Homework #2

(Due: April 26, 2019)

GROUP NUMBER: 2

Group Members	
Name	SBU ID
Wasif Khan	111675205
Jamshed Khan	112269083

1 Parallel Sorting

1.1 (a)

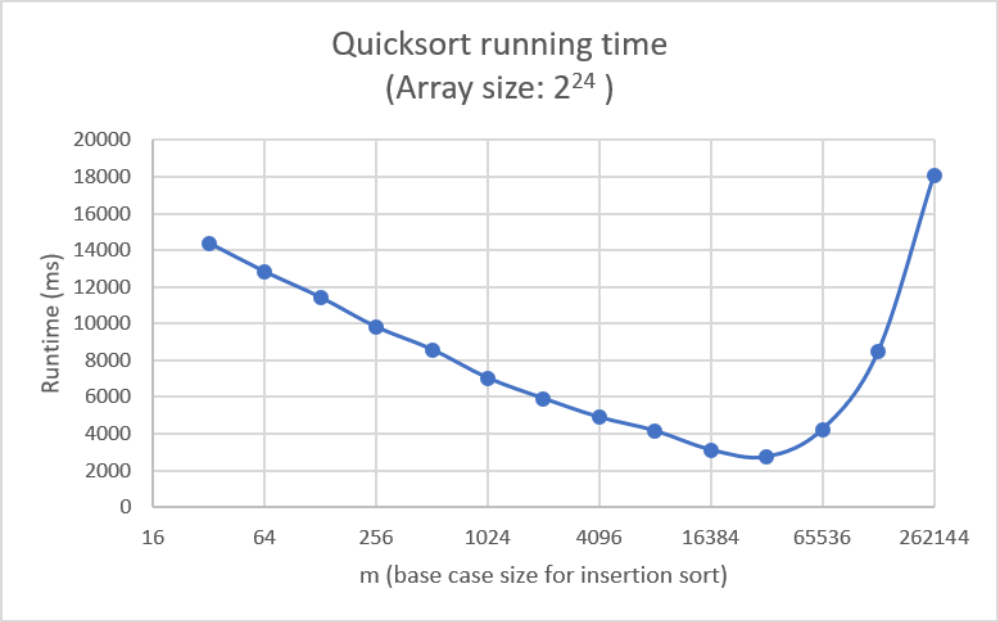
The codes for the two algorithms have been attached along with the submission.

1.2 (b)

Value of $n' = 16,777,216$ (i.e. 2^{24})

The resulting table and graph are shown below.

Value of m	Value of m (Power of 2)	Time taken (milliseconds)
32	2^5	14384
64	2^6	12820
128	2^7	11425
256	2^8	9806
512	2^9	8582
1024	2^{10}	7028
2048	2^{11}	5909
4096	2^{12}	4889
8192	2^{13}	4147
16384	2^{14}	3108
32768	2^{15}	2724
65536	2^{16}	4214
131072	2^{17}	8456
262144	2^{18}	18078



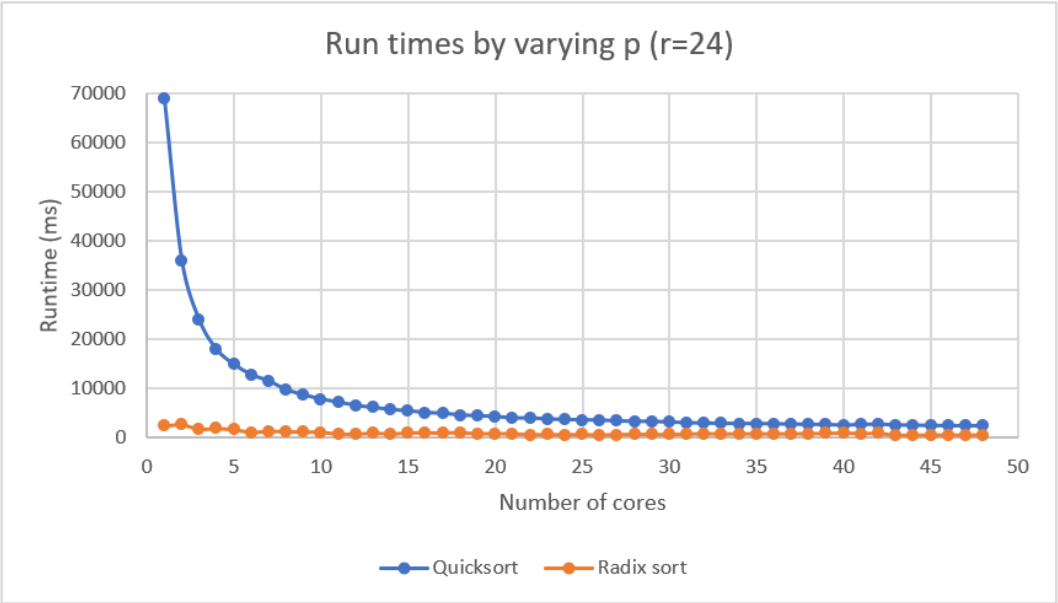
1.3 (c)

