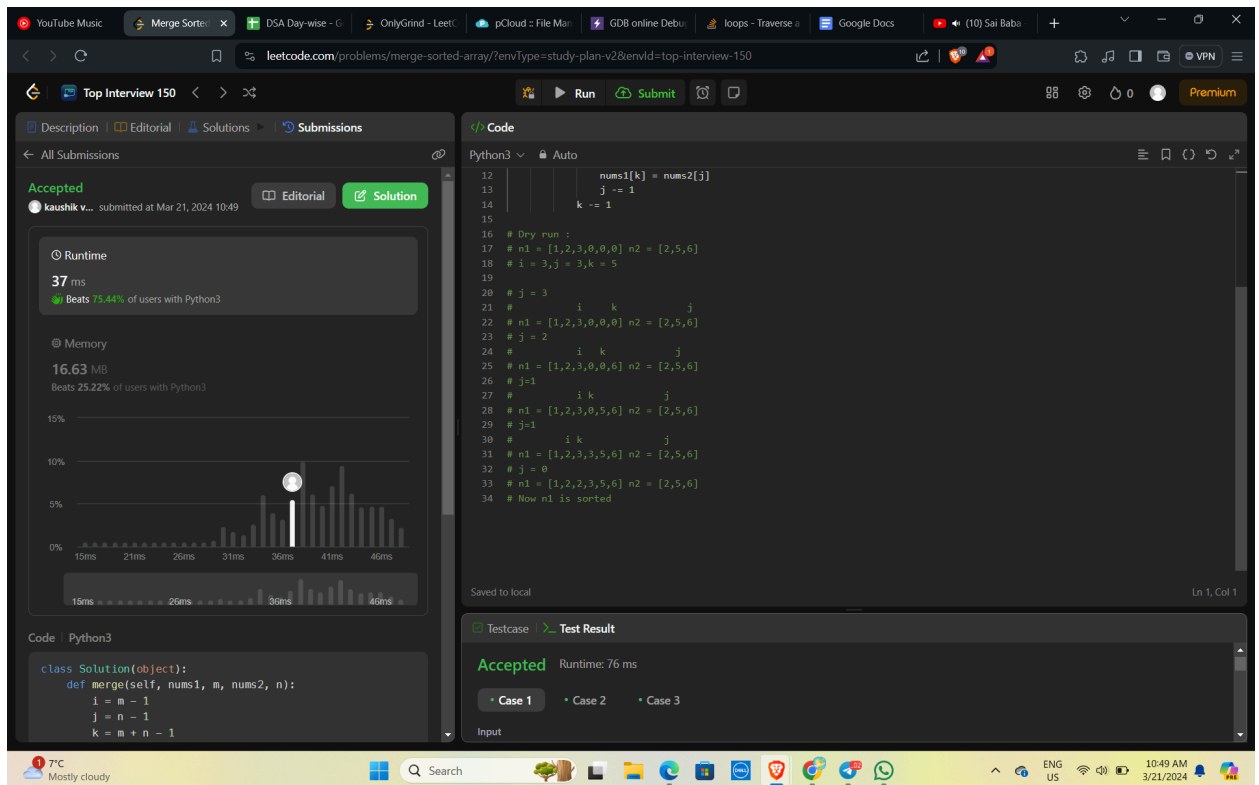


Day45 - March 21st 2024

## 1. Started my day by solving leetcode problems



The screenshot shows a LeetCode submission for the "Merge Sorted Array" problem. The submission is accepted, with a runtime of 37 ms and memory usage of 16.63 MB. The code is in Python3 and uses a two-pointer approach to merge two sorted arrays.

**Runtime:** 37 ms  
Beats 75.44% of users with Python3

**Memory:** 16.63 MB  
Beats 25.22% of users with Python3

**Code (Python3):**

```
class Solution(object):
    def merge(self, nums1, m, nums2, n):
        i = m - 1
        j = n - 1
        k = m + n - 1
```

**Testcase:** Accepted  
Runtime: 76 ms

**Input:**

```
nums1 = [1,2,3,0,0,0] m = 3
nums2 = [2,5,6] n = 3
```

YouTube Music Merge Sorted Merge Sorted DSA Day-wise OnlyGrind - Le pCloud - File M GDB online De loops - Travers Google Docs (10) Sai Bai

leetcode.com/problems/merge-sorted-array/solutions/4906749/python-solution-with-dry-run-included-in-code/

Problem List > > > > >

Description Editorial Solutions Submissions

All Solutions

Python Solution with dry run included(in code).

