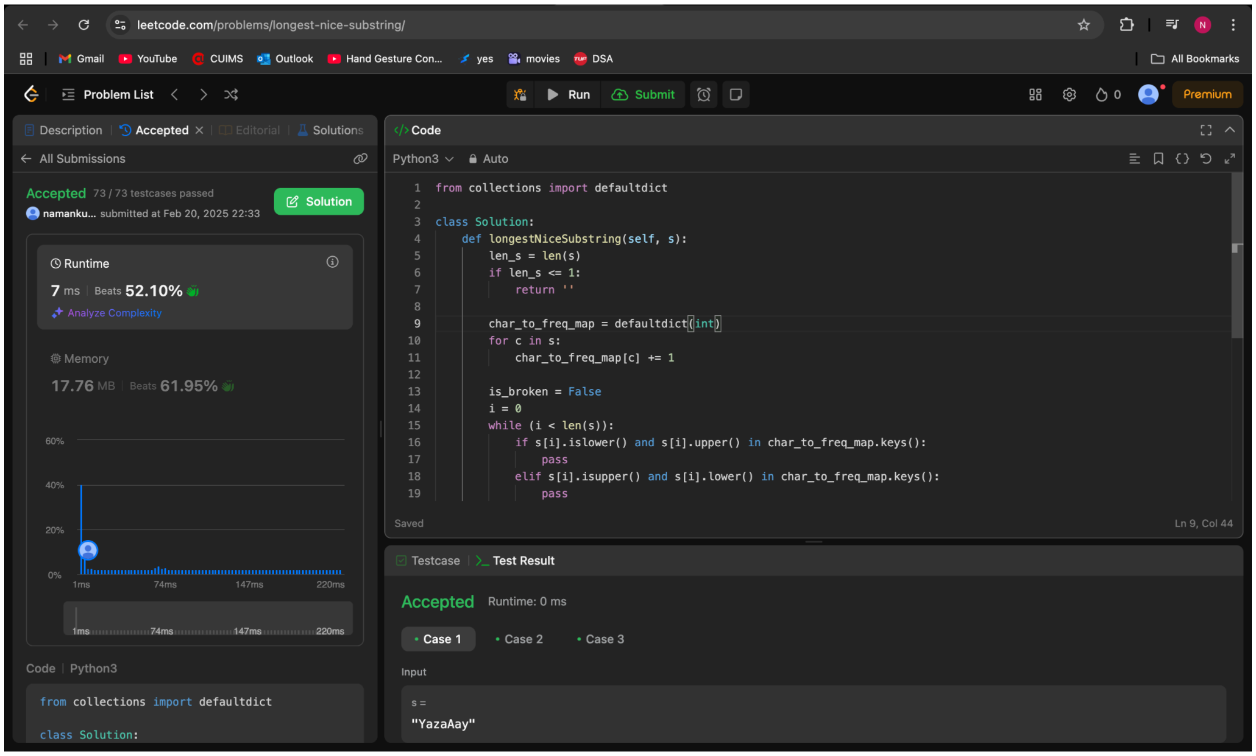
AP ASSIGNMENT - 4

Name : Naman Kumar
UID : 22ICS10001
Class : FL_IOT_604 (A)

1763. Longest Nice Substring

```
from collections import defaultdict

class Solution:
    def longestNiceSubstring(self, s):
        len_s = len(s)
        if len_s <= 1:
            return ''

        char_to_freq_map = defaultdict(int)
        for c in s:
            char_to_freq_map[c] += 1

        is_broken = False
        i = 0
        while (i < len(s)):
            if s[i].islower() and s[i].upper() in char_to_freq_map.keys():
                pass
            elif s[i].isupper() and s[i].lower() in char_to_freq_map.keys():
                pass
            else:
                is_broken = True
                break
            i += 1

        if not is_broken:
            return s

        longest_nice_substr_1 = self.longestNiceSubstring(s[:i])
```

```python
        longest_nice_substr_2 = self.longestNiceSubstring(s[i+1:])

        if len(longest_nice_substr_1)>=len(longest_nice_substr_2):
            return longest_nice_substr_1
        else:
            return longest_nice_substr_2
```



