

CSCE 735 Fall 2023 – HW4

Name: Ashutosh Chauhan

UIN: 232009024

1. (80 points) Complete the MPI-based code provided in `qsort_hypercube.cpp` to implement the parallel quicksort algorithm for a d -dimensional hypercube with $p=2^d$ processors. 60 points will be awarded if the code compiles and executes the following command successfully.

```
mpirun -np 2 ./qsort_hypercube.exe 4 -1
```

5 points will be awarded for each of the following tests that are executed successfully.

```
mpirun -np 4 ./qsort_hypercube.exe 4 -2
```

```
mpirun -np 8 ./qsort_hypercube.exe 4 -1
```

```
mpirun -np 16 ./qsort_hypercube.exe 4 0
```

```
mpirun -np 16 ./qsort_hypercube.exe 20480000 0
```

```
mpirun -np 2 ./qsort_hypercube.exe 4 -1
```

```
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 2 ./qsort_hypercube.exe 4 -1
[Proc: 0] number of processes = 2, initial local list size = 4, hypercube quicksort time = 0.001575
[Proc: 0] check_list: local_error = 0
[Proc: 1] check_list: local_error = 0
[Proc: 0] Congratulations. The list has been sorted correctly.
```

```
mpirun -np 4 ./qsort_hypercube.exe 4 -2
```

```
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 4 ./qsort_hypercube.exe 4 -2
[Proc: 0] number of processes = 4, initial local list size = 4, hypercube quicksort time = 0.005218
[Proc: 0] check_list: local_error = 0
[Proc: 1] check_list: local_error = 0
[Proc: 2] check_list: local_error = 0
[Proc: 3] check_list: local_error = 0
[Proc: 0] Congratulations. The list has been sorted correctly.
```

```
mpirun -np 8 ./qsort_hypercube.exe 4 -1
```

```
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 8 ./qsort_hypercube.exe 4 -1
[Proc: 0] number of processes = 8, initial local list size = 4, hypercube quicksort time = 0.007198
[Proc: 0] check_list: local_error = 0
[Proc: 1] check_list: local_error = 0
[Proc: 2] check_list: local_error = 0
[Proc: 3] check_list: local_error = 0
[Proc: 4] check_list: local_error = 0
[Proc: 5] check_list: local_error = 0
[Proc: 6] check_list: local_error = 0
[Proc: 7] check_list: local_error = 0
[Proc: 0] Congratulations. The list has been sorted correctly.
```

```
mpirun -np 16 ./qsort_hypercube.exe 4 0
```

```
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 16 ./qsort_hypercube.exe 4 0
[Proc: 0] number of processes = 16, initial local list size = 4, hypercube quicksort time = 0.010348
[Proc: 0] check_list: local_error = 0
[Proc: 1] check_list: local_error = 0
[Proc: 2] check_list: local_error = 0
[Proc: 3] check_list: local_error = 0
[Proc: 4] check_list: local_error = 0
[Proc: 5] check_list: local_error = 0
[Proc: 6] check_list: local_error = 0
[Proc: 7] check_list: local_error = 0
[Proc: 8] check_list: local_error = 0
[Proc: 9] check_list: local_error = 0
[Proc: 10] check_list: local_error = 0
[Proc: 11] check_list: local_error = 0
[Proc: 12] check_list: local_error = 0
[Proc: 13] check_list: local_error = 0
[Proc: 14] check_list: local_error = 0
[Proc: 15] check_list: local_error = 0
[Proc: 0] Congratulations. The list has been sorted correctly.
```

```
mpirun -np 16 ./qsort_hypercube.exe 20480000 0
```

```
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 16 ./qsort_hypercube.exe 20480000 0
[Proc: 0] number of processes = 16, initial local list size = 20480000, hypercube quicksort time = 2.559692
[Proc: 0] check_list: local_error = 0
[Proc: 1] check_list: local_error = 0
[Proc: 2] check_list: local_error = 0
[Proc: 3] check_list: local_error = 0
[Proc: 4] check_list: local_error = 0
[Proc: 5] check_list: local_error = 0
[Proc: 6] check_list: local_error = 0
[Proc: 7] check_list: local_error = 0
[Proc: 8] check_list: local_error = 0
[Proc: 9] check_list: local_error = 0
[Proc: 10] check_list: local_error = 0
[Proc: 11] check_list: local_error = 0
[Proc: 12] check_list: local_error = 0
[Proc: 13] check_list: local_error = 0
[Proc: 14] check_list: local_error = 0
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 15] check_list: local_error = 0
```

