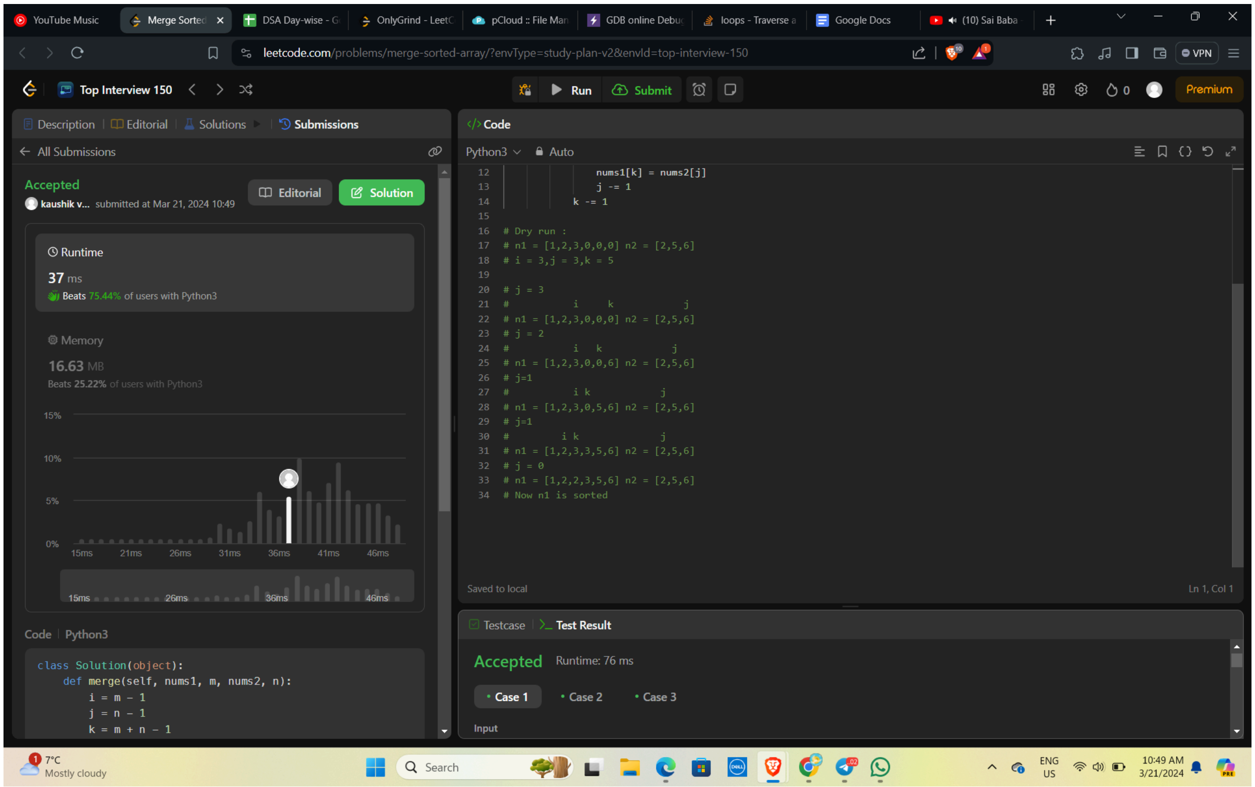
Day45 - March 21st 2024

1. Started my day by solving leetcode problems



Accepted
kaushik v... submitted at Mar 21, 2024 10:49

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
```

```
                nums1[k] = nums2[j]
                j -= 1
            k -= 1