Accepted 73 / 73 testcases passed
namanku... submitted at Feb 20, 2025 22:33

Runtime
7 ms | Beats 52.10%

Memory
17.76 MB | Beats 61.95%

Code | Python3

```python
from collections import defaultdict

class Solution:
```

```python
from collections import defaultdict

class Solution:
    def longestNiceSubstring(self, s):
        len_s = len(s)
        if len_s <= 1:
            return ''

        char_to_freq_map = defaultdict(int)
        for c in s:
            char_to_freq_map[c] += 1

        is_broken = False
        i = 0
        while (i < len(s)):
            if s[i].islower() and s[i].upper() in char_to_freq_map.keys():
                pass
            elif s[i].isupper() and s[i].lower() in char_to_freq_map.keys():
                pass
```

Accepted Runtime: 0 ms

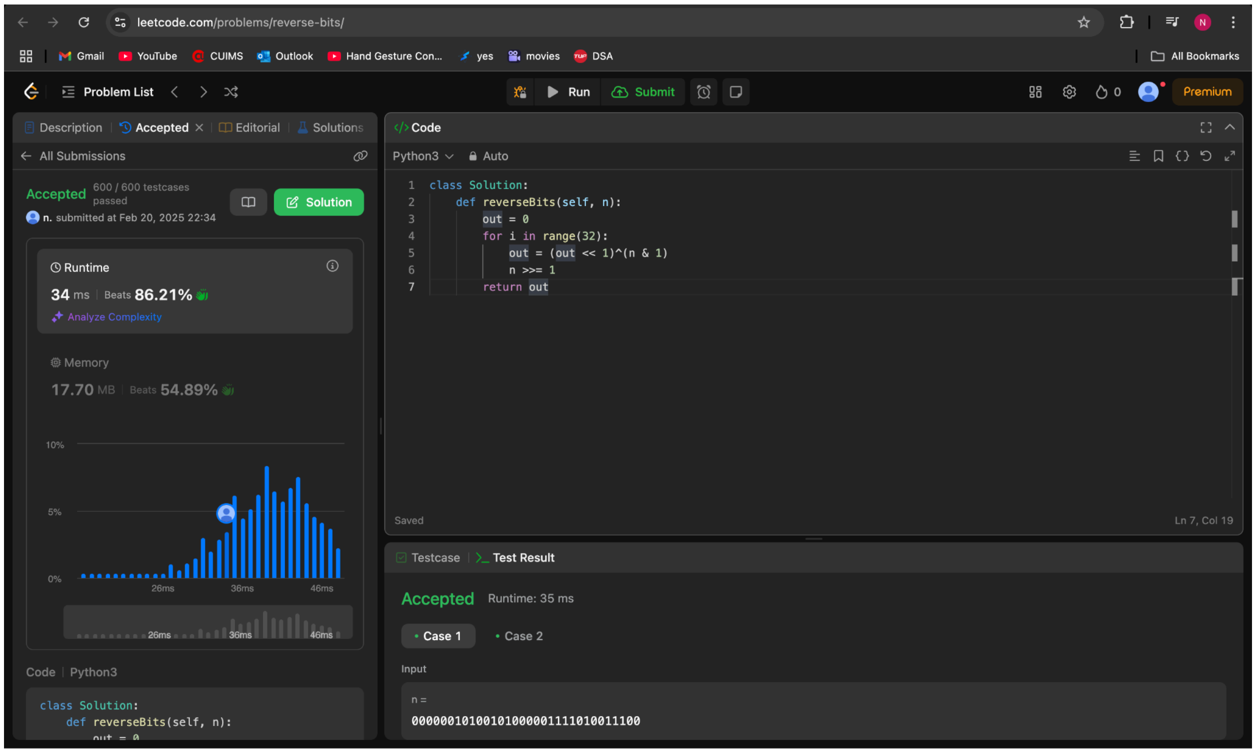
Case 1 Case 2 Case 3

Input

s =
"YazaAay"

190. Reverse Bits

```python
class Solution:
    def reverseBits(self, n):
        out = 0
        for i in range(32):
            out = (out << 1)^(n & 1)
            n >>= 1
        return out
```