2. (5 points) *Weak Scalability Study*: Run your code to sort a distributed list of size $n \times p$ where n is the size of the local list on each process and p is the number of processes. For your experiments, use $n=20,480,000$ and $p = 1, 2, 4, 8, 16, 32$, and 64 . Set $\text{type}=0$. Plot the execution time, speedup, and efficiency of your code as a function of p . Use logarithmic scale for the x-axis.
- Note that the size of the list to be sorted is proportional to the number of processes p . In order to get speedup for a specific value of p , you need to determine the execution time to sort a list of size $n \times p$ with one process. As an example, speedup for $p = 4$ is the ratio of execution time for a list of size $81,920,000$ with one process (T_1) to the execution time for a list of size $20,480,000$ with 4 processes (T_4).

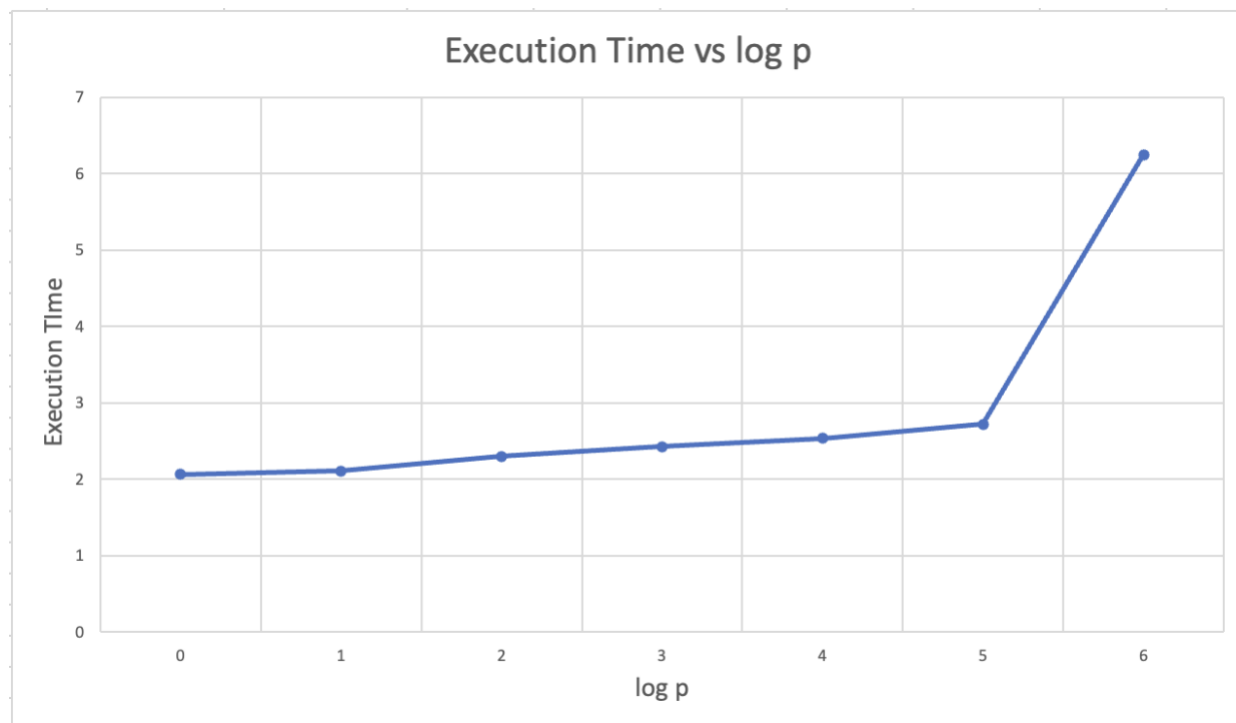
```

[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 1 ./qsort_hypercube.exe 1310720000 0
[Proc: 0] number of processes = 1, initial local list size = 1310720000, hypercube quicksort time = 163.641815
[Proc: 0] check_list: local_error = 0
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 1 ./qsort_hypercube.exe 655360000 0
[Proc: 0] number of processes = 1, initial local list size = 655360000, hypercube quicksort time = 76.725709
[Proc: 0] check_list: local_error = 0
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 1 ./qsort_hypercube.exe 327680000 0
[Proc: 0] number of processes = 1, initial local list size = 327680000, hypercube quicksort time = 37.463075
[Proc: 0] check_list: local_error = 0
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 1 ./qsort_hypercube.exe 163840000 0
[Proc: 0] number of processes = 1, initial local list size = 163840000, hypercube quicksort time = 18.685603
[Proc: 0] check_list: local_error = 0
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 1 ./qsort_hypercube.exe 81920000 0
