Assignment 3
Sorting: Putting your affairs in order

WRITEUP

Zack Traczyk
CSE13S - Spring 2021

Due: April 25th at 11:59 pm

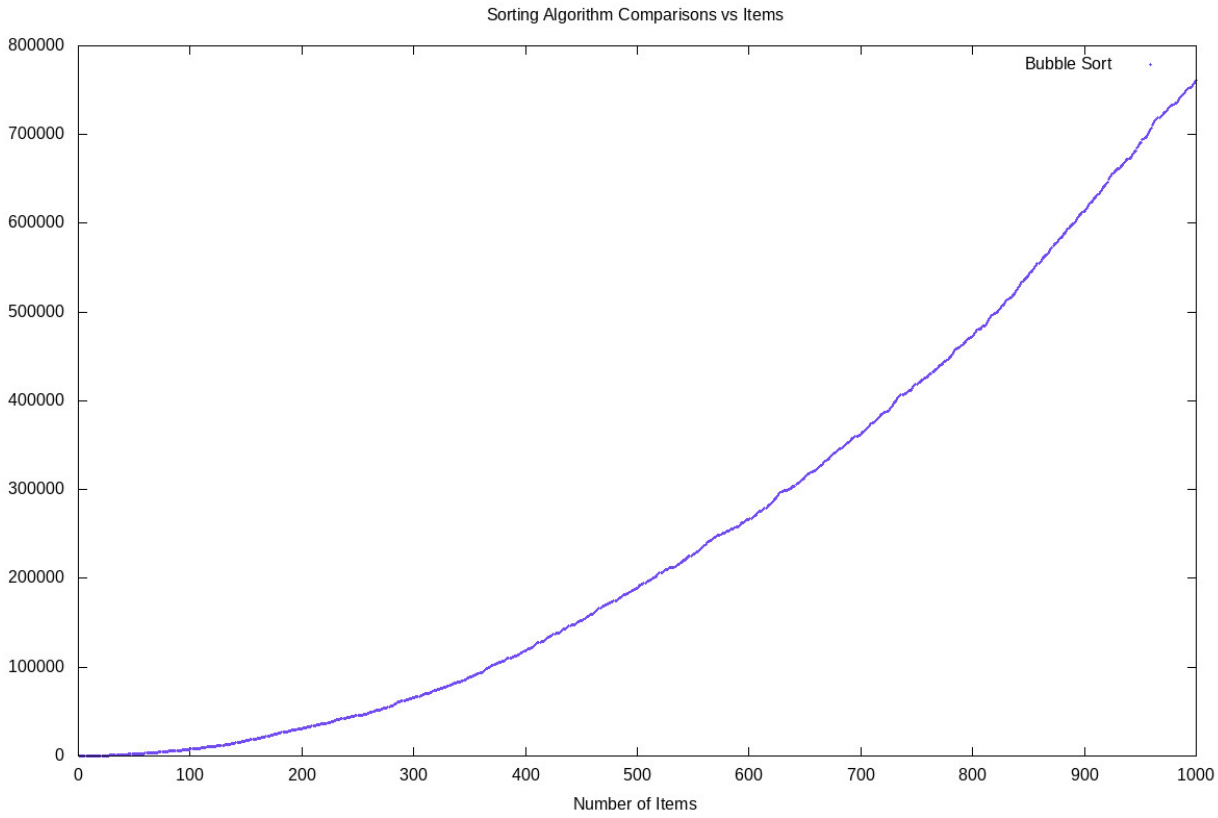
1 Time Complexity

1.1 Bubble Sort

For larger arrays, Bubble Sort had the worst time complexity out of all the sorts implemented. Figure 1.1 displays the relationship between the amount of comparisons the algorithm performs for the number of items of the given array. As shown, Bubble Sort has a worst case complexity of $O(n^2)$ as it has two nested for loops that usually iterate through the entirety of the array.

A reversed array requires the most amount of comparisons in bubble sort.

