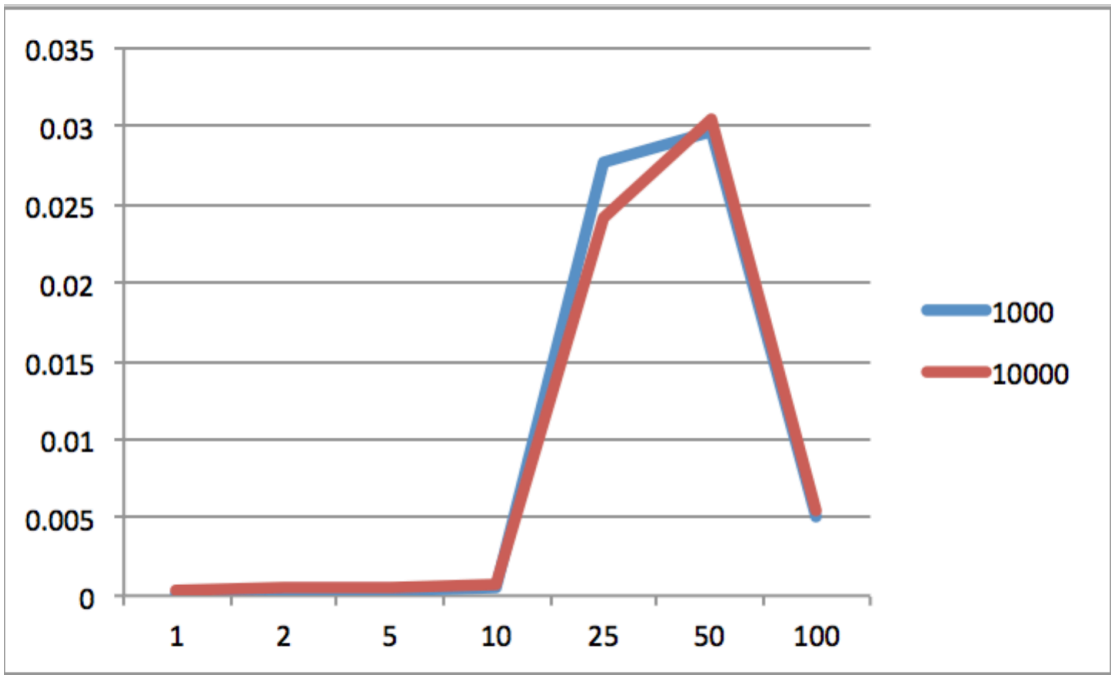
Parallel Computing Lab 2
Name: Elaine Ang
NetID: ra1695

Timing data:

	1	2	5	10	25	50	100
1000	0. 000042	0. 000107	0. 000258	0. 000434	0. 027619	0. 029701	0. 005067
10000	0. 000347	0. 000557	0. 000508	0. 000753	0. 024094	0. 030413	0. 005396

Time used comparison:



Explanation:
There is a serious load imbalance in the main for-loop. Since I labeled all the crossed out value as 0, and not doing all the operations if an element is 0, I put more work on the first several threads (that has a lot of value to cross out), but less work for the following ones (that probably have no values for crossing out). However, since there is an implicit barrier at the end of omp parallel for, the for-loop finishes when the slowest thread finish.
Besides, those threads are accessing and updating non-contiguous memory, which leads to lots of cache misses. It takes relatively a long to go back to memory and bring back a consecutive block to cache.

There should be a speed up in the first for-loop where I initialize the array. However, the slow down of the second for-loop got serious and eventually slows down the whole process.