[Proc: 0] number of processes = 1, initial local list size = 81920000, hypercube quicksort time = 8.831022
[Proc: 0] check_list: local_error = 0
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 1 ./qsort_hypercube.exe 40960000 0
[Proc: 0] number of processes = 1, initial local list size = 40960000, hypercube quicksort time = 4.389655
[Proc: 0] check_list: local_error = 0
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 1 ./qsort_hypercube.exe 20480000 0
[Proc: 0] number of processes = 1, initial local list size = 20480000, hypercube quicksort time = 2.062669
[Proc: 0] check_list: local_error = 0
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 2 ./qsort_hypercube.exe 20480000 0
[Proc: 0] number of processes = 2, initial local list size = 20480000, hypercube quicksort time = 2.112942
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 4 ./qsort_hypercube.exe 20480000 0
[Proc: 0] number of processes = 4, initial local list size = 20480000, hypercube quicksort time = 2.299575
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 8 ./qsort_hypercube.exe 20480000 0
[Proc: 0] number of processes = 8, initial local list size = 20480000, hypercube quicksort time = 2.429869
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 16 ./qsort_hypercube.exe 20480000 0
[Proc: 0] number of processes = 16, initial local list size = 20480000, hypercube quicksort time = 2.535947
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 32 ./qsort_hypercube.exe 20480000 0
[Proc: 0] number of processes = 32, initial local list size = 20480000, hypercube quicksort time = 2.723084
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 64 ./qsort_hypercube.exe 20480000 0
[Proc: 0] number of processes = 64, initial local list size = 20480000, hypercube quicksort time = 6.250126
[Proc: 0] Congratulations. The list has been sorted correctly.

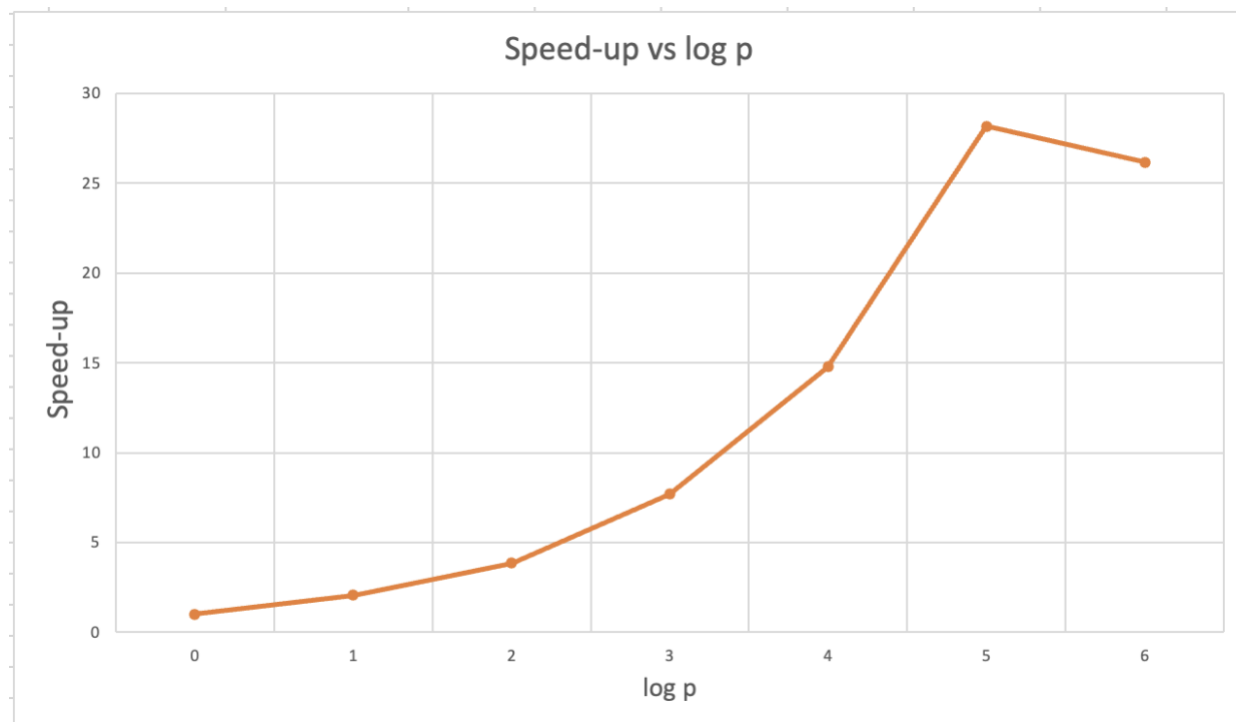
```