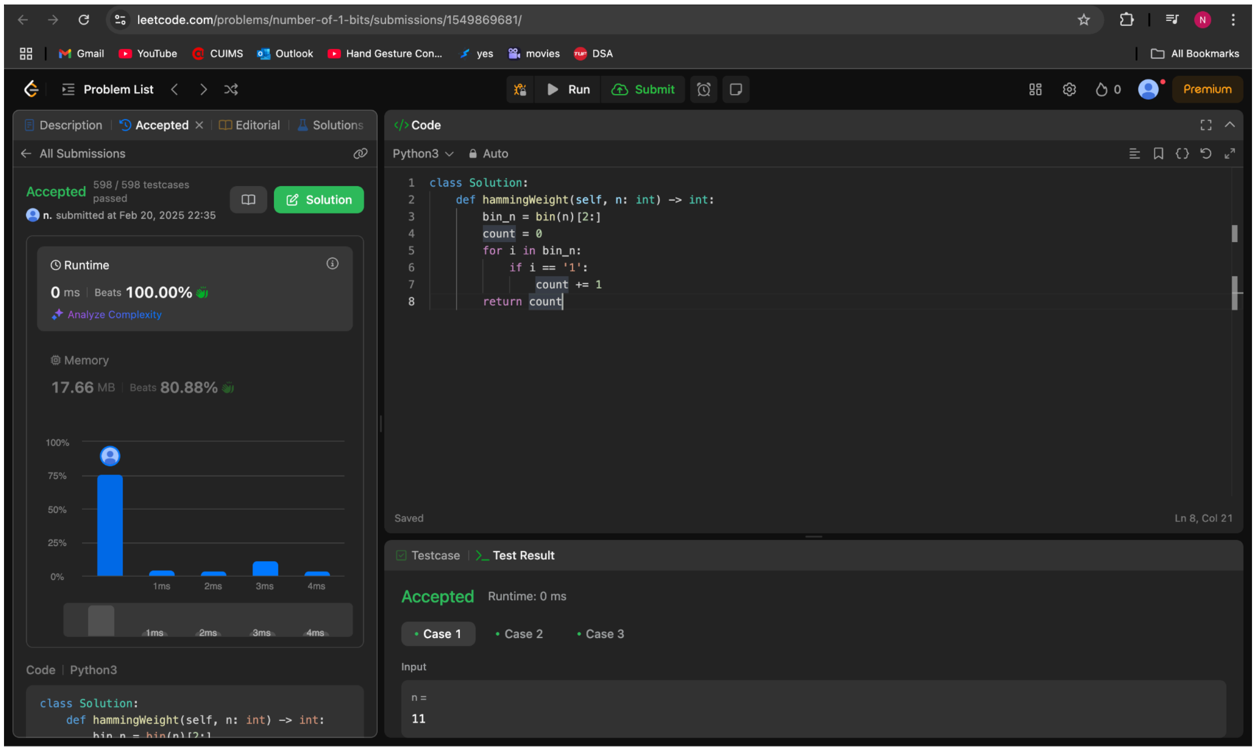


```python
class Solution:
    def reverseBits(self, n):
        out = 0
        for i in range(32):
            out = (out << 1)^(n & 1)
            n >>= 1
        return out
```

```
00000010100101000001111010011100
```

191. Number of 1 Bits

```python
class Solution:
    def hammingWeight(self, n: int) -> int:
        bin_n = bin(n)[2:]
        count = 0
        for i in bin_n:
            if i == '1':
                count += 1
        return count
```


leetcode.com/problems/number-of-1-bits/submissions/1549869681/

Accepted 598 / 598 testcases passed

n. submitted at Feb 20, 2025 22:35

Runtime 0 ms | Beats 100.00%

Memory 17.66 MB | Beats 80.88%

Code | Python3

```
class Solution:
    def hammingWeight(self, n: int) -> int:
        bin_n = bin(n)[2:]
        count = 0
        for i in bin_n:
            if i == '1':
                count += 1
        return count
```