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

# j = 3
#           i     k             j
# n1 = [1,2,3,0,0,0] n2 = [2,5,6]
# j = 2
#           i   k             j
# n1 = [1,2,3,0,0,6] n2 = [2,5,6]
# j=1
#           i k             j
# n1 = [1,2,3,0,5,6] n2 = [2,5,6]
# j=1
#         i k               j
# n1 = [1,2,3,3,5,6] n2 = [2,5,6]
# j = 0
# n1 = [1,2,2,3,5,6] n2 = [2,5,6]
# Now n1 is sorted
```

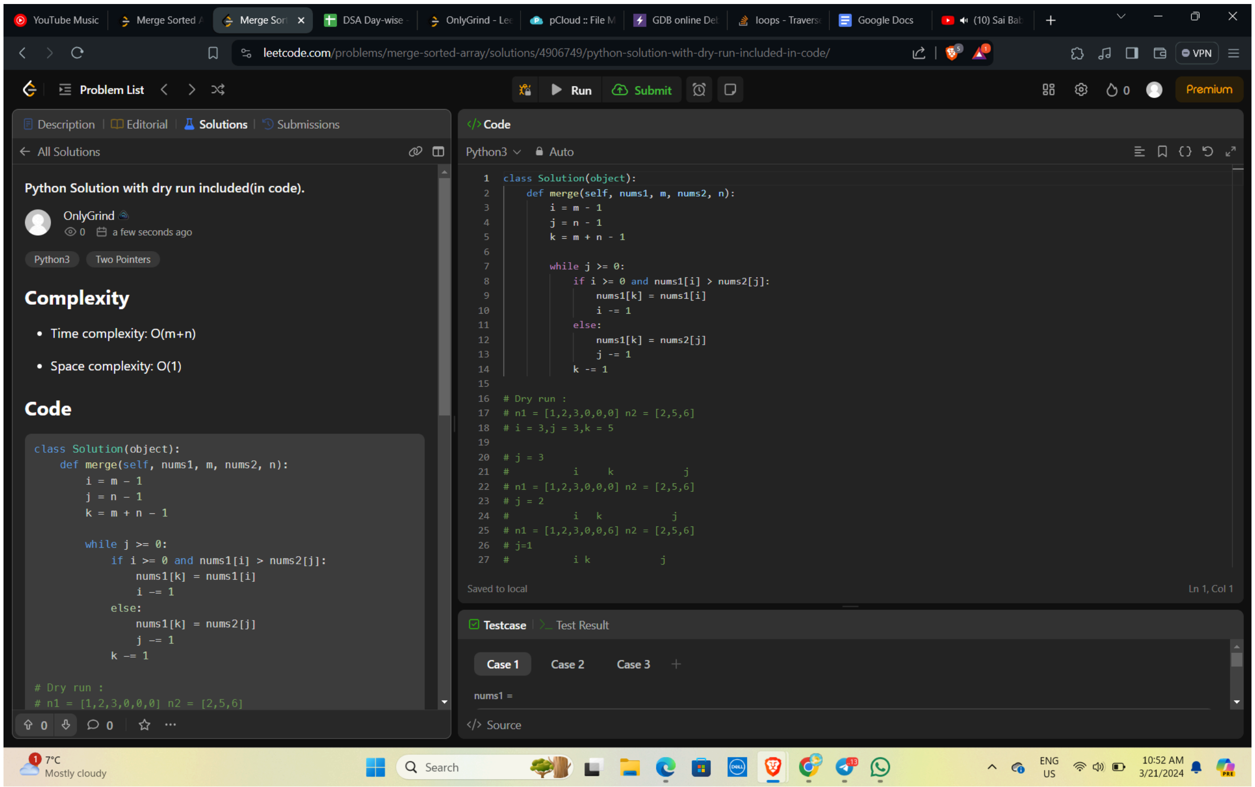
Saved to local

Ln 1, Col 1

Accepted Runtime: 76 ms