p	log p	n*p	T1	Tp	Speedup	Efficiency
1	0	20480000	2.062669	2.062669	1	1
2	1	40960000	4.389655	2.112942	2.077508517	1.038754258
4	2	81920000	8.831022	2.299575	3.8402844	0.9600711
8	3	163840000	18.685603	2.429869	7.689963121	0.96124539
16	4	327680000	37.463075	2.535947	14.77281465	0.923300916
32	5	655360000	76.725709	2.723084	28.1760346	0.880501081
64	6	1310720000	163.641815	6.250126	26.18216257	0.40909629

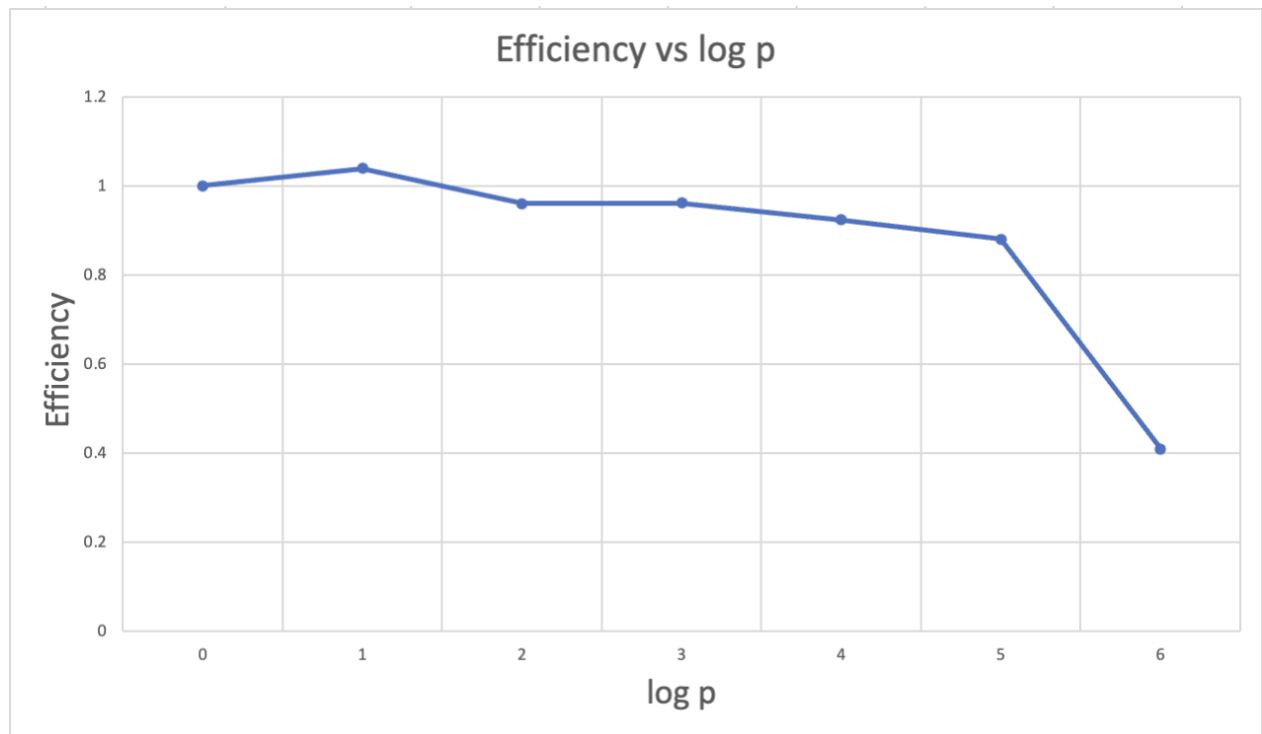
Execution Time Plot:



Speedup Plot:



Efficiency Plot:



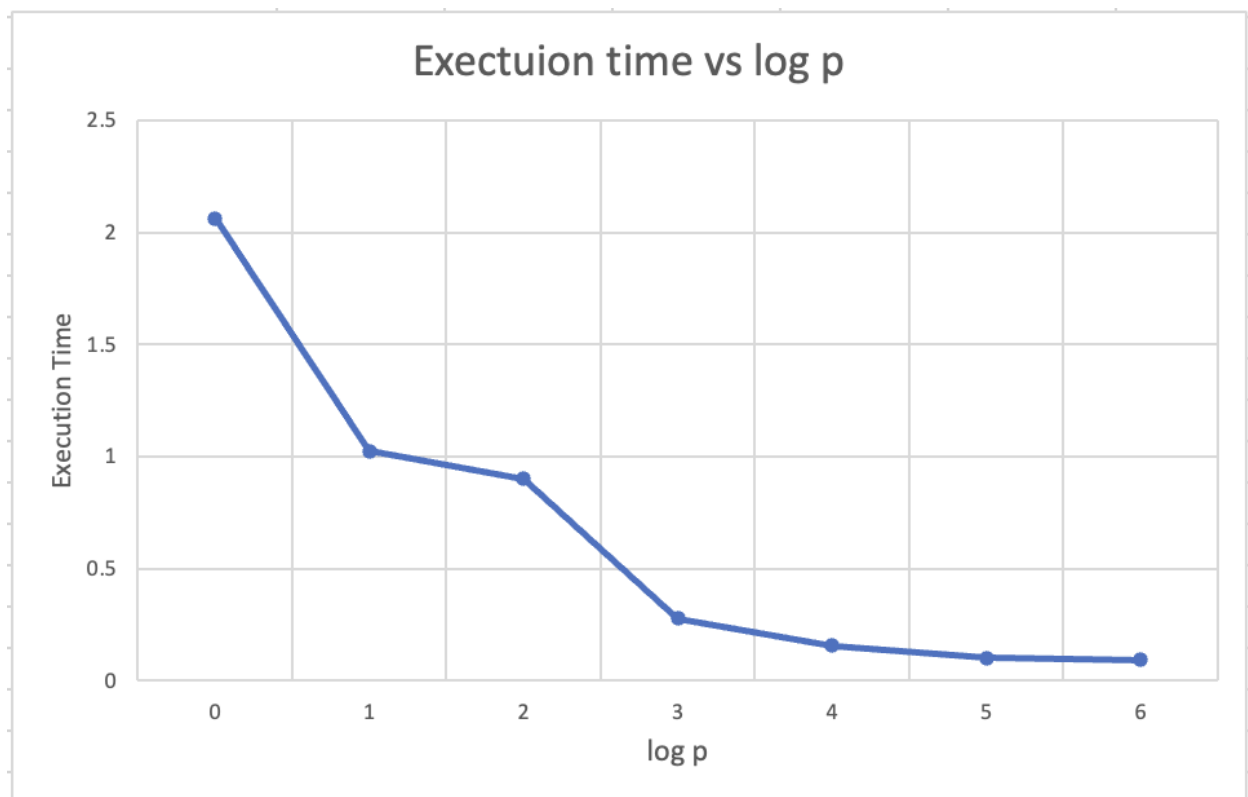
3. (5 points) *Strong Scalability Study*: Now run your code with $n=20,480,000/p$ where $p = 1, 2, 4, 8, 16, 32$, and 64. Set type=0. Plot the execution time, speedup, and efficiency of your code as a function of p . Use logarithmic scale for the x-axis.