Saved Ln 8, Col 21

Testcase | Test Result

Accepted Runtime: 0 ms

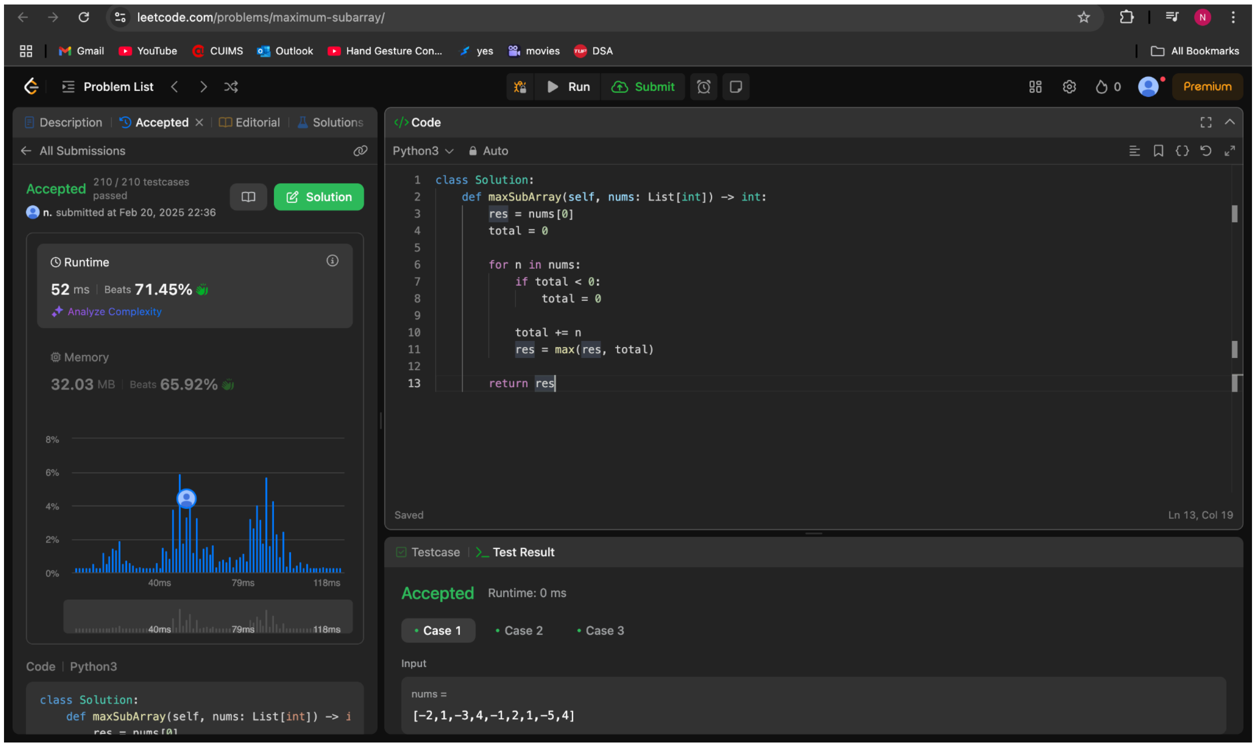
Case 1 • Case 2 • Case 3

Input

n =
11

53. Maximum Subarray

```
class Solution:
   def maxSubArray(self, nums: List[int]) -> int:
       res = nums[0]
       total = 0

       for n in nums:
           if total < 0:
               total = 0

           total += n
           res = max(res, total)

       return res
```



Accepted 210 / 210 testcases passed

n. submitted at Feb 20, 2025 22:36

Runtime 52 ms | Beats 71.45%

Memory 32.03 MB | Beats 65.92%

```python
class Solution:
    def maxSubArray(self, nums: List[int]) -> int:
        res = nums[0]
        total = 0

        for n in nums:
            if total < 0:
                total = 0

            total += n
            res = max(res, total)

        return res
```

