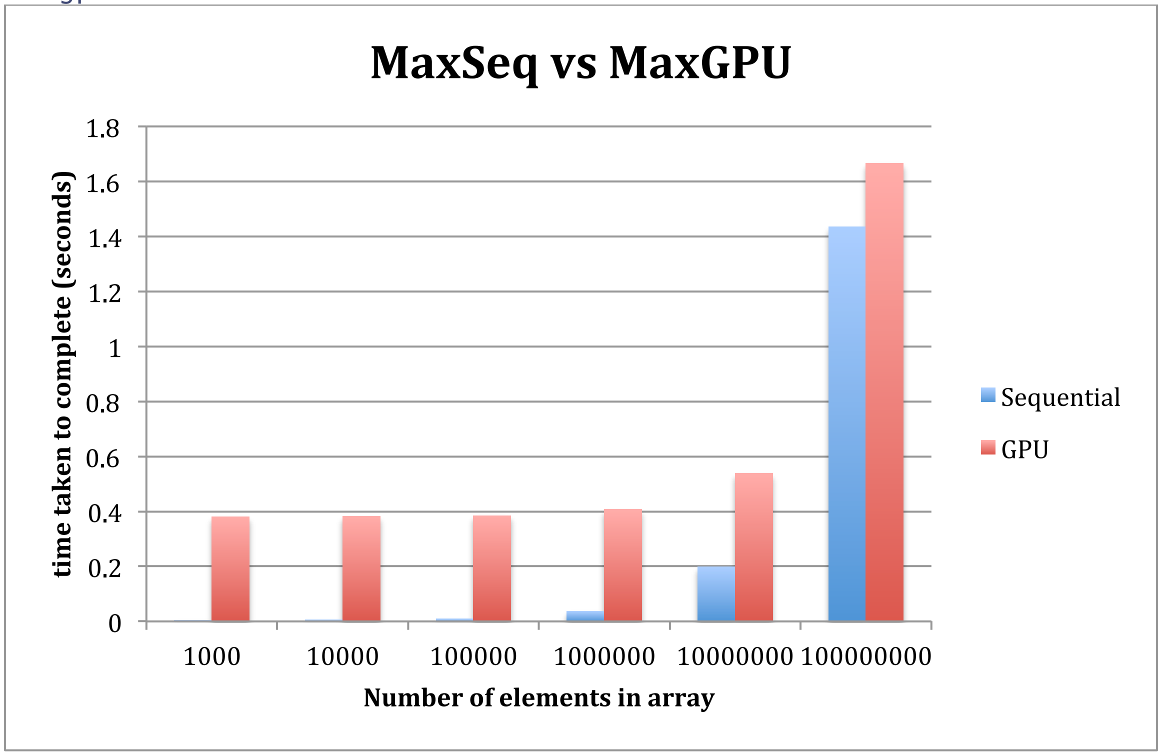
Carol He
Parallel Computing
Lab 3 Analysis

1. I chose to have 512 threads per block, because it is a multiple of 32, which is the standardized warp size. I chose to have 512 threads instead of 1024 because in order for my code to work for X = 1000, the number of threads per block had to be less than that number.
   To choose grid size, I did (size + THREADS_PER_BLOCK - 1)/THREADS_PER_BLOCK. I did this to compensate for extra elements that do not fit within a block.
2. On cuda1, compile with `nvcc -arch=compute_35 -code=sm_35 maxgpu.cu`
3. 

4. The speedup is not noticeable until 100 million, where it is nearly observed. The lack of speedup is probably due to contention of resources on the servers. It is also not observable because the number of elements is too small and cudaMalloc and cudaMemcpy are very expensive. With smaller X's, there is less computation, so it does not outweigh the cost of these communications yet. We would probably see speedup with larger X's.
   Also, note that the relative difference between the sequential time and GPU time decreases exponentially. This is because as the data size grows, there is more computation for each block to do relative to the amount of communication needed. Though no speedup can be observed in this dataset, overall, maxSeq is still getting much slower each time in comparison to the rate at which maxGPU is getting slower.