Case 1 · Case 2 · Case 3

Input



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

Python Solution with dry run included(in code).

OnlyGrind

Python3 · Two Pointers

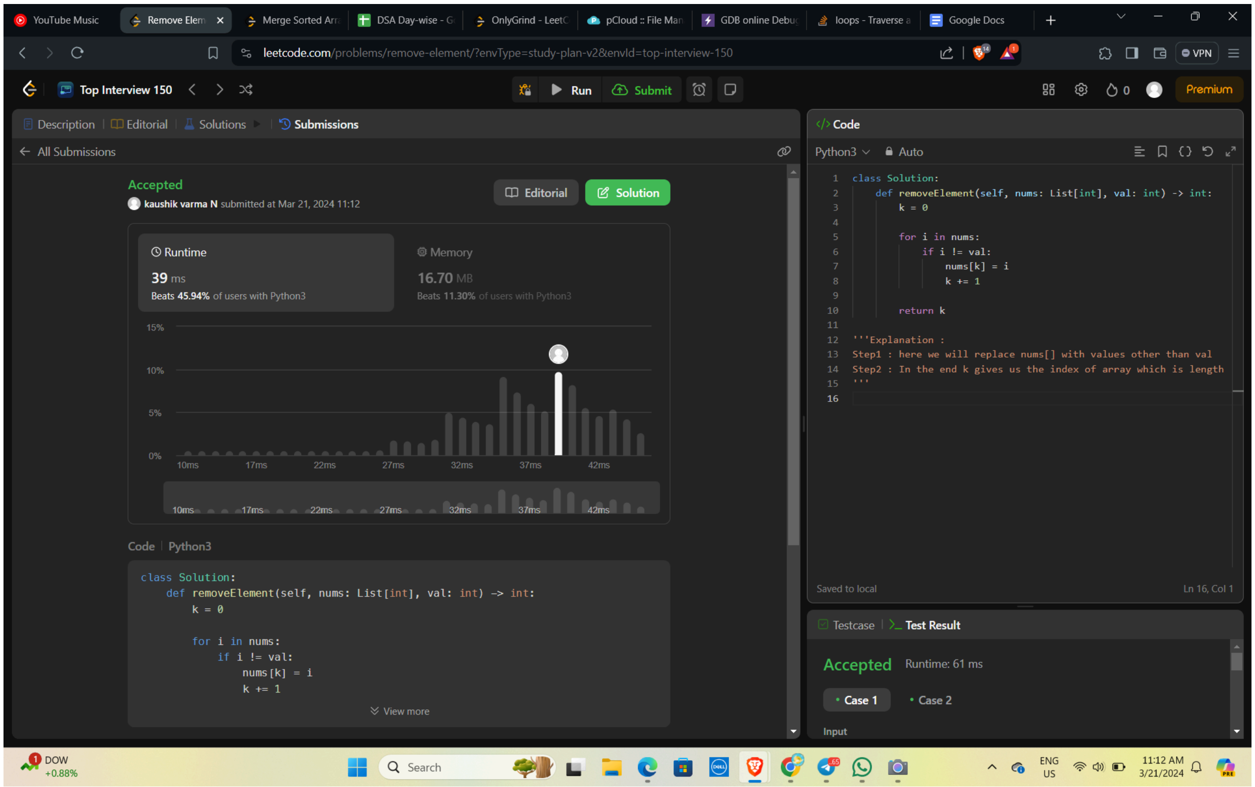
Complexity

- 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]
# i = 3,j = 3,k = 5

# j = 3
#            i     k            j
# n1 = [1,2,3,0,0,0] n2 = [2,5,6]
# j = 2
#            i   k            j
# n1 = [1,2,3,0,0,6] n2 = [2,5,6]
# j=1
#            i k            j
```

