

CSE1301 Exercise Sheet 11 Sorting and List Manipulation

Exercise 1

- a) Apply the three sorting algorithms presented in class (selection sort, insertion sort and bubble sort) to sort the following list of names in ascending order:
`Sarah, Jane, Erin, Jessie, Susan, Amy`
- b) Modify the selection sort function so that it can sort a list of names (strings) in alphabetical order (rather than a list of numbers).
- c) Modify the insertion sort function so that it receives an additional parameter that determines the direction of the sorting operation (ascending or descending).
- d) Modify the bubble sort function so that it sorts an array of structs, and it receives as a parameter the struct member to be used for sorting, e.g., student ID or name.

Exercise 2

Design an algorithm that, given a “pivot” element divides an unsorted list of numbers into two lists, the first list contains all elements that are smaller than the pivot element, the second part contains all elements greater than or equal to the pivot element. What is the time complexity of your algorithm. Can you gain anything by having a pre-sorted initial list of numbers? Consider the case where you use the operation only once on a given list and the case where you use it multiple times. Code your algorithm as a C function.

Exercise 3

Consider the operations Find, Add and Delete. How are these operations performed on a sorted list and an unsorted list? Compare the time complexity of these operations when applied to a sorted list and when applied to an unsorted list.

Exercise 4

Write an algorithm to remove all duplicates from an unsorted list of numbers. What is the time complexity of your algorithm? Can you achieve $O(n \log n)$ for a list length n ?

Exercise 5

An array of size N is being used to store a list of names (strings) sorted in alphabetical order.

- a) Modify the algorithm `addElement` and the corresponding function presented in class so that it handles names.
- b) Write an algorithm to delete a name from the list. Code your algorithm as a C function called `deleteElement`.