James Yim

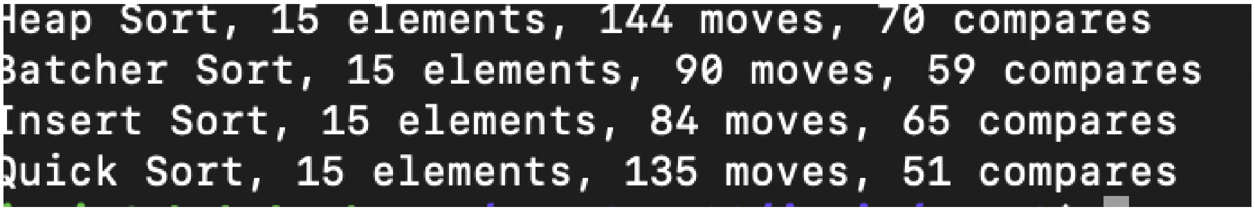
CSE13s Winter 2022

Assignment 3 Writeup.pdf

Introduction

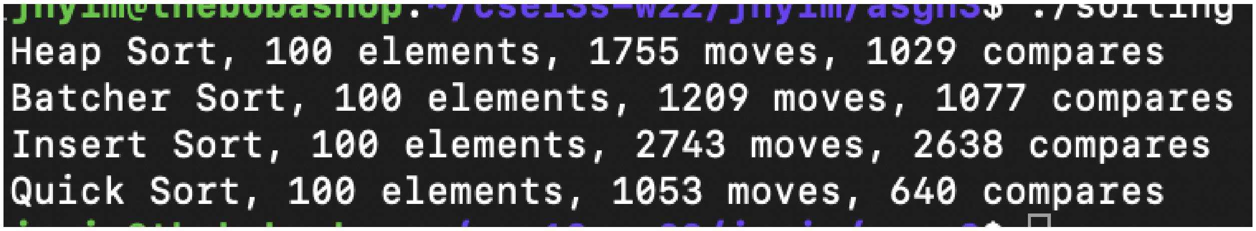
This writeup will focus on how different sorting algorithms perform differently, despite ultimately ending up with the same result. We will focus on four different types of sorts. Insertion Sort, Heap Sort, Batcher Sort, and Quick Sort, will be the ones we will look at. The statistics that we will be using to compare are their amount of moves and comparisons that they perform with a set amount of elements.

No, they aren't all the same



```
Heap Sort, 15 elements, 144 moves, 70 compares
Batcher Sort, 15 elements, 90 moves, 59 compares
Insert Sort, 15 elements, 84 moves, 65 compares
Quick Sort, 15 elements, 135 moves, 51 compares
```

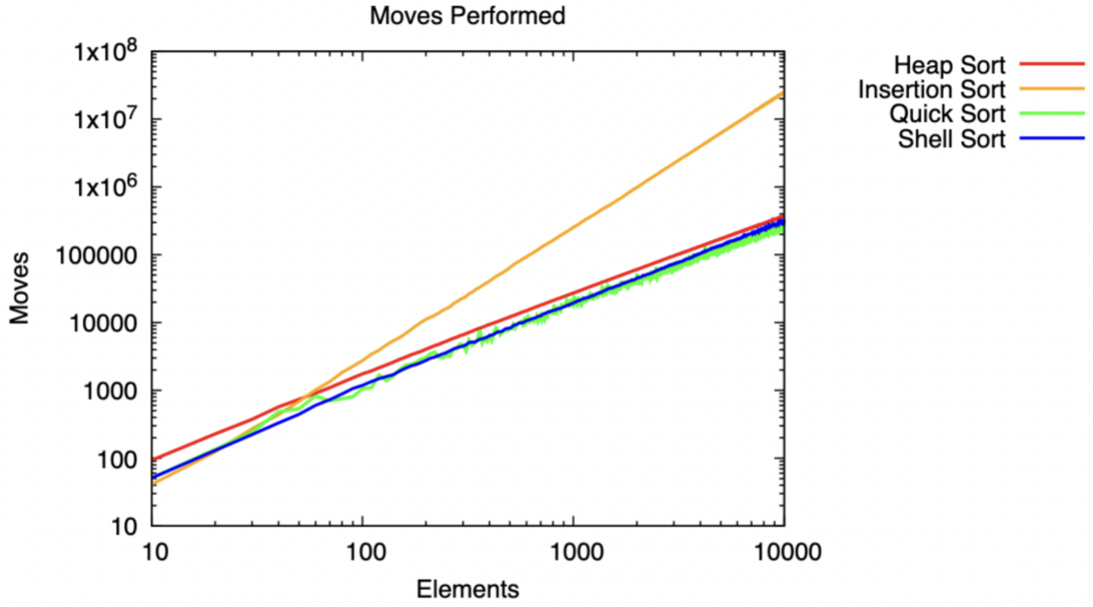
Here we have our four different sorting algorithms sorting 15 random elements. We can see that Heap sort performed with the most amount of moves and compares, whereas Insertion sort performed with the least amount of comparisons and moves.



```
jnyim@thebobashop:~/cse13s-w22/jnyim/asgn3$ ./sorting
Heap Sort, 100 elements, 1755 moves, 1029 compares
Batcher Sort, 100 elements, 1209 moves, 1077 compares
Insert Sort, 100 elements, 2743 moves, 2638 compares
Quick Sort, 100 elements, 1053 moves, 640 compares
```

Here we have the same four sorting algorithms but sorting 100 random elements. The results differ this time. Here Insertion Sort performs the worst and Heap sort performs better than before. The best performance being Quick Sort.

Why is this? Well it has to do with the scale of elements.



graph from Assignment 3 pdf
We can see that at lower element numbers, insertion sort is quick. while the others are higher. As we get to a bigger amount of elements Insertion sort starts to struggle. This has to do with its n^2 computational complexity, vs the nlog(n) complexities of the other sorts. We can see that the three other sorts are seemingly the same.

Summary

It is important to think about what we are doing when we are programming. Although all sorting algorithms end up with the same result, it is how they get there that it is important to think about. As programmers we need to think about what we need from our tools and in this case think about what we are sorting. We have many different ways of doing the same thing, we just need to choose the best option for our needs.

message to graders:
I am sorry that this writeup is really lazy. I already didn't finish the code in time and I am really burnt out on this specific assignment. Will make up for it on the next one! Thank you for all you guys do <3.