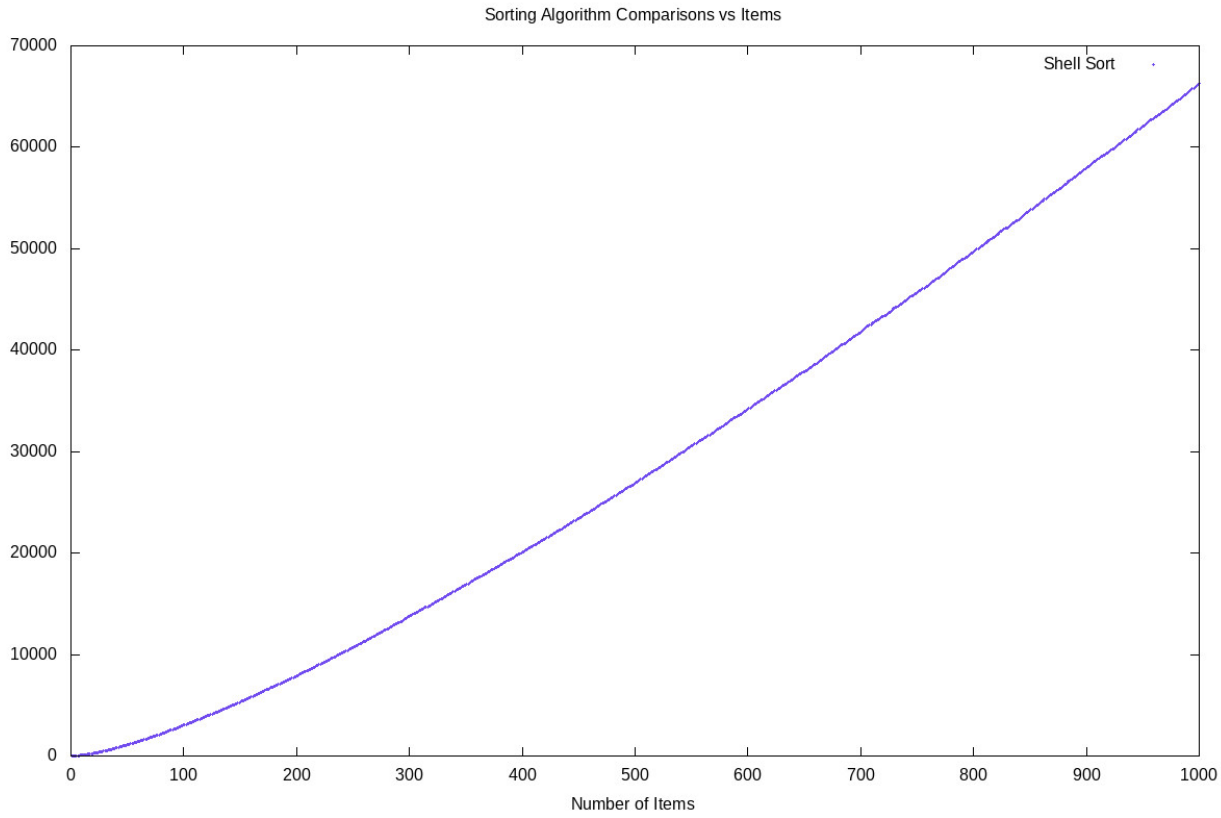
The corresponding C value for bubble sort is



1.2 Shell Sort

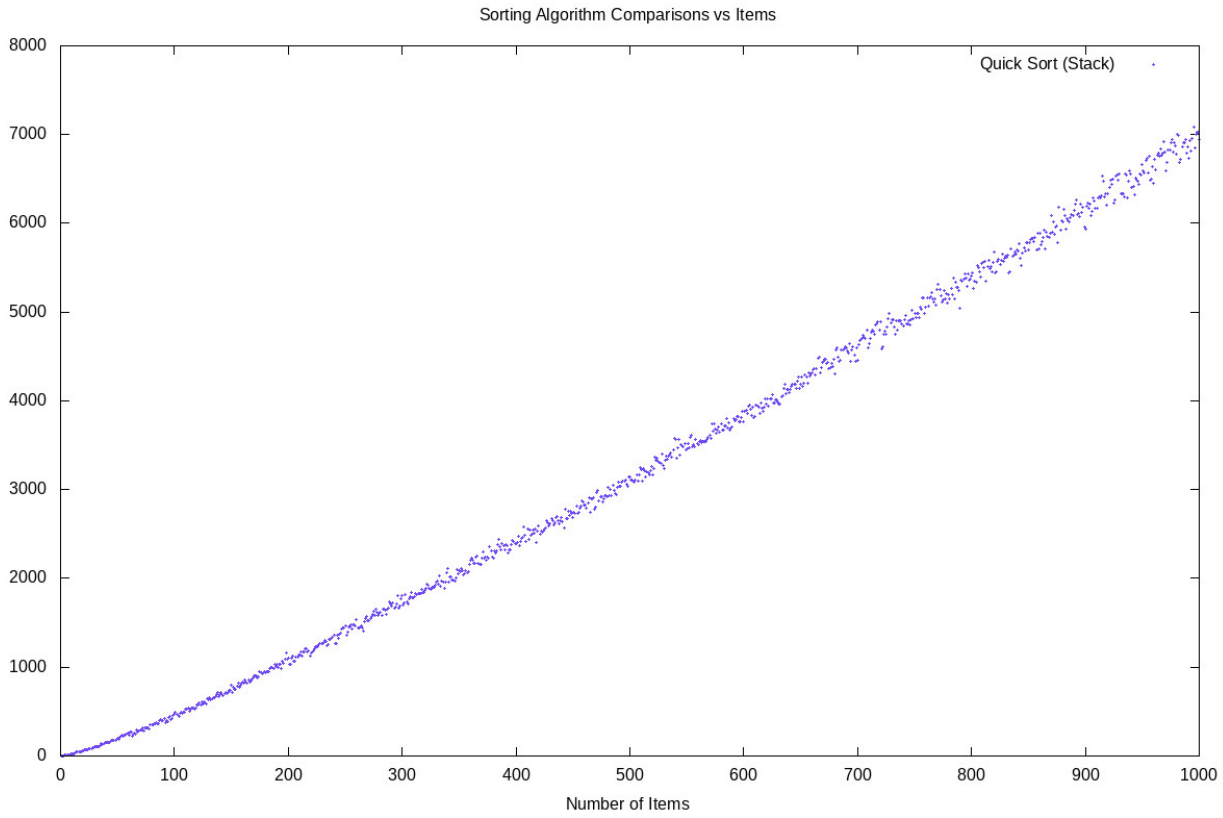
Shell sort preformed much better than Bubble Sort. Figure 1.1.