Unlike the weak scalability study, here the size of the list to be sorted remains unchanged at 20,480,000 even as you increase the number of processes. To determine speedup for any p you need to compare the execution time on p processes to the execution time for a list of size 20,480,000 with one process.

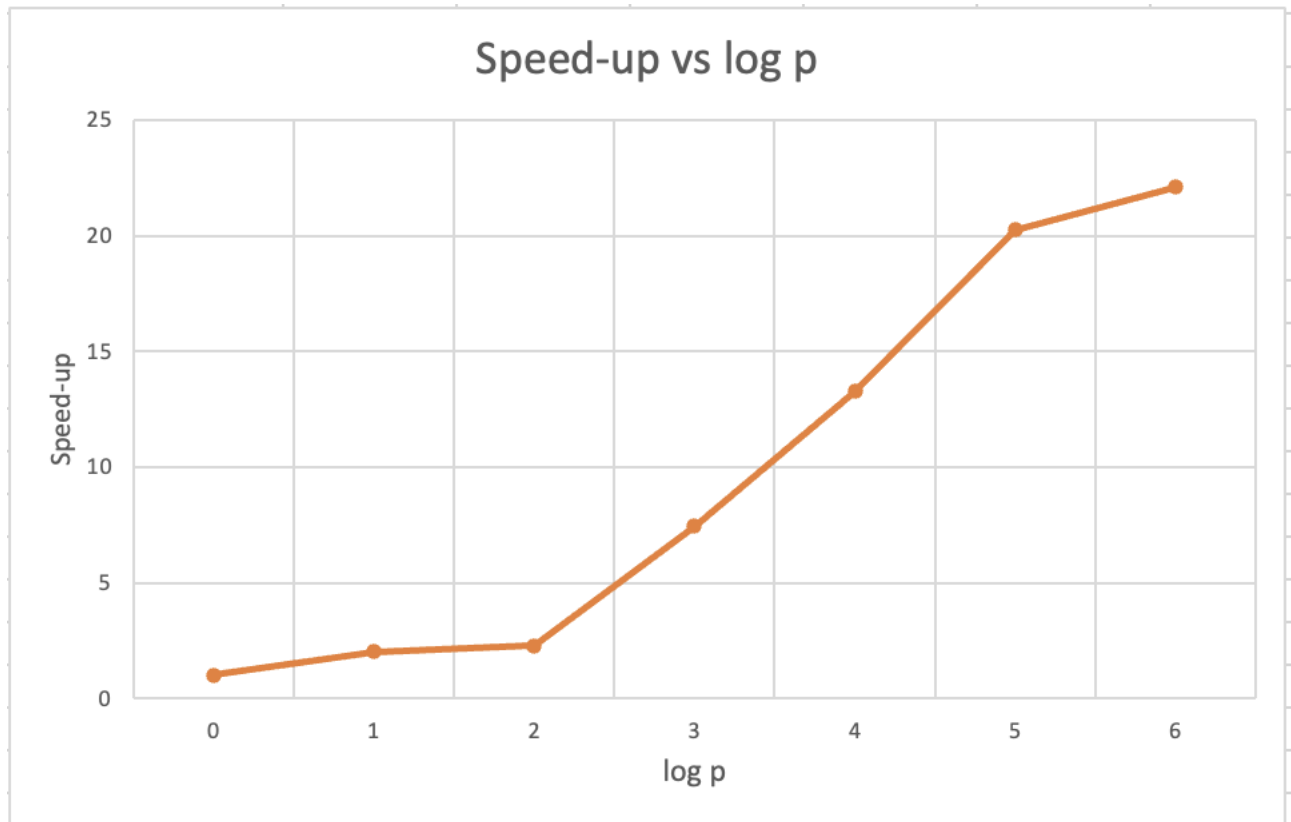
```
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 1 ./qsort_hypercube.exe 20480000 0
[Proc: 0] number of processes = 1, initial local list size = 20480000, hypercube quicksort time = 2.062902
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 2 ./qsort_hypercube.exe 10240000 0
[Proc: 0] number of processes = 2, initial local list size = 10240000, hypercube quicksort time = 1.025209
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 4 ./qsort_hypercube.exe 5120000 0
[Proc: 0] number of processes = 4, initial local list size = 5120000, hypercube quicksort time = 0.901140
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 8 ./qsort_hypercube.exe 2560000 0
[Proc: 0] number of processes = 8, initial local list size = 2560000, hypercube quicksort time = 0.277168
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 16 ./qsort_hypercube.exe 1280000 0
[Proc: 0] number of processes = 16, initial local list size = 1280000, hypercube quicksort time = 0.155519
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 32 ./qsort_hypercube.exe 640000 0
[Proc: 0] number of processes = 32, initial local list size = 640000, hypercube quicksort time = 0.101797
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 64 ./qsort_hypercube.exe 320000 0
[Proc: 0] number of processes = 64, initial local list size = 320000, hypercube quicksort time = 0.093303
[Proc: 0] Congratulations. The list has been sorted correctly.
```

p ▼	log p ▼	n/p ▼	Tp ▼	Speedup ▼	Efficiency ▼
1	0	20480000	2.062902	1	1
2	1	10240000	1.025209	2.012177029	1.006088515
4	2	5120000	0.90114	2.289213663	0.572303416
8	3	2560000	0.277168	7.442785603	0.9303482
16	4	1280000	0.155519	13.26463005	0.829039378
32	5	640000	0.101797	20.26486046	0.633276889
64	6	320000	0.093303	22.10970708	0.345464173

