Assignment-4

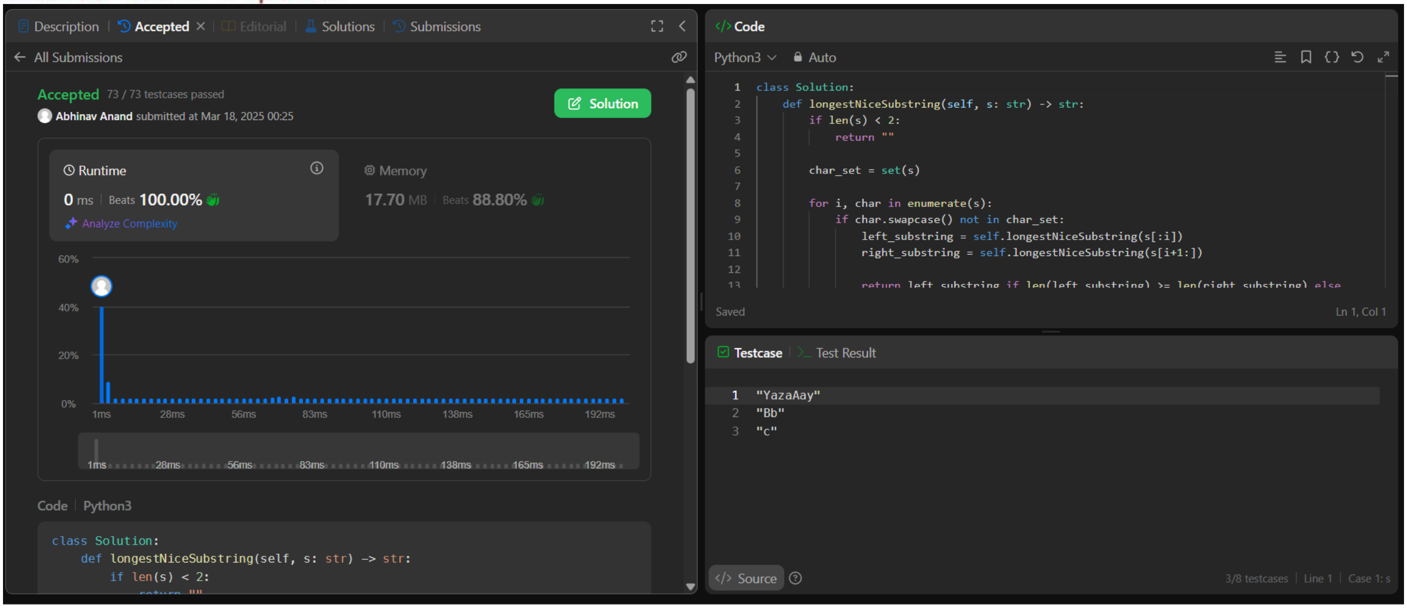
Student Name: Abhinav Anand **UID: 22BCS15064**
Branch: CSE **Section/Group: IOT-609-B**
Semester: 6th **Date: 17-03-2025**
Subject Name: AP Lab **Subject Code: 22CSP-351**

Ques 1: Longest Nice Substring

Code: -

```
class Solution {
public:
    string longestNiceSubstring(string s) {
        if (s.size() < 2) return "";
        unordered_set<char> st(s.begin(), s.end());
        for (int i = 0; i < s.size(); i++) {
         if (st.find(tolower(s[i])) != st.end() && st.find(toupper(s[i])) != st.end()) continue;
         string left = longestNiceSubstring(s.substr(0, i));
          string right = longestNiceSubstring(s.substr(i + 1));
           if (left.size() >= right.size()) return left;
           else  return right; }
           return s; }};
```

Submission: -



Accepted 73 / 73 testcases passed

Abhinav Anand submitted at Mar 18, 2025 00:25

Runtime: 0 ms | Beats 100.00%

Memory: 17.70 MB | Beats 88.80%

Code | Python3

```
class Solution:
    def longestNiceSubstring(self, s: str) -> str:
        if len(s) < 2:
```

```
class Solution:
    def longestNiceSubstring(self, s: str) -> str:
        if len(s) < 2:
            return ""

        char_set = set(s)

        for i, char in enumerate(s):
            if char.swapcase() not in char_set:
                left_substring = self.longestNiceSubstring(s[:i])
                right_substring = self.longestNiceSubstring(s[i+1:])

                return left_substring if len(left_substring) >= len(right_substring) else
```

```
"YazaAay"
"Bb"
"c"
```