Here, $r = 24$.

The resulting table and graph are shown below.

Number of cores (p)	Time for quick sort (milliseconds)	Time for radix sort (milliseconds)
1	69024	2446
2	36069	2700
3	23971	1773
4	17940	1912
5	14891	1654
6	12800	1095
7	11511	1331
8	9727	1220
9	8711	1177
10	7822	1097
11	7202	727
12	6545	675
13	6111	944
14	5694	714
15	5446	1000
16	5089	966
17	4928	979
18	4581	1026
19	4429	810
20	4280	831
21	3982	788
22	3975	602
23	3773	614

24	3642	575
25	3595	628
26	3546	527
27	3365	590
28	3306	697
29	3270	636
30	3155	634
31	2991	723
32	2937	825
33	2869	747
34	2844	797
35	2847	775
36	2774	778
37	2729	801
38	2654	800
39	2651	891
40	2498	951
41	2656	837
42	2700	871
43	2540	488
44	2501	483
45	2424	503
46	2403	507
47	2377	513
48	2384	525

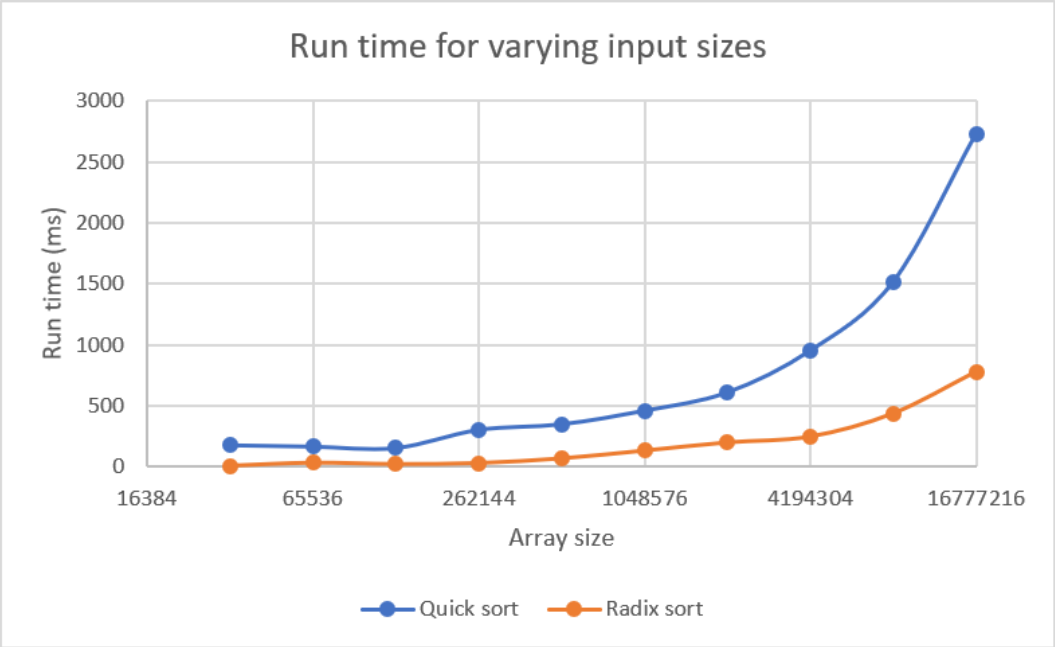


1.4 (d)

Here, $m = 32,768$ (i.e. 2^{15}) . We will vary the input size from 2^{15} to 2^{24} since the value of r was established to be 24 in part (c). Number of cores used would be 48 since it is the highest number of cores available on the *skx-dev* nodes for a single node.

