DAA Lab 1: Insertion and Selection sort

### Meet Jain, 23BCP093

**Theory**

**Selection Sort:**

In selection sort, the smallest value among the unsorted elements of the array is selected in every pass and inserted to its appropriate position into the array. It is also the simplest algorithm. It is an in-place comparison sorting algorithm. In this algorithm, the array is divided into two parts, first is sorted part, and another one is the unsorted part. Initially, the sorted part of the array is empty, and unsorted part is the given array. Sorted part is placed at the left, while the unsorted part is placed at the right.

In selection sort, the first smallest element is selected from the unsorted array and placed at the first position. After that second smallest element is selected and placed in the second position. The process continues until the array is entirely sorted.

**Insertion Sort:**

The Insertion Sort algorithm uses one part of the array to hold the sorted values, and the other part of the array to hold values that are not sorted yet. The algorithm takes one value at a time from the unsorted part of the array and puts it into the right place in the sorted part of the array, until the array is sorted.

**Code**

#include <bits/stdc++.h>

using namespace std;

void printarr(int arr[],int n){

    for(int i =0;i<n;i++){

        cout << arr[i] <<" ";

    }

    cout << endl;

}

int **selection\_sort**(int arr[],int n){

    for(int i = 0;i<n;i++){

        int min = arr[i];

        //finding the minimum in further part of the array and swapping it

        for(int j = i+1;j < n;j++){

            if(arr[j] < min){

                int temp = arr[j];

                arr[j] = arr[i];

                arr[i] = temp;

                min = arr[i];

            }

        }

    }

    printarr(arr,n);

}

int **insertion\_sort**(int arr[], int n){

    for(int i = 1;i<n;i++){

        int j = i;

        //swap until the element reaches its correct index in sorted part of

        while(j>0){

            if(arr[j-1] > arr[j]){

                int temp = arr[j];

                arr[j] = arr [j-1];

                arr[j-1] = temp;

            }

            j--;

        }

    }

    printarr(arr,n);

}

int main(){

    int arr1[5] = {5,42,200,29,1};

    int arr2[5] = {51,42,20,29,1};

    int n=5;

    insertion\_sort(arr1,n);

    selection\_sort(arr2,n);

}

**Output**

**A black screen with orange text

Description automatically generated**

**Analysis**

A close-up of a notebook

Description automatically generatedA piece of paper with writing on it

Description automatically generated