Accepted Runtime: 0 ms

Case 1 · Case 2 · Case 3

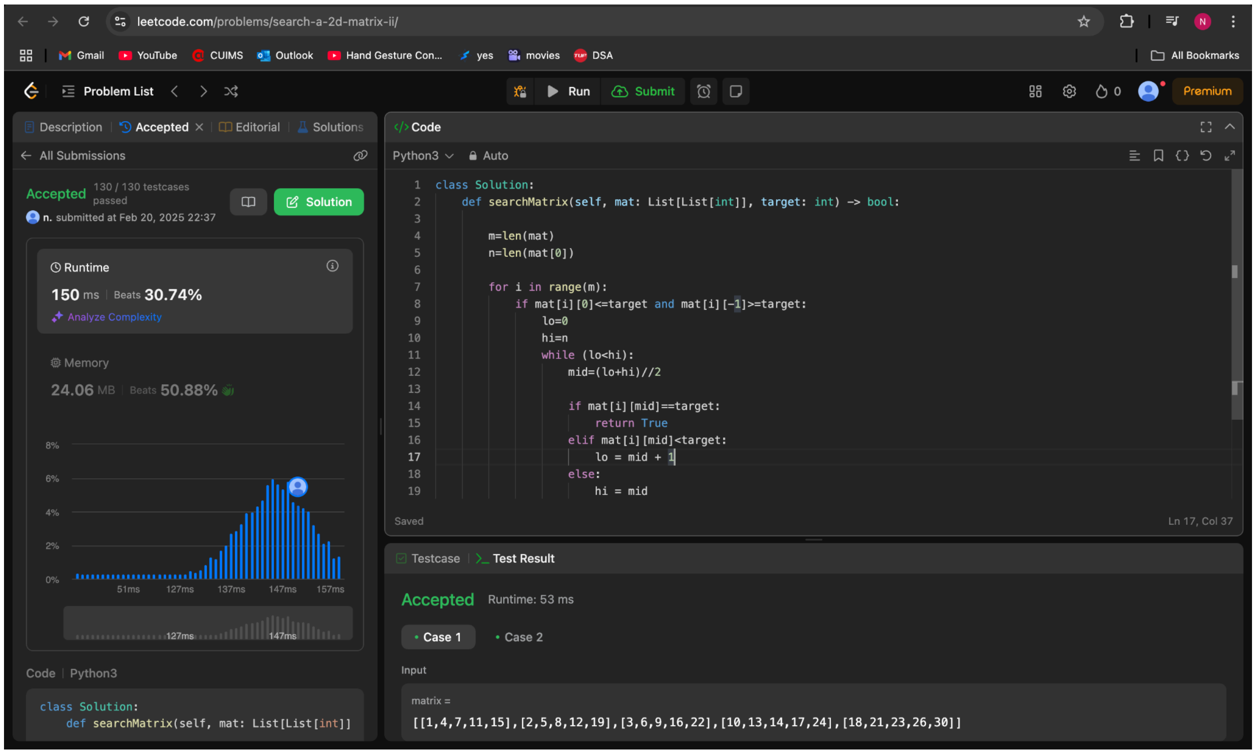
Input

nums =
[-2,1,-3,4,-1,2,1,-5,4]

240. Search a 2D Matrix II

```python
class Solution:
    def searchMatrix(self, mat: List[List[int]], target: int) -> bool:

        m=len(mat)
        n=len(mat[0])

        for i in range(m):
            if mat[i][0]<=target and mat[i][-1]>=target:
                lo=0
                hi=n
                while (lo<hi):
                    mid=(lo+hi)//2

                    if mat[i][mid]==target:
                        return True
                    elif mat[i][mid]<target:
```

```
                    lo = mid + 1
                else:
                    hi = mid

        return False
```



Accepted 130 / 130 testcases passed

n. submitted at Feb 20, 2025 22:37

Runtime

150 ms | Beats 30.74%

Analyze Complexity

Memory

24.06 MB | Beats 50.88%

Code | Python3

```
class Solution:
    def searchMatrix(self, mat: List[List[int]]
```

```
class Solution:
    def searchMatrix(self, mat: List[List[int]], target: int) -> bool:

        m=len(mat)
        n=len(mat[0])

        for i in range(m):
            if mat[i][0]<=target and mat[i][-1]>=target:
                lo=0
                hi=n
                while (lo<hi):
                    mid=(lo+hi)//2

                    if mat[i][mid]==target:
                        return True
                    elif mat[i][mid]<target:
                        lo = mid + 1
                    else:
                        hi = mid
```