OnlyGrind  
0 a few seconds ago

Python3 Two Pointers

### Complexity

- Time complexity:  $O(m+n)$
- Space complexity:  $O(1)$

### Code

```
class Solution(object):
    def merge(self, nums1, m, nums2, n):
        i = m - 1
        j = n - 1
        k = m + n - 1

        while j >= 0:
            if i >= 0 and nums1[i] > nums2[j]:
                nums1[k] = nums1[i]
                i -= 1
            else:
                nums1[k] = nums2[j]
                j -= 1
            k -= 1

# Dry run :
# n1 = [1,2,3,0,0,0] n2 = [2,5,6]
```

Python3 Auto

```
1 class Solution(object):
2     def merge(self, nums1, m, nums2, n):
3         i = m - 1
4         j = n - 1
5         k = m + n - 1
6
7         while j >= 0:
8             if i >= 0 and nums1[i] > nums2[j]:
9                 nums1[k] = nums1[i]
10                i -= 1
11            else:
12                nums1[k] = nums2[j]
13                j -= 1
14            k -= 1
15
16 # Dry run :
17 # n1 = [1,2,3,0,0,0] n2 = [2,5,6]
18 # i = 3, j = 3, k = 5
19
20 # j = 3
21 # i k j
22 # n1 = [1,2,3,0,0,0] n2 = [2,5,6]
23 # j = 2
24 # i k j
25 # n1 = [1,2,3,0,0,0] n2 = [2,5,6]
26 # j=1
27 # i k j
```

Saved to local Ln 1, Col 1

Testcase Test Result

Case 1 Case 2 Case 3 +

nums1 =

</> Source

7°C Mostly cloudy

YouTube Music Remove Element Merge Sorted Arr DSA Day-wise - G OnlyGrind - Leet pCloud - File Mar GDB online Debu loops - Traverse Google Docs

leetcode.com/problems/remove-element/?envType=study-plan-v2&envId=top-interview-150

Top Interview 150 > > > > >

Description Editorial Solutions Submissions

All Submissions

Accepted

kaushik varma N submitted at Mar 21, 2024 11:12

Runtime 39 ms Beats 45.94% of users with Python3

Memory 16.70 MB Beats 11.30% of users with Python3

Code Python3

```
class Solution:
    def removeElement(self, nums: List[int], val: int) -> int:
        k = 0

        for i in nums:
            if i != val:
                nums[k] = i
                k += 1

        return k
```

Saved to local Ln 16, Col 1

Testcase Test Result

Accepted Runtime: 61 ms

Case 1 Case 2

Input

DOW +0.88%

- Find the DSA google doc for 21st : [DSA\\_Arrays\\_03/21/2024](#)
- Spark theory is in process : [SparkTH Day8 03/21/2024](#)

#### 4. Ended my day by solving complex SQL questions from data lemur

The screenshot shows the DataLemur profile page for user kaushik varma. The page includes a profile section with a yellow circular avatar containing 'KV' and the name 'kaushik varma'. To the right, the 'Problems Solved' section shows a star icon with the number 8, and a breakdown of problems solved by difficulty: Easy (3/70), Medium (3/85), and Hard (2/47). Below this is a calendar view for the last year (May to March) showing the number of questions solved per day. A tooltip indicates '4 questions solved at Thu Mar 21 2024'. At the bottom, there is a referral code 'datalemur.com?referralCode=NSm2kJIS' and a 'Copy To Clipboard' button.

The screenshot shows the DataLemur question page for 'Highest-Grossing Items [Amazon SQL Interview Question]'. The page is titled 'Highest-Grossing Items [Amazon SQL Interview Question]' and is categorized as 'Medium' difficulty. It includes a description of the problem, a table schema for 'product\_spend', an example input table, and a code editor for writing SQL queries.

**product\_spend Table:**

Column Name	Type
category	string
product	string
user_id	integer
spend	decimal
transaction_date	timestamp

**product\_spend Example Input:**

category	product	user_id	spend	transaction_date
appliance	refrigerator	165	246.00	12/26/2021 12:00:00
appliance	refrigerator	123	299.99	03/02/2022 12:00:00
appliance	washing machine	123	219.80	03/02/2022 12:00:00

The code editor shows two SQL queries. The first query uses window functions to rank products by spend within each category for the year 2022. The second query uses a similar approach but includes a row number to handle ties.

```
1 with cte as(
2   SELECT *,rank() over(PARTITION BY category,product order by spend desc) as rank1 FROM pi
3   ,cte2 as(
4     select category,product,sum(spend) as spend from cte
5     where EXTRACT(year from transaction_date) = '2022'
6     group by category,product),
7   cte3 as(
8     select *,rank() over(partition by category order by spend desc) as rank2 from cte2)
9   select category,product,spend
10  from cte3
11  where rank2 <= 2
12
13  --select * from product_spend
14
15
16
17
18
19
20
21 with cte as(
22   SELECT category,product,SUM(spend) as total_spend,row_number() over(PARTITION BY category,product
23   from product_spend
24   where EXTRACT(year from transaction_date) = '2022'
25   GROUP BY category,product)
26   select category,product,total_spend
27   from cte
28   where rn <= '2'
29   --order by category,s desc
30
31
32
33
```