22AIE212-Design and Analysis of Algorithms

Lab Sheet 2

Iterative programs-Part 2

Name : Anuvind M P

Roll no: AM.EN.U4AIE22010

---

1. You are given a sorted array A of size n. Write an iterative program to remove the duplicates from the array. For example, if A[] = {2, 7, 7, 11, 24, 24, 24, 29, 36, 36}, your output should be B[] = {2, 7, 11, 24, 29, 36}.
    a. Count the operations to get the closed-form equation of running time (worst case).
    b. Submit the program for the problem https://leetcode.com/problems/remove-duplicates-from-sorted-array/ and submit the snapshot of acceptance as proof.
    c. What is the time complexity?

CODE :

```
def removeDuplicates(nums):

        a = []
        for i in range(len(nums)):
            if nums[i] != nums[i-1]:
                a.append(nums[i])
        nums = a
        return nums

A = [2, 7, 7, 11, 24, 24, 24, 29, 36, 36]
print(removeDuplicates(A))
```

OUTPUT :

```
...   [2, 7, 11, 24, 29, 36]
```

a.

```
def removeDuplicates(nums):

        a = []
        for i in range(len(nums)):
            if nums[i] != nums[i-1]:
                a.append(nums[i])
        nums = a
        return nums
```

T(n)

1

n

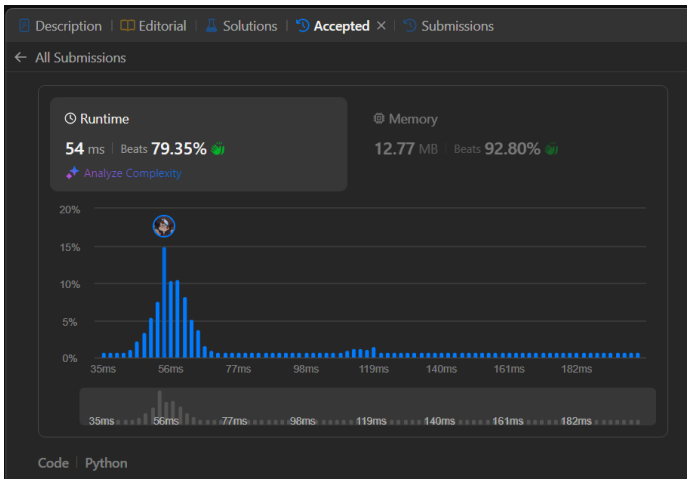
1

➔ T(n)=1+ n +1 = n+3

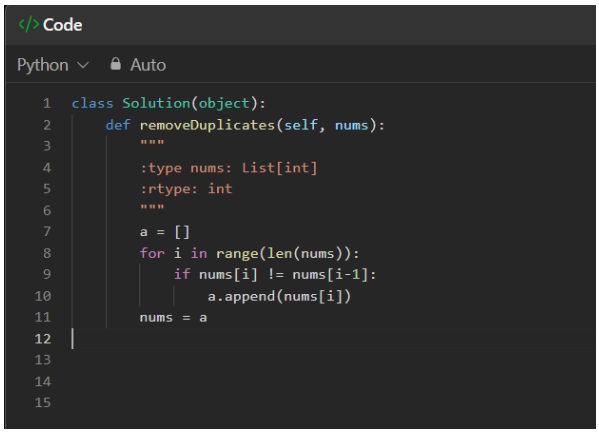
T(n) = O(n)

b.

Leetcode :



Leetcode :



c. Time Complexity = O(n)

2. Consider an array A of size n. Split A[] into the two arrays Low[] and High[] such that Low[] contains all elements < A[0] and High[] contains all elements >= A[0].
    a. Write an iterative algorithm and implement it.
    b. What is the time complexity?

CODE :

```
def splitArray(nums):
    Low, High, chk= [], [], nums[0]
    for i in nums:
        if i < chk:
            Low.append(i)
        else:
            High.append(i)
    return Low,High

nums = [5,7,2,3,5,1,2,9,3]
a,b = splitArray(nums)
print(f"low : {a} \nhigh : {b}")
```

OUTPUT :

```
array : [5, 7, 2, 3, 5, 1, 2, 9, 3]
low : [2, 3, 1, 2, 3]
high : [5, 7, 5, 9]
```

Time complexity : O(n)

3. Given two sorted lists A[1..n] and B[1..n], write an algorithm to merge them into a single sorted list C[1..2n]. For example, if A[] = {1,3,6,7} and B[] = {2,4,5,8}, then C[] = {1,2,3,4,5,6,7,8}.
    a. Find the complexity
    b. Submit the program for the problem https://leetcode.com/problems/merge-two-sorted-lists/ and submit the snapshot of acceptance as proof

CODE :

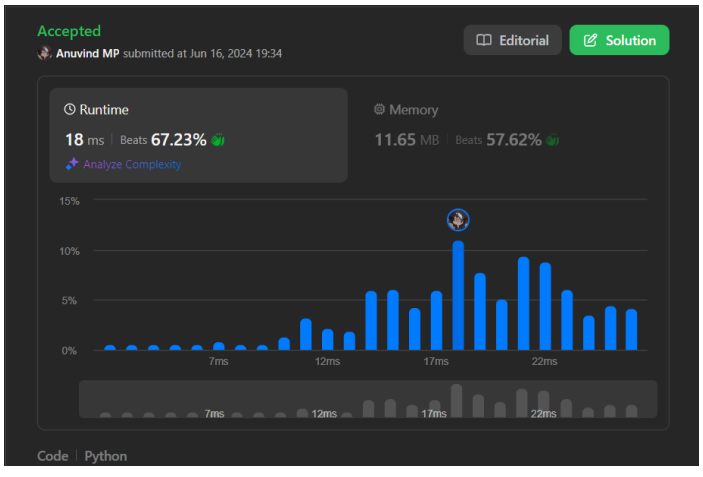
```python
def merge(A, B):
    res = []
    i,j = 0, 0
    while i < len(A) and j < len(B):
        if A[i] < B[j]:
            res.append(A[i])
            i+=1
        else:
            res.append(B[j])
            j+=1
    if i<len(A):
        res+=A[i:]
    if j<len(B):
        res+=B[j:]

    return res

A = [1,3,6,7]
B = [2,4,5,8]
print(merge(A,B))
```

