CSS1051: Advanced Computing Lab - 1

Topic: Data Structures and Algorithms

Instruction: Download the docx file, type in your response in the space provided, save in pdf and then upload the file.

Assignment 3: Assume that only three sorting algorithms namely, Insertion sort, Bubble sort and Selection sort exist in the universe of comparison based sorting (i.e, there is no Quick sort, Heap sort, Merge sort etc..). Design a hybrid sort by appropriately selecting two out of these three sorting algorithms to complement the advantages and disadvantages of these individual algorithms. Your objective would be to ensure that for all possible practical instances, your hybrid algorithm should perform (in terms of fundamental arithmetic and logical operations such as number of comparisons and total number of swaps) at least no worse than standalone application of any of these three algorithms.

- **Q1:** Write a clean pseudocode and explain the different steps of your algorithm.

Your response:

```
Hybrid sort using Bubble sort and Selection sort :

Initialize an empty array arr []
Num <- - size of the array
Store the elements in the defined array

Call the bubble_sort (arr) and store in variable result
```

```
def bubble_sort(A):
    for i in range(num):
        swapped = False
        for j in range(0, num - i - 1):
           if arr[j] > arr[j + 1]:

                swapped = True
                break
        break

    return swapped
```

```
if result = False:
     The array is already in sorted order and return sorted array
and
     Number of comparison = num – 1
     Number of swapping = 0
```

Else:

Call the selection_sort (arr) and it will sort the entered array and return the

Number of comparison = comp_count

Number of swapping = swap_count

```
def selection_sort(A):
    n = len(A)
    swap_count = comp_count = 0
    for i in range(n - 1):
        min_idx = i
        for j in range(i + 1, n):
            if A[min_idx] > A[j]:
                min_idx = j

            comp_count += 1
        if i != min_idx:
            A[i], A[min_idx] = A[min_idx], A[i]
            swap_count += 1
    print('The array after sorting : \n', A)
    print('Number of comparison : \n', comp_count)
    print('Number of swapping : \n', swap_count)
```

- **Q2:** Briefly explain the key points of your algorithm that ensure fulfilment of the primary objective as mentioned in the problem statement above).

Your response:

In our hybrid sorting technique (We use Bubble sort and Selection sort)

First we check that the given array is already sorted or not. For this, we use bubble sort technique to ensure that it will happen in O(n) time with zero swapping and (n – 1) comparison.

If our array is already sorted, then our algorithm will return the sorted array in O(n) with zero swapping and it will do (n – 1) comparison.

If the given array is not in sorted order, then we call selection_sort(arr). Here we use selection sort technique because in the selection sort, the number of swapping is atmost the number of elements in the array i.e O(n). And it will sort our array in O(n) swapping with the same number of comparisons as other sorting techniques do (insertion sort or bubble sort)

So our hybrid sorting technique will sort the given array

Best case : When array is already sorted
    Using Bubble sort,
    Number of swapping : 0
    Number of Comparison : n – 1

Average case and Worst case:
    Using Selection sort,
    Number of swapping : O(n)
    Number of comparison : O(n^2)

- **Q3:** Write a program (preferably in python) to implement your algorithm. Apart from the sorted array, your program should also output the total number of comparisons and the total number of swaps done. Provide the screenshot of the input output for few sample input instances. Your sample inputs must contain boundary cases (such as already sorted in desired order, already sorted in reverse order, almost sorted, etc.) Briefly explain the key points of your algorithm that ensure fulfilment of the primary objective as mentioned in the problem statement above).
  [Upload the source code of your program along with this docx file]

  Your response:

  Already Sorted:

```
C:\Users\Rohit\PycharmProjects\Code\venv\Scripts\python.exe C:/Users/Rohit/PycharmProjects/Code/Hybrid_Sorting.py
Enter the number of elements in the list : 5
Enter the elements in the list :
1
2
3
4
5
The entered array is already sorted :

[1, 2, 3, 4, 5]
Number of comparison :  4
Number of swapping :  0
```

  Sorted in Reverse Order:

```
C:\Users\Rohit\PycharmProjects\Code\venv\Scripts\python.exe C:/Users/Rohit/PycharmProjects/Code/Hybrid_Sorting.py
Enter the number of elements in the list : 5
Enter the elements in the list :
5
4
3
2
1
The array after sorting :
 [1, 2, 3, 4, 5]
Number of comparison :
 10
Number of swapping :
 2
```

Almost Sorted:

```
C:\Users\Rohit\PycharmProjects\Code\venv\Scripts\python.exe C:/Users/Rohit/PycharmProjects/Code/Hybrid_Sorting.py
Enter the number of elements in the list : 5
Enter the elements in the list :
1
2
3
6
5
The array after sorting :
 [1, 2, 3, 5, 6]
Number of comparison :
 10
Number of swapping :
 1
```


Honour pledge:

- The responses of the questions presents above are done by myself.
- I haven't taken any help from any other students of the class or any other person.

- I haven’t referred any book, web resource or any other study materials (available offline or online).
- I haven’t helped any other students in completing the assignment.

Submitted by:

Name:
Rohit Kumar

Roll No. 22CS4112
