Oniel Toledo

November 8, 2013

CS 302 Section 1001

Project 5

Report

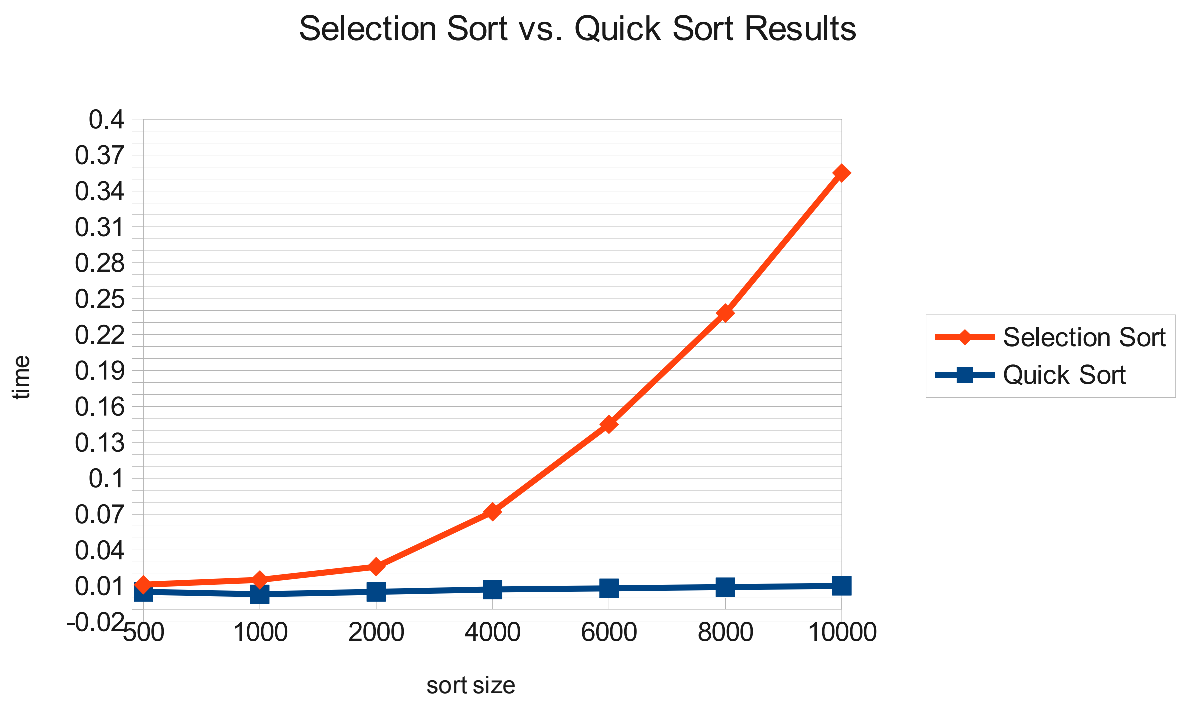
Introduction

In this lab report I discuss my experimental results when sorting using the selection and quick sort respectively. In my program (sorter.o) the user is asked to provided the type of sort and the size of the array to sort via command line arguments. From what I know, selection sort is of order $O(n^2)$ while quick sort has a more dynamic nature. At its fastest execution time quick sorts executes at O(n*log(n)), in the worse case scenario it will run at $O(n^2)$.

Hypothesis

I expect my quick sort results to be much faster than my selection sort for large number sorts, however, in smaller sorts quick sort should have a time complexity of $O(n^2)$.

Results



Data

sort type	size	time
selection	10000	0.0345s
selection	8000	0.229s
selection	6000	0.137s
selection	4000	0.065s
selection	2000	0.021s
selection	1000	0.012s
selection	500	0.006s
quick	10000	0.010s
quick	8000	0.009
quick	6000	0.008
quick	4000	0.007
quick	2000	0.005
quick	1000	0.003
quick	500	0.005

Conclusion

Based on my results it is obvious that quick sort is a a lot faster than selection sort. What is important to note that quick sort's speed deteriorates at lower comparison sizes. In my 500 sort size quick sort took longer than when sorting 1000. Somewhere along that boundary the transition from $O(n*\log(n))$ to $O(n^2)$ took place. I also inputted sizes 50, 100, and 200 and the results for selection and quick sort were the same, the time complexity of quick sort had reached $O(n^2)$.

Raw Data from terminal

```
oniel@gateway:~$ time ./sorter quick 10000

real	0m0.010s
user	0m0.004s
sys	0m0.004s
oniel@gateway:~$ clear

oniel@gateway:~$ time ./sorter quick 10000

real	0m0.009s
user	0m0.000s
sys	0m0.004s
oniel@gateway:~$ time ./sorter quick 10000

real	0m0.010s
user	0m0.004s
sys	0m0.004s
oniel@gateway:~$ time ./sorter quick 8000

real	0m0.009s
user	0m0.008s
sys	0m0.000s
oniel@gateway:~$ time ./sorter quick 6000

real	0m0.008s
user	0m0.004s
sys	0m0.000s
oniel@gateway:~$ time ./sorter quick 4000

real	0m0.007s
user	0m0.000s
sys	0m0.004s
oniel@gateway:~$ time ./sorter quick 2000

real	0m0.005s
user	0m0.000s
sys	0m0.004s
oniel@gateway:~$ time ./sorter quick 1000

real	0m0.003s
user	0m0.000s
sys	0m0.000s
oniel@gateway:~$ time ./sorter quick 500
```

```
real    0m0.005s
user    0m0.000s
sys     0m0.004s
oniel@gateway:~$ time ./sorter quick 500

real    0m0.005s
user    0m0.004s
sys     0m0.000s
oniel@gateway:~$ time ./sorter quick 100

real    0m0.005s
user    0m0.004s
sys     0m0.000s
oniel@gateway:~$ time ./sorter quick 20000

real    0m0.012s
user    0m0.012s
sys     0m0.000s
oniel@gateway:~$ time ./sorter quick 200

real    0m0.005s
user    0m0.000s
sys     0m0.000s
oniel@gateway:~$ time ./sorter selection 10000

real    0m0.345s
user    0m0.336s
sys     0m0.008s
oniel@gateway:~$ time ./sorter selection 8000

real    0m0.229s
user    0m0.224s
sys     0m0.004s
oniel@gateway:~$ time ./sorter selection 6000

real    0m0.137s
user    0m0.136s
sys     0m0.000s
oniel@gateway:~$ time ./sorter selection 4000

real    0m0.065s
user    0m0.068s
sys     0m0.000s
oniel@gateway:~$ time ./sorter selection 2000

real    0m0.021s
user    0m0.016s
sys     0m0.000s
```

```
oniel@gateway:~$ time ./sorter selection 1000

real	0m0.012s
user	0m0.008s
sys	0m0.004s
oniel@gateway:~$ time ./sorter selection 500

real	0m0.006s
user	0m0.004s
sys	0m0.000s
oniel@gateway:~$ time ./sorter selection 50

real	0m0.004s
user	0m0.000s
sys	0m0.000s
oniel@gateway:~$ time ./sorter quick 50

real	0m0.004s
user	0m0.000s
sys	0m0.004s
oniel@gateway:~$
```