OUTPUT :

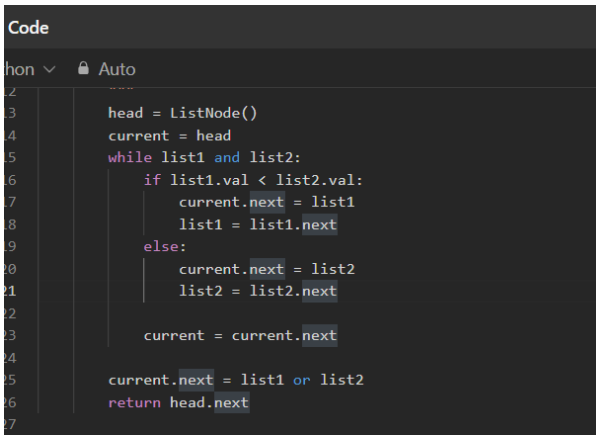
```
[35]  ✓ 0.0s
...   [1, 2, 3, 4, 5, 6, 7, 8]
```

- ⇨ **Time complexity : O(n)**
- ⇨ **n = len(A) + len(B)**

Leetcode :



Leetcode :



4. There is a class with m students and n exams. You are given a 0-indexed m x n integer matrix called score, where score[i][j] denotes the score the ith student got in the jth exam. The matrix score contains distinct integers only. You are also given an integer k. Sort the students (i.e., the rows of the matrix) by their scores in the kth (0-indexed) exam from the highest to the lowest. Return the matrix after sorting it.
    a. Find the time complexity
    b. Submit the program for the problem https://leetcode.com/problems/sort-the-students-by-their-kth-score/description/ and submit the snapshot of acceptance as proof.

CODE :

```
def sortTheStudents(score, k):
        output = []
        x = [i[k] for i in score]

        graph = {}
        j = 0
        for i in x:
            graph[i] = j
            j+=1

        x.sort(reverse = True)

        for i in x:
            key = graph[i]
            output.append(score[key])

        return output

score = [[10,6,9,1],[7,5,11,2],[4,8,3,15]]
print("Before sorting : ",score)
print("\nAfter sorting :
",sortTheStudents(score,2))
```

OUTPUT :

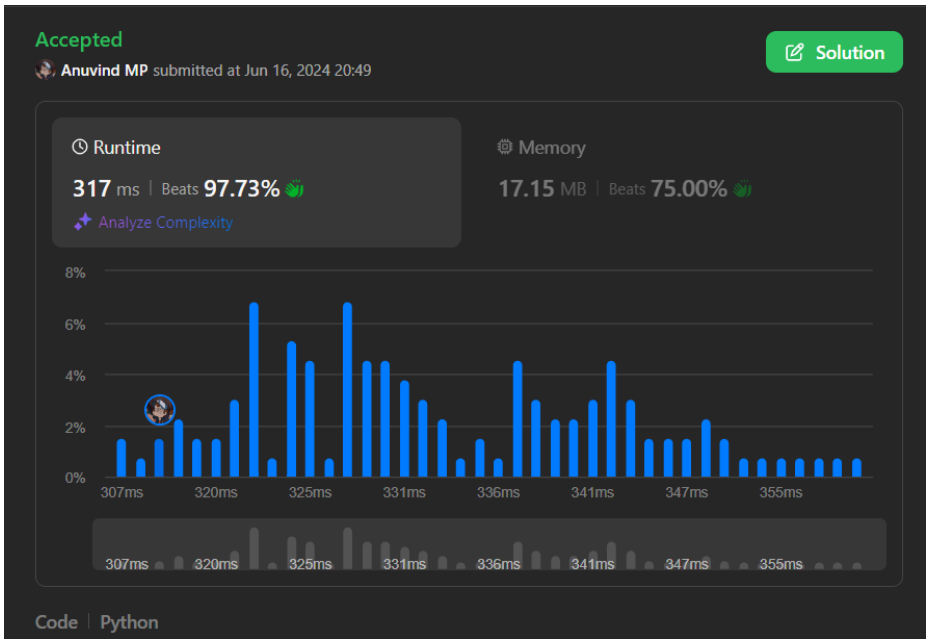
```
Before sorting :  [[10, 6, 9, 1], [7, 5, 11, 2], [4, 8, 3, 15]]

After sorting :  [[7, 5, 11, 2], [10, 6, 9, 1], [4, 8, 3, 15]]
```

Time Complexity :

⇨ **T(n) = n + n + nlogn + n = O(nlogn)**

⇨ *Sorting takes nlogn time*

Leetcode :



Leetcode :

Code

Python | Auto

```python
class Solution(object):
    def sortTheStudents(self, score, k):
        """
        :type score: List[List[int]]
        :type k: int
        :rtype: List[List[int]]
        """
        output = []
        x = [i[k] for i in score]

        graph = {}
        j = 0
        for i in x:
            graph[i] = j
            j+=1

        x.sort(reverse = True)

        for i in x:
            key = graph[i]
            output.append(score[key])

        return output
```