The resulting table and graph are shown below.

Input size	Time for quick sort (milliseconds)	Time for radix sort (milliseconds)
2^{15}	177	10
2^{16}	166	35
2^{17}	154	24
2^{18}	303	32
2^{19}	349	70
2^{20}	460	134
2^{21}	610	201
2^{22}	953	249
2^{23}	1517	439
2^{24}	2731	781


Run time for varying input sizes
Run time (ms)
3000
2500
2000
1500
1000
500
0
16384
65536
262144
1048576
4194304
16777216
Array size
Quick sort
Radix sort

2 Parallel MSF

2.1 (a)

The randomized parallel minimum spanning forest (MSF) algorithm have been implemented with three different approaches for concurrent-write simulation:

- radix sort; (additional)
- radix sort with ranking by counting sort;
- binary search.

The implementations are attached with the submission of the homework. Besides, these are also available at this GitHub repository.

2.2 (b)

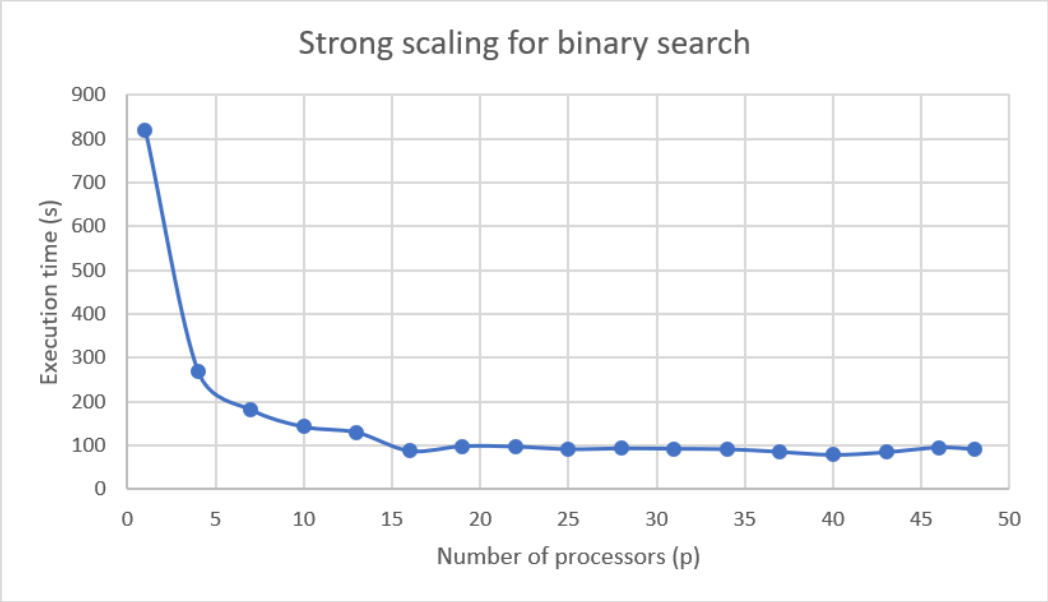
The following table tabulates the running times of the three MSF implementations (one additional) on 48 cores, which is the highest number of cores available on the *skx-dev* compute nodes. The output files are attached with the homework submission. If the name of the input file is `xxxxxx-in.txt`, then the corresponding output files are `xxxxx-MST-radix-out.txt`, `xxxxx-MST-sort-out.txt`, and `xxxxx-MST-search-out.txt` where concurrent write is simulated using radix sort, radix sort with ranking by counting sort, and binary search respectively.

