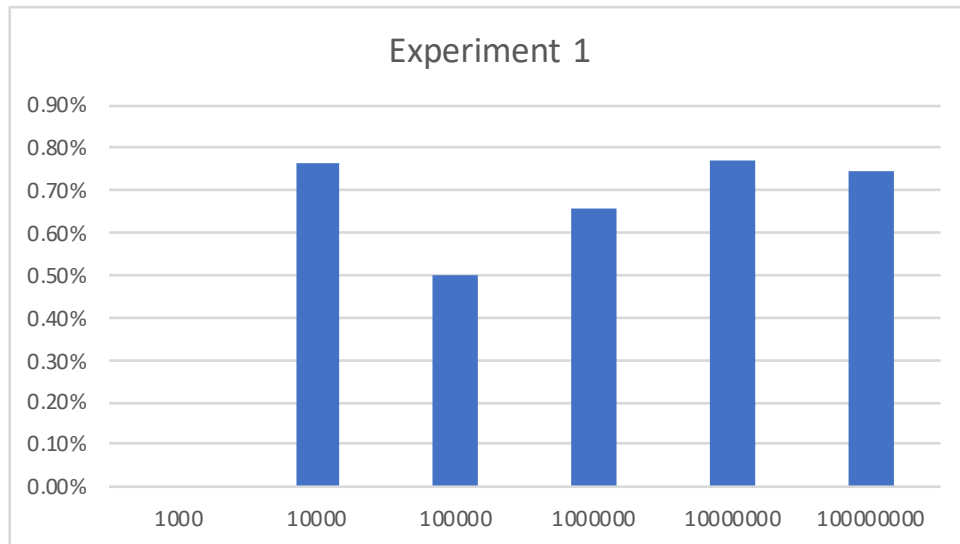


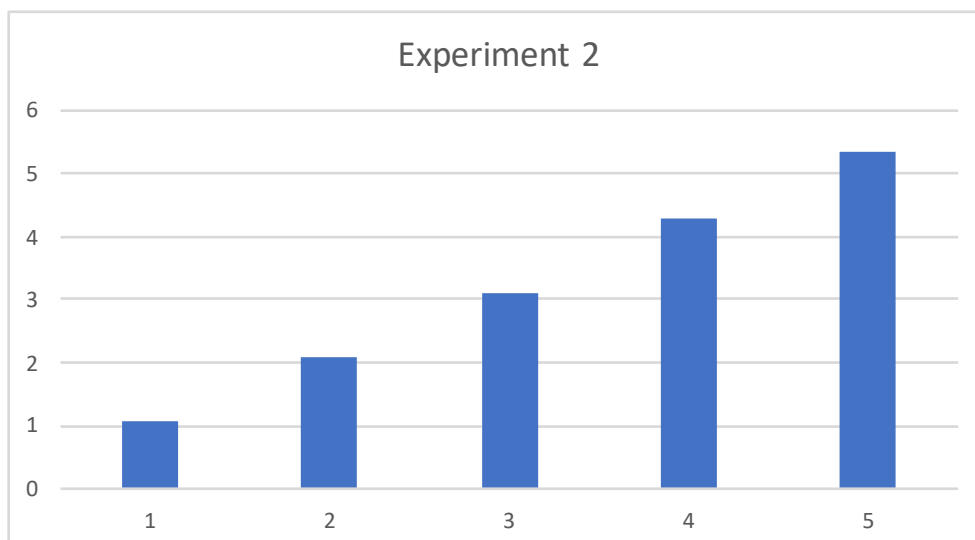
### Experiment 1

10 threads	1000	10000	100000	1000000	10000000	100000000
para	0.000056	0.00009167	0.00036133	0.00308633	0.03518733	0.34374467
real	0.008	0.012	0.0723333	0.47133333	4.58	46.0623333
percentage	0.00%	0.76%	0.50%	0.65%	0.77%	0.75%



### Experiment 2

1000000num	para	speedup
sequential	0.02420733	1
1	0.022316	1.0847522
2	0.01170133	2.0687674
3	0.00781233	3.09860567
4	0.00566067	4.27640721
5	0.004518	5.35797477



### Experiment 3

50 bins

1000	para	speedup	effeciency
1	0.000036	1	1
2	0.000035	1.02857143	0.51428571
3	0.000034	1.05882353	0.35294118
4	0.000031	1.16129032	0.29032258
5	0.000044	0.81818182	0.16363636

10000

para	speedup	effeciency
1	1	1
2	1.71253623	0.85626812
3	2.1549193	0.71830643
4	2.65539326	0.66384831
5	2.88207317	0.57641463

100000

para	speedup	effeciency
1	1	1
2	1.88473184	0.94236592
3	2.8382889	0.9460963
4	3.58309683	0.89577421
5	4.53896777	0.90779355

1000000

para	speedup	effeciency
1	1	1
2	1.90713363	0.95356682
3	2.85651016	0.95217005
4	3.94228952	0.98557238
5	4.9393537	0.98787074

10000000

para	speedup	effiecincy
1	1	1
2	1.23779956	0.61889978
3	1.62899298	0.54299766
4	1.83991364	0.45997841
5	2.58926328	0.51785266

	1000	10000	100000	1000000	10000000
1	1	1	1	1	1
2	0.514285714	0.85626812	0.94236592	0.95356682	0.61889978
3	0.352941176	0.71830643	0.9460963	0.95217005	0.54299766
4	0.290322581	0.66384831	0.89577421	0.98557238	0.45997841
5	0.163636364	0.57641463	0.90779355	0.98787074	0.51785266

## Analysis

1. According to experiment 1, the fraction of the sequential part from the overall execution time first decreases, then increase, then decrease as the problem size increases. But the change is very small.
2. As the threads increases, the speedup increases in lab 2. For lab1, it's not very obvious, even has slowdown. To justify the lab2, refer the data and graph in experiment 2. To justify lab1, I attach the data below.
3. As the problem size increases, the efficiency first increases and reach the peak in 1000000 and then decline. As the number of threads increases, the efficiency generally continues goes down, except for the problem size 1000000 nums. This matches my expectation because generally we can reach the optimal efficiency when we have suitable data size and number of threads. After this point, the efficiency will go down. Before this point, the efficiency goes up. Thus, the result matches.

time	100	1,000	10,000	100,000	1,000,000	10,000,000	100,000,000
1	0.247708	0.247855	0.249976	0.247334	0.247634	0.248679	0.263497
2	0.26238	0.262427	0.26212	0.261853	0.261886	0.262242	0.271834
4	0.292702	0.2822	0.292073	0.291159	0.281293	0.292234	0.2902
8	0.345438	0.344592	0.337542	0.33733	0.328526	0.339459	0.34765

speedup	100	1,000	10,000	100,000	1,000,000	10,000,000	100,000,000
2	0.944081	0.944472	0.9536701	0.944553	0.945579	0.9482806	0.96933055
4	0.846281	0.878296	0.8558682	0.849481	0.880342	0.85095848	0.90798415
8	0.717084	0.719271	0.7405775	0.733211	0.753773	0.73257448	0.75793758