Saved to local · Ln 1, Col 1



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

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

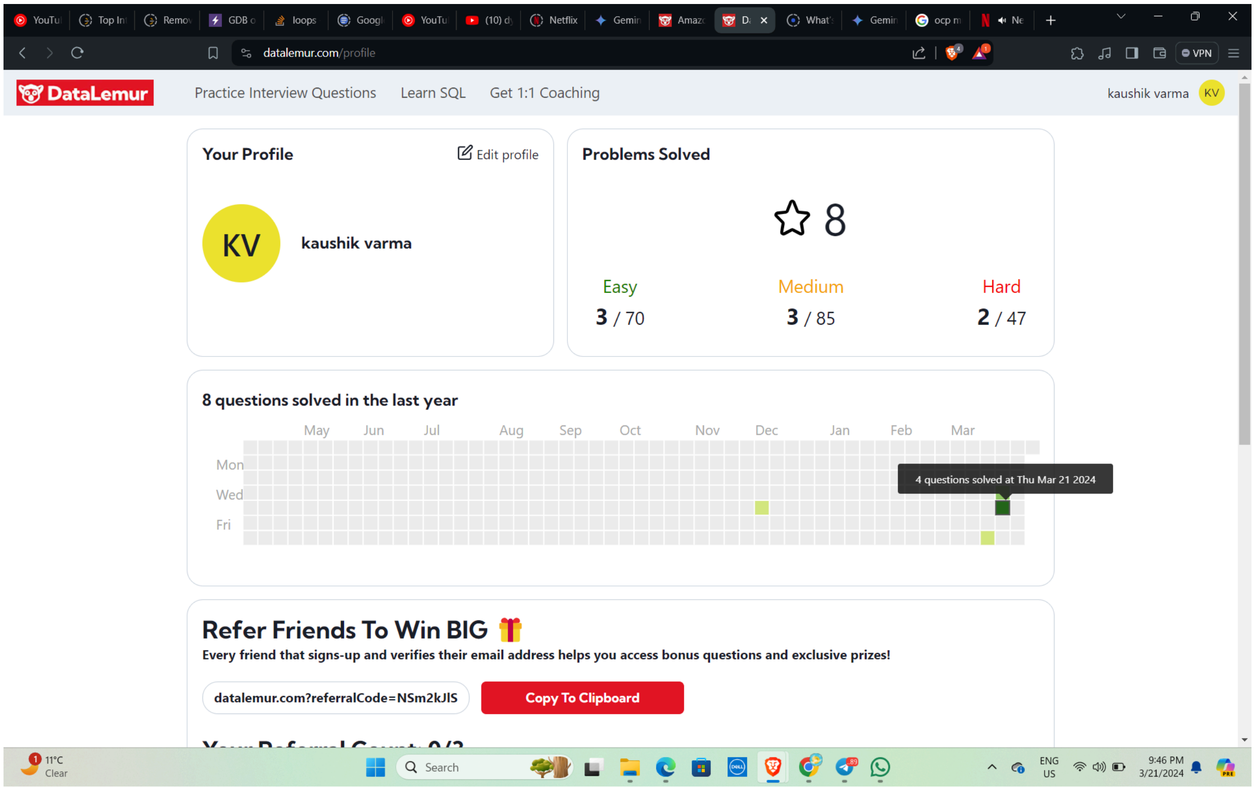
```
class Solution:
    def removeElement(self, nums: List[int], val: int) -> int:
        k = 0

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

        return k

'''Explanation :
Step1 : here we will replace nums[] with values other than val
Step2 : In the end k gives us the index of array which is length
'''
```