Saved

Ln 17, Col 37

Accepted Runtime: 53 ms

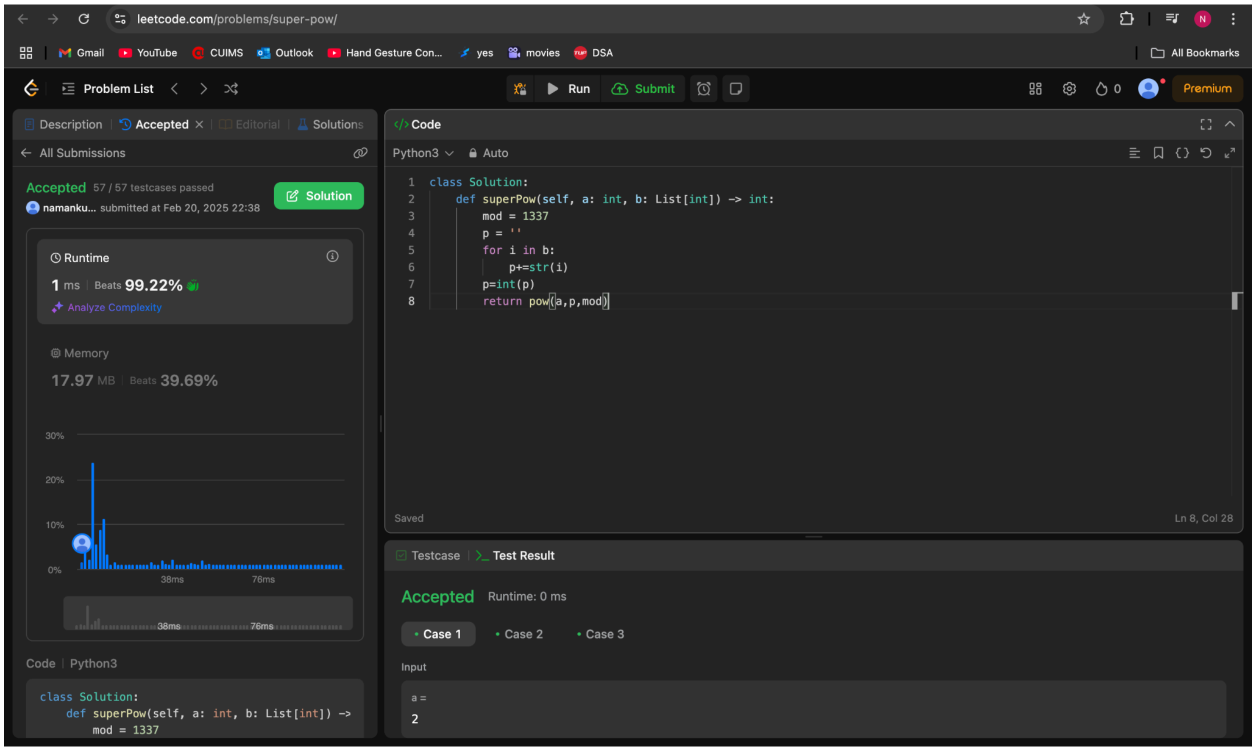
Input

matrix =

```
[[1,4,7,11,15],[2,5,8,12,19],[3,6,9,16,22],[10,13,14,17,24],[18,21,23,26,30]]
```

372. Super Pow

```
class Solution:
    def superPow(self, a: int, b: List[int]) -> int:
        mod = 1337
        p = ''
        for i in b:
            p+=str(i)
        p=int(p)
        return pow(a,p,mod)
```



Accepted 57 / 57 testcases passed

namanku... submitted at Feb 20, 2025 22:38

Runtime

1 ms | Beats 99.22%

Memory

17.97 MB | Beats 39.69%

```python
class Solution:
    def superPow(self, a: int, b: List[int]) -> int:
        mod = 1337
        p = ''
        for i in b:
            p+=str(i)
        p=int(p)
        return pow(a,p,mod)
```

Accepted Runtime: 0 ms

Case 1 · Case 2 · Case 3

Input

a =

2

Code | Python3

```python
class Solution:
    def superPow(self, a: int, b: List[int]) ->
        mod = 1337
```