Spring 2018 CSE613 HW2

Abiyaz Chowdhury
StonyBrook ID: 111580554

May 28, 2018

Task A

A, B

The running time for randomized quicksort (m = 32), with varying the size of the input, is shown below:

n	Time (ms)
128	0.08
256	0.05
512	0.09
1,024	0.17
2,048	0.33
4,096	0.63
8,192	1.3
16,384	2.79
32,768	5.65
65,536	11.34
131,072	26.82
262,144	54.64
524,288	103.21
1,048,576	184.45
2,097,152	464.74
4,194,304	926.11
8,388,608	1,610.77
16,777,216	3,222.84
33,554,432	6,143.68
67,108,864	13,170.8
134,217,728	25,389.5

We will use $n = 67,108,864(2^{26})$ for the subsequent calculations. Next, we vary the base case cutoff (using insertion sort for the base case) m, keeping n fixed:

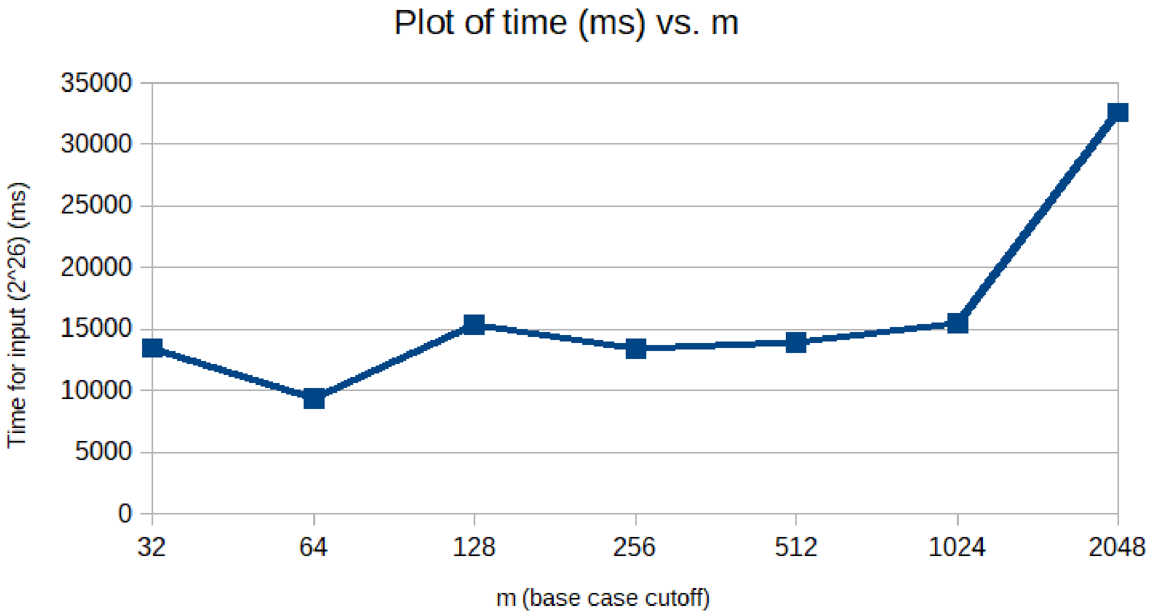

Figure 1: Time (ms) versus. m (base case cutoff)

The optimal value obtained is $m = 64$ for this input $n = 67,108,864$.

C

On one processing core, using $m = 64$, we find the largest n such that quicksort runs in under 2 minutes:

n	Time (ms)
1,024	0.16
2,048	0.34
4,096	0.75
8,192	1.58
16,384	3.7
32,768	7.71
65,536	17.73
131,072	36.15
262,144	76.5
524,288	159.17
10,248,576	332.9
2,097,152	703.27
4,194,304	1,455.94
8,388,608	2,997.58
16,777,216	6,161.48
33,554,432	12,938
67,108,864	24,312.9
134,217,728	51,203.8
268,435,456	101,078

Largest value obtained is $n = 268435456(2^{28})$. We do the same for radix sort:

n	Time (ms)
1,024	0.29
2,048	0.37
4,096	0.63
8,192	0.94
16,384	1.64
32,768	3.13
65,536	5.36
131,072	9.69
262,144	18.11
524,288	31.12
1,048,576	68.1
2,097,152	164.49
4,194,304	321.78
8,388,608	648.62
16,777,216	1,278.82
33,554,432	2,276.62
67,108,864	3,952.93
134,217,728	7,840.52
268,435,456	19,385
536,870,912	50,673.8
1,073,741,824	76,844.5

At this point, the program crashed (perhaps due to memory overload), but the highest was $n = 1073741824(2^{30})$. The highest value for both implementations is therefore $n = 268435456(2^{28})$. We now vary processor cores, and run on both algorithms using the input $n = 268435456(2^{28})$.