Saved to local · Ln 16, Col 1

Test Result: Accepted · Runtime: 61 ms · Case 1 · Case 2

2. Find the DSA google doc for 21st : DSA_Arrays_03/21/2024
3. Spark theory is in process : SparkTH Day8 03/21/2024

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



Your Profile — Edit profile

KV kaushik varma

Problems Solved

☆ 8

Easy	Medium	Hard
3 / 70	3 / 85	2 / 47

8 questions solved in the last year

May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar

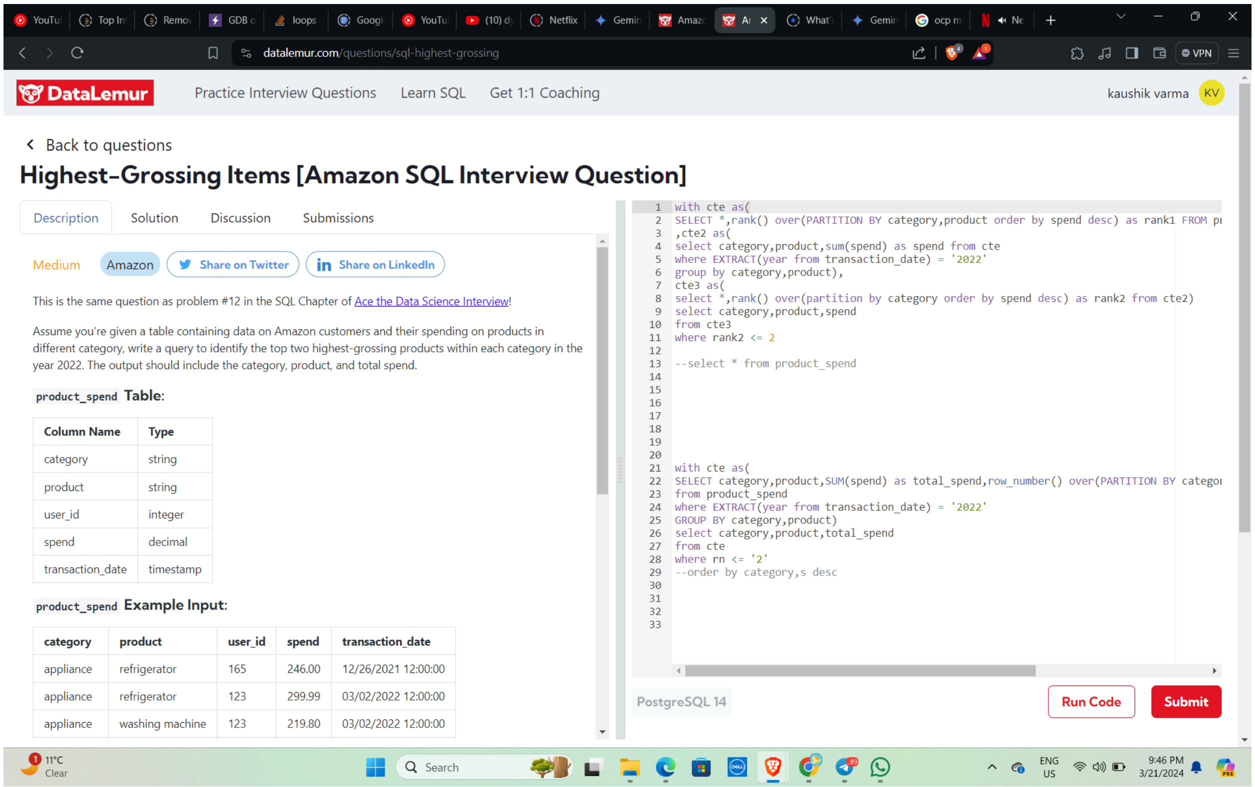
Mon Wed Fri

4 questions solved at Thu Mar 21 2024

Refer Friends To Win BIG

Every friend that signs-up and verifies their email address helps you access bonus questions and exclusive prizes!

datalemur.com?referralCode=NSm2kJIS [Copy To Clipboard]



Highest-Grossing Items [Amazon SQL Interview Question]

This is the same question as problem #12 in the SQL Chapter of Ace the Data Science Interview!

Assume you're given a table containing data on Amazon customers and their spending on products in different category, write a query to identify the top two highest-grossing products within each category in the year 2022. The output should include the category, product, and total spend.

`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

```
with cte as(
SELECT *,rank() over(PARTITION BY category,product order by spend desc) as rank1 FROM pı
,cte2 as(
select category,product,sum(spend) as spend from cte
where EXTRACT(year from transaction_date) = '2022'
group by category,product),
cte3 as(
select *,rank() over(partition by category order by spend desc) as rank2 from cte2)
select category,product,spend
from cte3
where rank2 <= 2

--select * from product_spend







with cte as(
SELECT category,product,SUM(spend) as total_spend,row_number() over(PARTITION BY categoı
from product_spend
where EXTRACT(year from transaction_date) = '2022'
GROUP BY category,product)
select category,product,total_spend
from cte
where rn <= '2'
--order by category,s desc
```