INDIAN INSTITUTE OF INFORMATION TECHNOLOGY , BHOPAL

Design and Analysis of Algorithm Lab

Name: Tanav Singh Bajaj

Scholar No. : 20U03037

Semester: Third

Branch: IT

Lab Coordinator: Dr. Yatendra Sahu

ASSIGNMENT-1

Q1. Explain the following sorting algorithm:-

1. Selection Sort
2. Bubble Sort

1. Selection Sort-

Selection sorting is a simple sorting algorithm. This sorting algorithm is an in-place comparison-based algorithm in which the list is divided into two parts, the sorted part at the left end and the unsorted part at the right end. Initially, the sorted part is empty and the unsorted part is the entire list.

The smallest element is selected from the unsorted array and swapped with the leftmost element, and that element becomes a part of the sorted array. This process continues moving an unsorted array boundary by one element to the right.

2. Bubble Sort-

Bubble sort is a simple sorting algorithm. This sorting algorithm is a comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order.

Q2. Write down the algorithm for both.

Algorithm for Selection Sort:-

Step 1 – Set MIN to location 0.
Step 2 – Search the minimum element in the list.
Step 3 – Swap with value at location MIN.
Step 4 – Increment MIN to point to the next element.
Step 5 – Repeat until list is sorted.

Algorithm for Bubble Sort:-

1. algorithm Bubble_Sort(list)
2. Pre: list != fi
3. Post: list is sorted in ascending order for all values
4. for i <- 0 to list:Count - 1
5. for j <- 0 to list:Count - 1
6. if list[i] < list[j]
7. Swap(list[i]; list[j])
8. end if
9. end for
10. end for
11. return list
12. end Bubble_Sort

Q3. Evaluate space and time complexity.

Bubble sort-

- Worst Case Time Complexity is: $O(N_2)$
- Average Case Time Complexity is: $O(N_2)$
- Best Case Time Complexity is: O(N)
- Space Complexity: O(1)

Selection sort-

- Worst Case Time Complexity is: $O(N_2)$
- Average Case Time Complexity is: $O(N_2)$
- Best Case Time Complexity is: $O(N_2)$
- Space Complexity: O(1)

Q4. Write down the code and attach ss of output.

Selection sort-

```
#include <iostream>
#include <bits/stdc++.h>

using namespace std;

int main()
{
  int n;
  cout << "Enter the value of n" << endl;
  cin >> n;
  int arr[n];
  cout << "Enter the elements" << endl;
  for (int i = 0; i < n; i++)
  {
     cin >> arr[i];
  }

  cout << "unsorted array is ";
  for (int i = 0; i < n; i++)
  {
```

```
    cout << arr[i] << " ";
  }
  cout << endl;
  int sm, pos, t;
  for (int i = 0; i < n - 1; i++)
  {
    sm = arr[i];
    pos = i;
    for (int j = i + 1; j < n; j++)
    {
      if (arr[j] < sm)
      {
        sm = arr[j];
        pos = j;
      }
    }
    t = arr[pos];
    arr[pos] = arr[i];
    arr[i] = t;
  }
  cout << "sorted array is ";
  for (int i = 0; i < n; i++)
  {
    cout << arr[i] << " ";
  }
  cout << endl;
  return 0;
}
```

Output:

```
Enter the value of n
4
Enter the elements
4
5
63
4
unsorted array is 4 5 63 4
sorted array is 4 4 5 63
```

Bubble sort-

```
#include <iostream>
#include <bits/stdc++.h>

using namespace std;

int main()
{
  int n;
  cout << "Enter the value of n" << endl;
  cin >> n;
  int arr[n];
  int t;
  cout << "Enter the value of elements" << endl;
  for (int i = 0; i < n; i++)
```

```
  {
     cin >> arr[i];
  }
  cout << "unsorted array is ";
  for (int i = 0; i < n; i++)
  {
     cout << arr[i] << " ";
  }
  cout << endl;
  for (int i = 0; i < n; i++)
  {
     for (int j = 0; j < n - i - 1; j++)
     {
        if (arr[j + 1] < arr[j])
        {
           t = arr[j + 1];
           arr[j + 1] = arr[j];
           arr[j] = t;
        }
     }
  }
  cout << "sorted array is ";
  for (int i = 0; i < n; i++)
  {
     cout << arr[i] << " ";
  }
  cout << endl;
  return 0;
}
```

Output:

```
Enter the value of n
5
Enter the value of elements
1 4 7 2 9
unsorted array is 1 4 7 2 9
sorted array is 1 2 4 7 9
```