Design and Analysis of Algorithm- 5311

Assignment 01

Name: Gaurav Sanjeev Taneja

Student Id: 1001955801

1. You have an array containing integers from 1 to 10 (not in order) but one number is missing (there is 9 numbers in the array).

a) write a pseudo code to find the missing number (10 points).

```
arr=[10,9,8,5,4,3,2,1,7]
n = length(arr)
total = (n+1 * n+2)/2
sum = 0
for i in range(start, end):
    sum += arr[i]
missing_value= total - sum
return missing_value # output 6
```

b) what is the worst case run time complexity of your suggested solution.
The worst case run time of the above algorithm is O(n)

2. You are given an array of integers:

a) write a pseudo code to find all pairs of numbers whose sum is equal to a particular number.

```
arr = [1,4,3,7,6]
sumToCompare = 7
for i in range(start, end):
if element of the array is less than compared length
for j in range(i,end):
if arr[i] + arr[j] == sumToCompare then
print(arr[i],arr[j])
```

b) what is the worst case run time complexity of your suggested?
The worst case run time of the above algorithm is O(n^2)

3. You are given an array of integers:

a. write a pseudo code to remove duplicates from your array

```
suppose arr = [1,4,9,2,2,3,4,5,6,6]
return list(set(arr))
```

b. what is the worst case run time complexity of your suggested solution?
The worst case run time of the above algorithm = O(n)

4. you are given 2 sorted arrays:

a. write a pseudo code to find the median of the two sorted arrays (combined)

```
arrOne = [1,3,5,7]
arrTwo = [2,4,6,8]
take two sorted array arrayOne and arrayTwo
mergedArray = arrayOne + arrayTwo
Now to sort the Merged array
declare two variables left and right
intialize left to 0 and right to n-1
calculate mid = left (right - left) // 2
call megerSort using (left, mid) and (mid + 1, right)
repeat step until left < right
then call merge (arr, left, mid,right)
return the list after merging # Output = [1,2,3,4,5,6,7,8]
Now divide index= lenght(sortedMegedList) // 2
print(sortedMegedList[index]) # median of merged list '4'
```

b. what is the worst case run time complexity of your suggested solution

The worst-case runtime complexity :O (n log n)

5. Use one of the sorting algorithms to sort the following array input = {4, 1, 2, 7, 10, 1, 2, 4, 4, 7, 1, 2, 1, 10, 1, 2, 4, 1, 2, 7, 10, 1, 2} Show step by step what happens for the input (no pseudo code is required).

We'll assume i as an index.
We can sort the above array using Insertion sort.
The iterations are:

For i=0: {1,4,2,7,10,1,2,4,4,7,1,2,1,10,1,2,4,1,2,7,10,1,2}
For i=1: {1,4,2,7,10,1,2,4,4,7,1,2,1,10,1,2,4,1,2,7,10,1,2}
For i=2: {1,2,4,7,10,1,2,4,4,7,1,2,1,10,1,2,4,1,2,7,10,1,2}
For i=3: {1,2,4,7,10,1,2,4,4,7,1,2,1,10,1,2,4,1,2,7,10,1,2}
For i=4: {1,2,4,7,10,1,2,4,4,7,1,2,1,10,1,2,4,1,2,7,10,1,2}
For i=5: {1,1,2,4,7,10,2,4,4,7,1,2,1,10,1,2,4,1,2,7,10,1,2}
For i=6: {1,1,2,2,4,7,10,4,4,7,1,2,1,10,1,2,4,1,2,7,10,1,2}
For i=7: {1,1,2,2,4,4,7,10,4,7,1,2,1,10,1,2,4,1,2,7,10,1,2}
For i=8: {1,1,2,2,4,4,4,7,10,7,1,2,1,10,1,2,4,1,2,7,10,1,2}
For i=9: {1,1,2,2,4,4,4,7,7,10,1,2,1,10,1,2,4,1,2,7,10,1,2}
For i=10: {1,1,1,2,2,4,4,4,7,7,10,2,1,10,1,2,4,1,2,7,10,1,2}
For i=11: {1,1,1,2,2,2,4,4,4,7,7,10,1,10,1,2,4,1,2,7,10,1,2}
For i=12: {1,1,1,1,2,2,2,4,4,4,7,7,10,10,1,2,4,1,2,7,10,1,2}
For i=13: {1,1,1,1,2,2,2,4,4,4,7,7,10,10,1,2,4,1,2,7,10,1,2}
For i=14: {1,1,1,1,1,2,2,2,4,4,4,7,7,10,10,2,4,1,2,7,10,1,2}
For i=15: {1,1,1,1,1,2,2,2,2,4,4,4,7,7,10,10,4,1,2,7,10,1,2}
For i=16: {1,1,1,1,1,2,2,2,2,4,4,4,4,7,7,10,10,1,2,7,10,1,2}
For i=17: {1,1,1,1,1,1,2,2,2,2,4,4,4,4,7,7,10,10,2,7,10,1,2}
For i=18: {1,1,1,1,1,1,2,2,2,2,2,4,4,4,4,7,7,10,10,7,10,1,2}
For i=19: {1,1,1,1,1,1,2,2,2,2,2,4,4,4,4,7,7,7,10,10,10,1,2}
For i=20: {1,1,1,1,1,1,2,2,2,2,2,4,4,4,4,7,7,7,10,10,10,1,2}
For i=21: {1,1,1,1,1,1,1,2,2,2,2,2,4,4,4,4,7,7,7,10,10,10,2}
For i=22: {1,1,1,1,1,1,1,2,2,2,2,2,2,4,4,4,4,7,7,7,10,10,10}

6. Answer the following questions:

   a. When does the worst case of Quicksort happen and what is the worst-case run time complexity in terms of big O? (5 points)
   An array is sorted in ascending or descending order it chooses the pivot element that divides the where one array has no elements, and the other array has n-1 elements.
   Also, when all the elements of the array are the same the pivot element divides into the n-1 array.
   The worst-case time complexity is O(n^2).

   b. When does the best case of bubble sort happen and what is the best-case run time complexity in terms of big O? (5 points)
   The best-case scenario is when the array is already sorted. The time complexity is O(n).

   c. What is the runtime complexity of Insertion sort in all 3 cases? Explain the situation which results in best, average, and worst-case complexity. (10 points)
   I. Best Case:
   The best-case time complexity is O(n). This happens only when the array input is sorted in ascending order.
   arrayInput = [1,2,3,4,5]
   II. Average Case:
   The time complexity is O(n^2). In this case,array is sorted and compared using the key element if less than key-value places before key element else after the key element.
   arrayInput = [1,3,2,6,4]
   III. Worst Case:
   The time complexity is O(n^2). This happens when the array is sorted in descending order. Number of iterations needed is n-1 for sorting and swapping.
   arrayInput = [9,8,7,6,5,4]