We obtain the following:

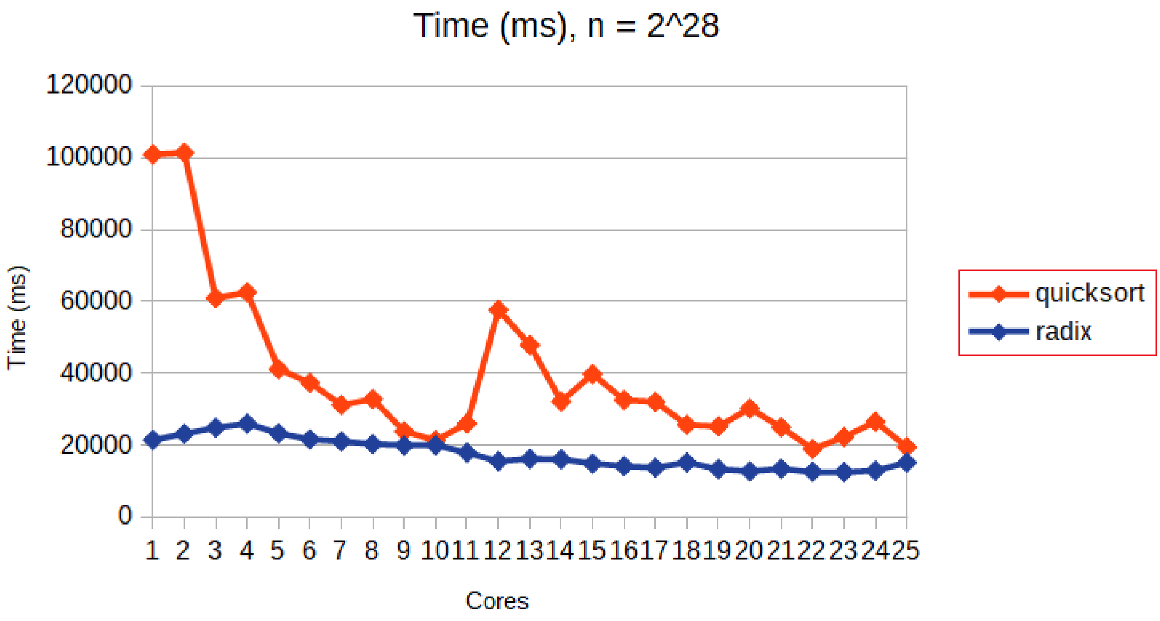

Figure 2: Time (ms) versus. number of cores

D

Finally, we vary n and measure the running time for both sorting algorithms on all cores.

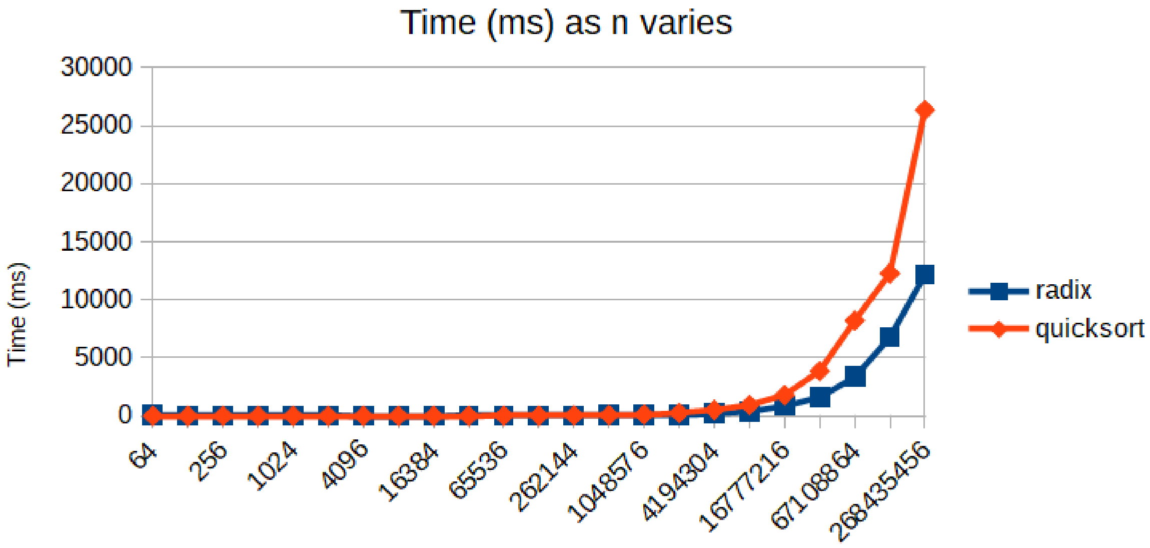

Figure 3: Time (ms) versus. n (input sizE)