SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF INFORMATION TECHNOLOGY

PROGRAMMING FOR PROBLEM SOLVING

I YEAR - I SEM

UNIT 3 – ARRAYS AND STRINGS

TOPIC 6 – Searching and Sorting

BUBBLE SORT

- Bubble sort in C to arrange numbers in ascending order; you can modify it for descending order and can also sort strings. The bubble sort algorithm isn't efficient as its both average-case as well as worst-case complexity are O(n2).

- **Bubble sort algorithm**
- Start at index zero, compare the element with the next one (a[0] & a[1] (a is the name of the array)), and swap if a[0] > a[1]. Now compare a[1] & a[2] and swap if a[1] > a[2]. Repeat this process until the end of the array. After doing this, the largest element is present at the end. This whole thing is known as a pass. In the first pass, we process array elements from [0,n-1].

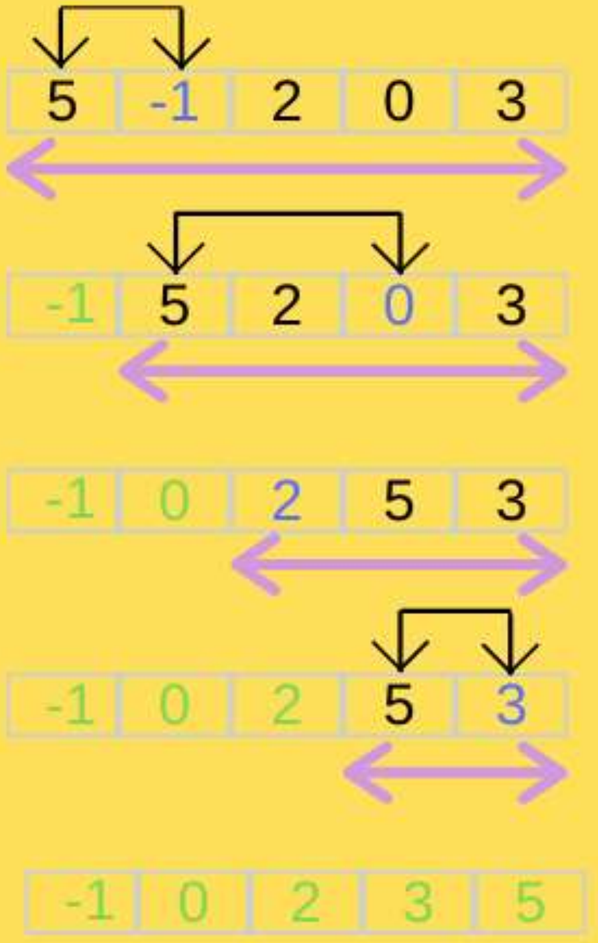
- Repeat step one but process array elements [0, n-2] because the last one, i.e., a[n-1], is present at its correct position. After this step, the largest two elements are present at the end.
- Repeat this process n-1 times.

SELECTION SORT

- Selection sort in C to sort numbers of an array in ascending order. With a little modification, it arranges numbers in descending order.

- **Selection sort algorithm (for ascending order)**
- Find the minimum element in the array and swap it with the element in the 1st position.
- Find the minimum element again in the remaining array[2, n] and swap it with the element at 2nd position, now we have two elements at their correct positions.
- We have to do this n-1 times to sort the array.

Selection Sort



Green = Sorted

Blue = Current minimum

Find minimum elements in unsorted array and swap if required (element not at correct location already).

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.
- **Algorithm**
- To sort an array of size n in ascending order:
- 1: Iterate from arr[1] to arr[n] over the array.
- 2: Compare the current element (key) to its predecessor.
- 3: If the key element is smaller than its predecessor, compare it to the elements before. Move the greater elements one position up to make space for the swapped element.

Insertion Sort Execution Example