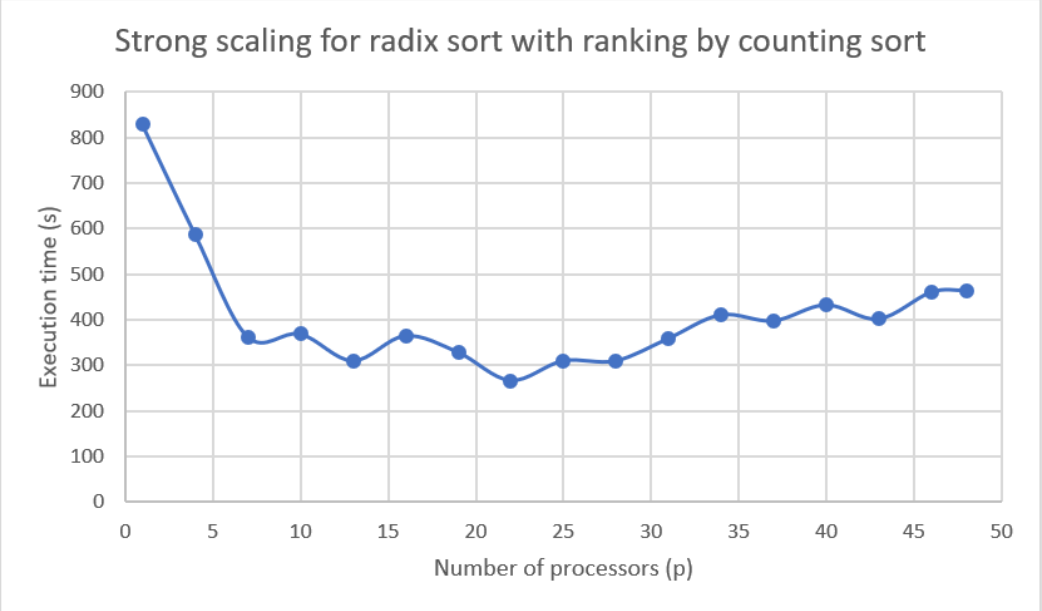
NB: For the largest graph, *com-friendster* ($n > 65$M, $m > 1.8$B), the $2m$-length edge-list alone would require ≈ 54 GB of memory, with two integer endpoints and one double floating point weight per edge. With other temporary memories required by the parallel algorithms, some of which require m length temporary arrays and copy of the edge-list, the extra space-complexity is $\Theta(m)$. The total memory required for the execution of the parallel algorithms on this file exceeds the memory provided by the supercomputing nodes to the general user.

Graph	Description	n (#vertices)	m (# edges)	Running time in seconds (Radix sort)	Running time in seconds (Radix sort with ranking by counting sort)	Running time in seconds (Binary search)
ca-AstroPh	Collaboration network of Arxiv Astro Physics	18.7K	396K	6	5	1
com-amazon	Amazon product network	334K	925K	13	16	5
com-dblp	DBLP collaboration network	317K	1M	15	25	5
roadNet-PA	Road network of Pennysylvania	1M	1.5M	31	39	15
roadNet-TX	Road network of Texas	1.4M	1.9M	35	45	18
roadNet-CA	Road network of California	2M	2.7M	53	83	28
as-skitter	Internet topology graph, from traceroutes run daily in 2005	1.7M	11M	210	199	30
com-lj	LiveJournal online social network	4M	34 M	548	504	95
com-orkut	Orkut online social network	3M	117 M	1479	1115	181

2.3 (c)

The code was executed on the *skx-dev* nodes which has a maximum of 48 cores to each node. We found out that the strong scaling trend can be found out using certain number of cores at interval. It's also found that radix sort with ranking by counting sort tend to increase in running time after a certain threshold region of core count is exceeded, with the speedup gradually decreasing. This finding may be attributed to certain increase in overheads once the threshold core count region is passed.s The results are shown in the graphs below.


Strong scaling for binary search
Execution time (s)
900
800
700
600
500
400
300
200
100
0
0
5
10
15
20
25
30
35
40
45
50
Number of processors (p)


Strong scaling for radix sort with ranking by counting sort
Execution time (s)
900
800
700
600
500
400
300
200
100
0
0
5
10
15
20
25
30
35
40
45
50
Number of processors (p)