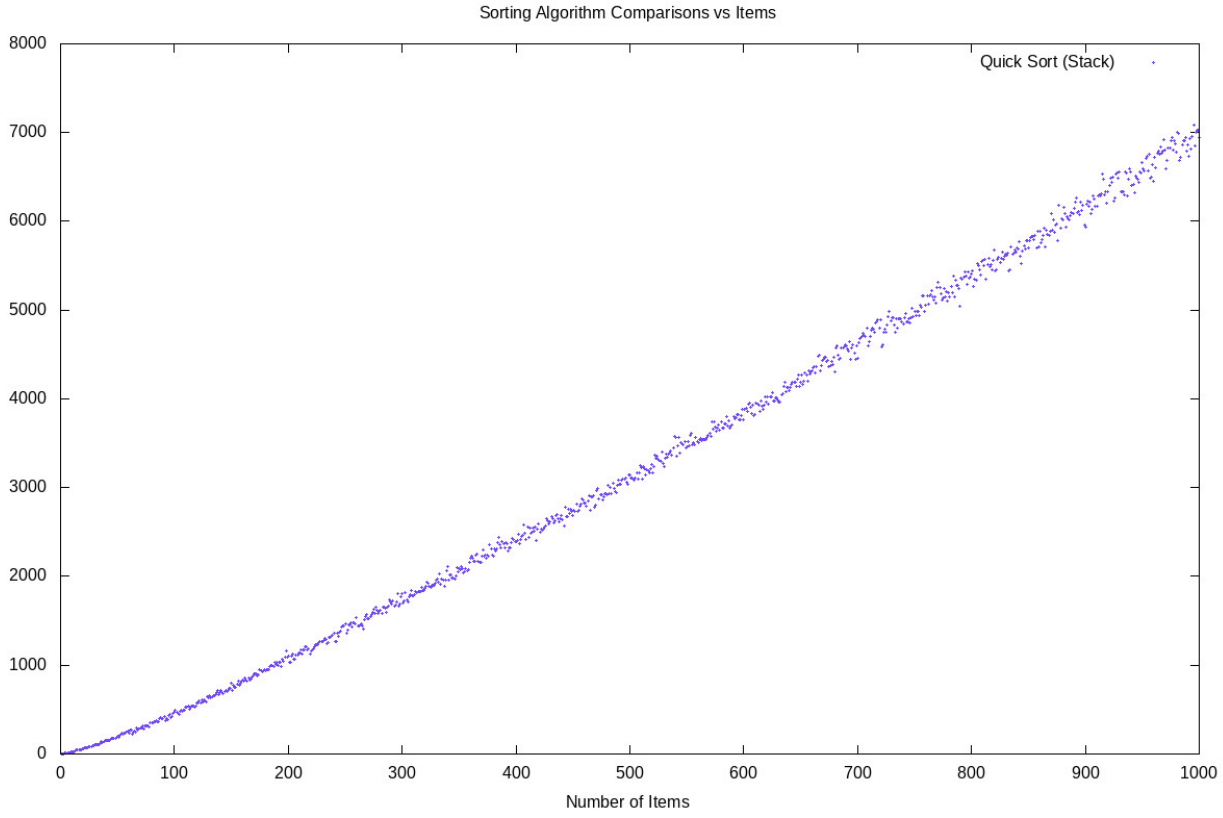
Reverse



1.3 Quick Sort

Shell sort preformed much better than Bubble Sort. Figure 1.3. Figure 1.3


Sorting Algorithm Comparisons vs Items
Quick Sort (Stack)
8000
7000
6000
5000
4000
3000
2000
1000
0
0
100
200
300
400
500
600
700
800
900
1000
Number of Items


Sorting Algorithm Comparisons vs Items
Quick Sort (Stack)
8000
7000
6000
5000
4000
3000
2000
1000
0
0
100
200
300
400
500
600
700
800
900
1000
Number of Items

More graphs

2 What I Learned

Picking the right algorithm is crucial for accomplishing fast code. Furthermore, right does not necessarily mean smallest Big-O or even the fastest. It is about considering memory usage, and size of array.