Report: PRRPRR01 Sorting Algorithms

Description of the program:

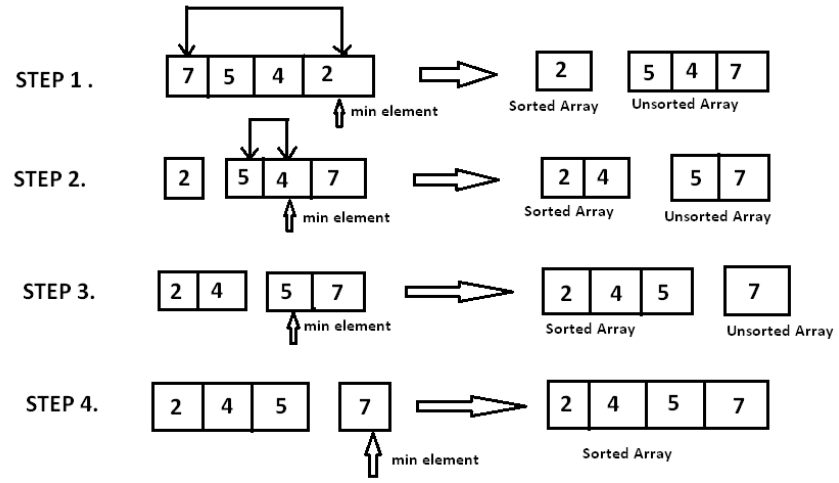
The program creates an array of the class People which contains a string name and an int age. It also creates a linked list containing a few numbers (this is a representation of the inevitable future in which names have gone obsolete and everyone is identified by a number (not laziness)). After this the program waits for user input. Depending on what is typed the linked list, or the array will be sorted by the specified algorithm.

Description of sorting algorithms:

Selection Sort:

Sorts an array by finding the smallest element from a subarray and putting it in the beginning of the sorted subarray.

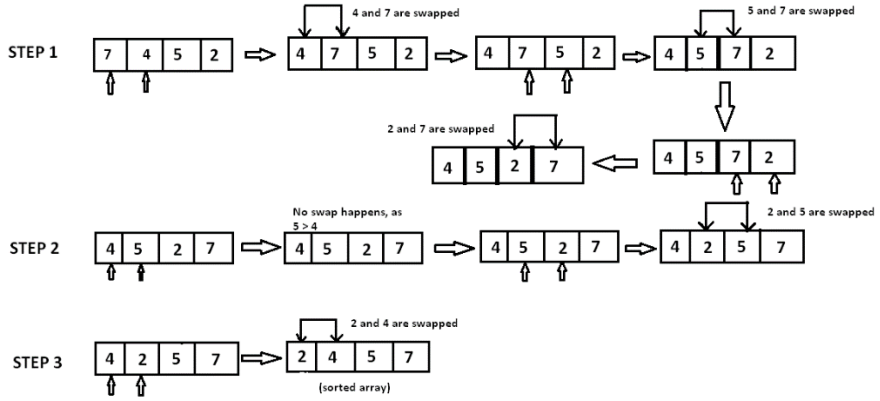
The time complexity is $O(n^2)$ since there are two nested loops



Bubble Sort:

Sorts an array by swapping adjacent elements if they are in the wrong order.

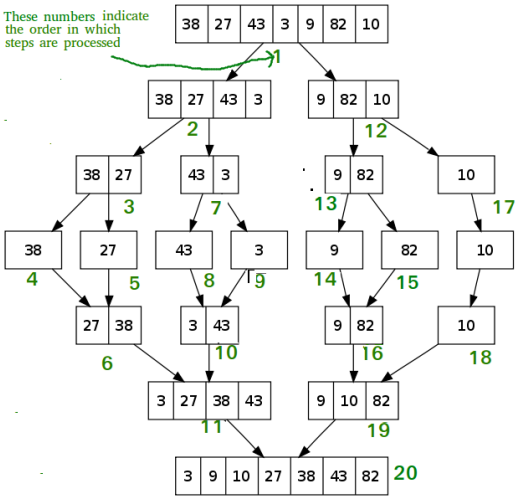
The time complexity is in the worst and average-case $O(n^n)$ however if the list is already sorted the time complexity will be O(n)



Merge Sort (Linked List):

Is a divide and conquer algorithm. It sorts the linked list (or array) by recursively dividing it into halves and putting it in the appropriate order until it reaches the size of one. It then begins merging the sole parts in the same manner as it divided them in the appropriate order.

The rime complexity of a Merge Sort is O(n*Log(*n*)*)



Bogo Sort:

Sorts the array by first checking if it is sorted otherwise randomizing the order until it is sorted.

The time complexity is in the worst-case O(∞) Since it is completely random the algorithm has no upper bound. The best case occurs if the array already sorted in which case it is O(n).

Gnome Sort:

Sorts the array by looking at the previous element and comparing it to the next element. If they are in the right order it continues to the next one otherwise it swaps them.

The time complexity is O(n^2) although it can be improved if the array is fully or partially sorted.

34 2 10 -9
2 34 10 -9
2 34 10 -9
2 10 34 -9
2 10 34 -9
2 10 34 -9
2 10 34 -9
2 10 -9 34
2 10 -9 34
2 -9 10 34
2 -9 10 34
-9 2 10 34
-9 2 10 34
-9 2 10 34
-9 2 10 34(Sorted output)

Description of the development process:

The development went smoothly. The core of the program was finished day 1 without any issues. Implementation of further sorting algorithms went rather smoothly as well. Since this was something I hadn't done before I spent a lot of time on Stackoverflow and GeeksforGeeks trying to learn the different types of algorithms (as well as borrowing some code). The linked list was also easy to implement, and I did not really have any significant issues. Overall, everything went smooth.

Things I have learned:

I have learned about O, different types of sorting algorithms and how to create a linked list.

Things I would like to learn:

I would like to learn a bunch of stuff, can't think of anything specific.