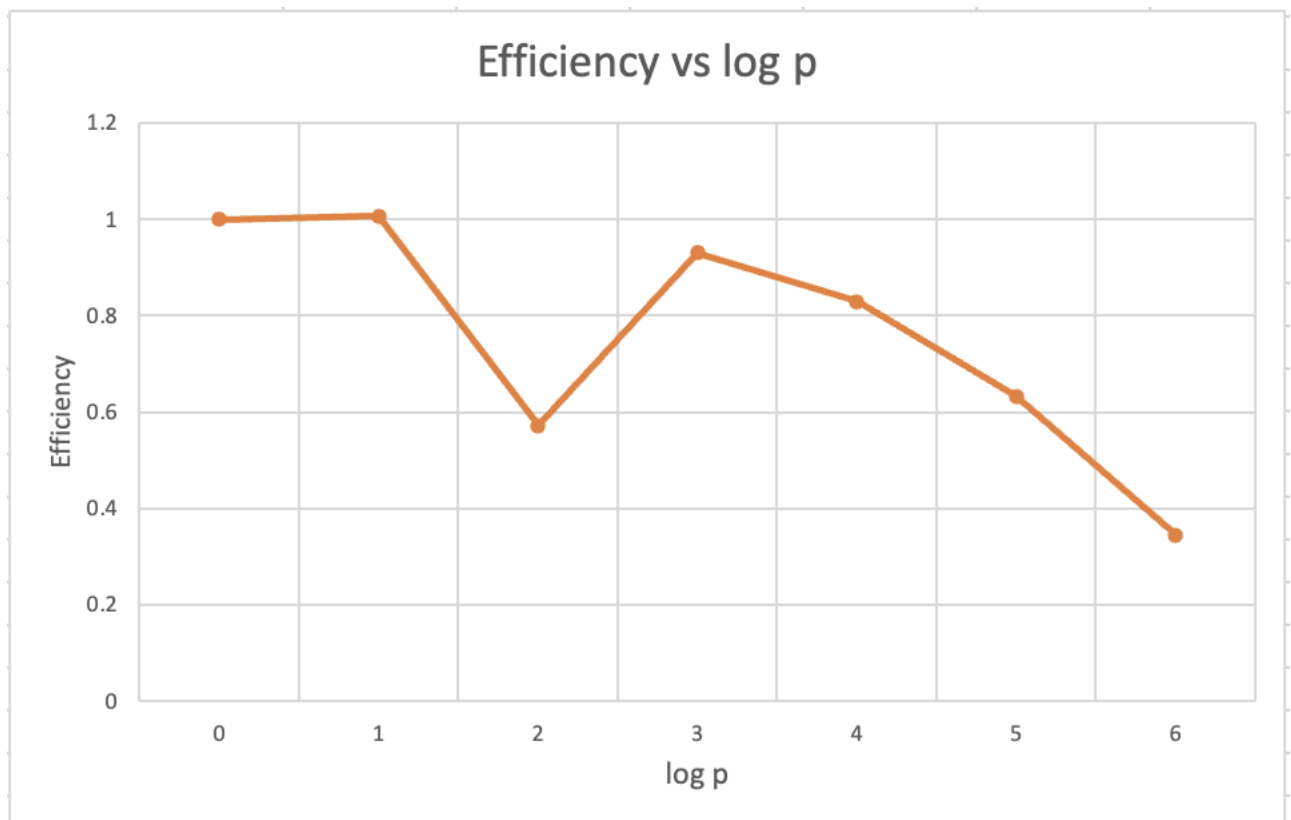
Execution Time Plot:



Speedup Plot:



Efficiency Plot:



4. (10 points) Modify the code to sort the list in descending order. Submit the modified code as `qsort_hypercube_descending.cpp`. 2 points will be awarded for each of the tests in Problem 1 that are executed successfully. (Note that the `check_list` routine needs to be modified to verify descending order.)

```
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 2 ./qsort_hypercube_descending.exe 4 -1
[Proc: 0] number of processes = 2, initial local list size = 4, hypercube quicksort time = 0.002093
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 4 ./qsort_hypercube_descending.exe 4 -2
[Proc: 0] number of processes = 4, initial local list size = 4, hypercube quicksort time = 0.000830
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 8 ./qsort_hypercube_descending.exe 4 -1
[Proc: 0] number of processes = 8, initial local list size = 4, hypercube quicksort time = 0.006599
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 16 ./qsort_hypercube_descending.exe 4 0
[Proc: 0] number of processes = 16, initial local list size = 4, hypercube quicksort time = 0.010168
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$ mpirun -np 16 ./qsort_hypercube_descending.exe 20480000 0
[Proc: 0] number of processes = 16, initial local list size = 20480000, hypercube quicksort time = 2.585349
[Proc: 0] Congratulations. The list has been sorted correctly.
[ashutosh@grace1 Ashu-HW4-735]$
```