Lab Manual

CL2001 – Data Structures

Fall-2023

National University of Computer and Emerging Sciences-FAST
Karachi Campus

Data Structures Lab#3

Course: Data Structures (CL2001) **Semester:** Fall 2023

Instructor: Shafique Rehman **T.A:** N/A

Note:

- Maintain discipline during the lab.
- Listen and follow the instructions as they are given.
- Just raise hand if you have any problem.
- Completing all tasks of each lab is compulsory.
- Get your lab checked at the end of the session**.**

Contents:

- Sorting (Selection Sort and Insertion Sort)
- Searching (Linear Search, Binary Search and interpolation Search)

Selection Sort:

Selection sort is a simple and efficient sorting algorithm that works by repeatedly selecting the smallest (or largest) element from the unsorted portion of the list and moving it to the sorted portion of the list. The algorithm repeatedly selects the smallest (or largest) element from the unsorted portion of the list and swaps it with the first element of the unsorted portion. This process is repeated for the remaining unsorted portion of the list until the entire list is sorted. One variation of selection sort is called "Bidirectional selection sort" that goes through the list of elements by alternating between the smallest and largest element, this way the algorithm can be faster in some cases.

Follow the below steps to solve the problem:

Initialize minimum value(min_idx) to location 0.
Traverse the array to find the minimum element in the array.
While traversing if any element smaller than min_idx is found then swap both the values.
Then, increment min_idx to point to the next element.
Repeat until the array is sorted.

Time Complexity
Best: ?(n^2)
Average: ?(n^2)
Worst: O(n^2)
Space Complexity: O(1)

Insertion Sort:

Insertion sort is a simple sorting algorithm that works similar to the way you sort playing cards in your hands. The array is virtually split into a sorted and an unsorted part. Values from the unsorted part are picked and placed at the correct position in the sorted part.

Characteristics of Insertion Sort:
This algorithm is one of the simplest algorithm with simple implementation
Basically, Insertion sort is efficient for small data values
Insertion sort is adaptive in nature, i.e. it is appropriate for data sets which are already partially sorted.
Working of Insertion Sort algorithm:
Consider an example: arr[]: {12, 11, 13, 5, 6}

12 11 13 5 6

First Pass:

Initially, the first two elements of the array are compared in insertion sort.

12 11 13 5 6

Here, 12 is greater than 11 hence they are not in the ascending order and 12 is not at its correct position. Thus, swap 11 and 12.
So, for now 11 is stored in a sorted sub-array.

11 12 13 5 6

Second Pass:

Now, move to the next two elements and compare them

11 12 13 5 6

Here, 13 is greater than 12, thus both elements seems to be in ascending order, hence, no swapping will occur. 12 also stored in a sorted sub-array along with 11

Third Pass:

Now, two elements are present in the sorted sub-array which are 11 and 12
Moving forward to the next two elements which are 13 and 5

11 12 13 5 6

Both 5 and 13 are not present at their correct place so swap them

11 12 5 13 6

After swapping, elements 12 and 5 are not sorted, thus swap again

11 5 12 13 6

Here, again 11 and 5 are not sorted, hence swap again

5 11 12 13 6

Here, 5 is at its correct position

Fourth Pass:

Now, the elements which are present in the sorted sub-array are 5, 11 and 12
Moving to the next two elements 13 and 6

5 11 12 13 6

Clearly, they are not sorted, thus perform swap between both

5 11 12 6 13

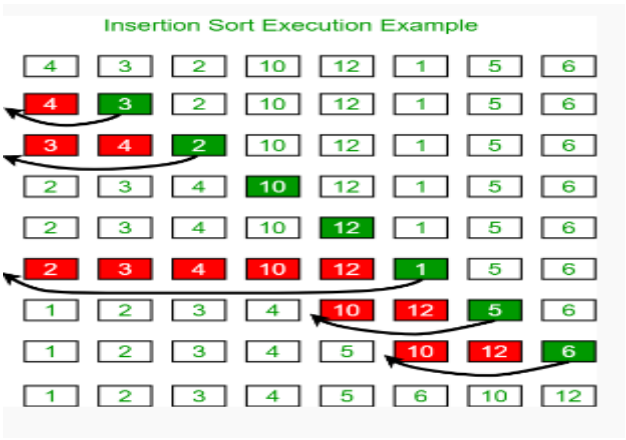
Now, 6 is smaller than 12, hence, swap again

5 11 6 12 13

Here, also swapping makes 11 and 6 unsorted hence, swap again

5 6 11 12 13

Finally, the array is completely sorted.



Time Complexity: O(N^2)

Auxiliary Space: O(1)

Linear Search:

Linear search is used to search a key element from multiple elements. Linear search is less used today because it is slower than binary search and hashing.

Algorithm:

Step 1: Traverse the array

Step 2: Match the key element with array element

Step 3: If key element is found, return the index position of the array element

Step 4: If key element is not found, return -1

Binary Search:

Binary search is used to search a key element from multiple elements. Binary search is faster than linear search.

In case of binary search, array elements must be in ascending order. If you have unsorted array, you can sort the array using Arrays.sort(arr) method.

Binary Search Algorithm: The basic steps to perform Binary Search are:

1. Sort the array in ascending order.
2. Set the low index to the first element of the array and the highindex to the last element.
3. Set the middle index to the average of the low and highindices.
4. If the element at the middle index is the target element, returnthe middle index.
5. If the target element is less than the element at the middleindex, set the high index to the middle index – 1.
6. If the target element is greater than the element at the middle index, set the low index to the middle index + 1.

Repeat steps 3-6 until the element is found or it is clear that the element is not present in the array.

Illustration