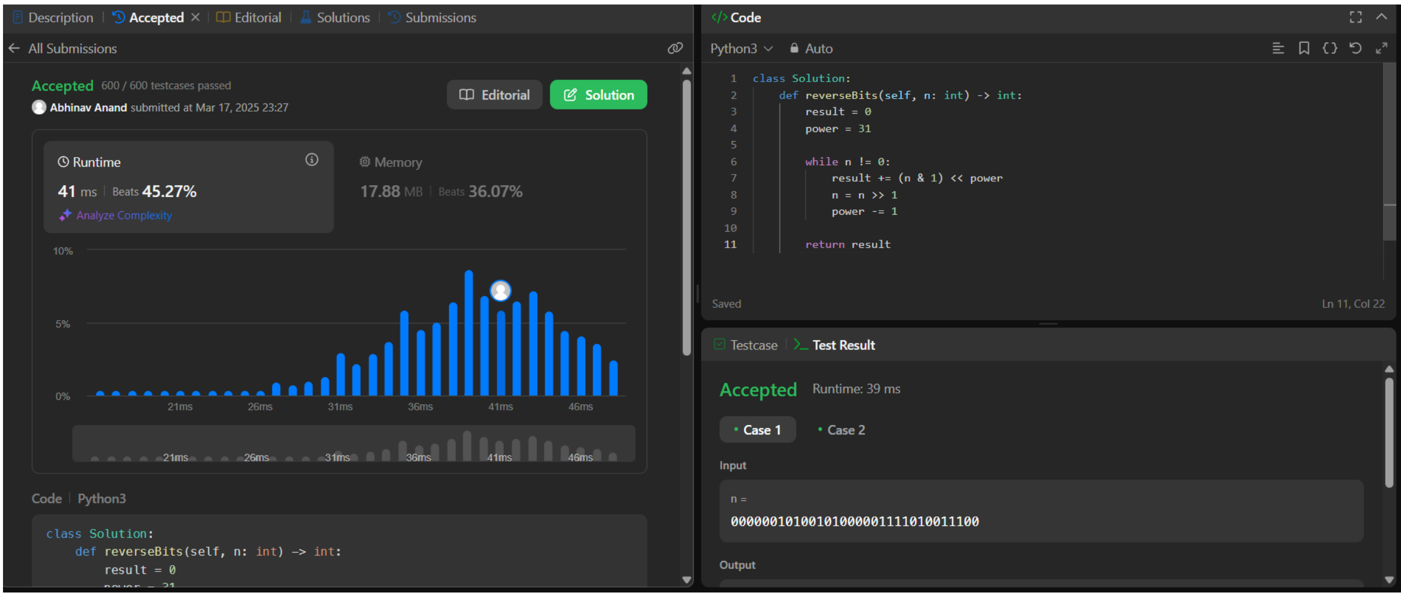
3/8 testcases | Line 1 | Case 1: s

Ques 2: Reverse Bits

Code: -

```
class Solution {
public:
    uint32_t reverseBits(uint32_t n) {
        uint32_t ans=0;
        for (int i = 0; i < 32; i++) {
            ans =  ans<<1;
            if(n&1){
                ans=ans|1;
            }
            n = n>>1;
        }
        return ans;
    }
};
```

Submission: -



Accepted 600 / 600 testcases passed

Abhinav Anand submitted at Mar 17, 2025 23:27

Runtime

41 ms | Beats 45.27%

Analyze Complexity

Memory

17.88 MB | Beats 36.07%

Code | Python3

```
class Solution:
    def reverseBits(self, n: int) -> int:
        result = 0
        power = 31
```

Python3 Auto

```
class Solution:
    def reverseBits(self, n: int) -> int:
        result = 0
        power = 31

        while n != 0:
            result += (n & 1) << power
            n = n >> 1
            power -= 1

        return result
```

Testcase | Test Result

Accepted Runtime: 39 ms

• Case 1 • Case 2

Input

```
n =
00000010100101000001111010011100
```

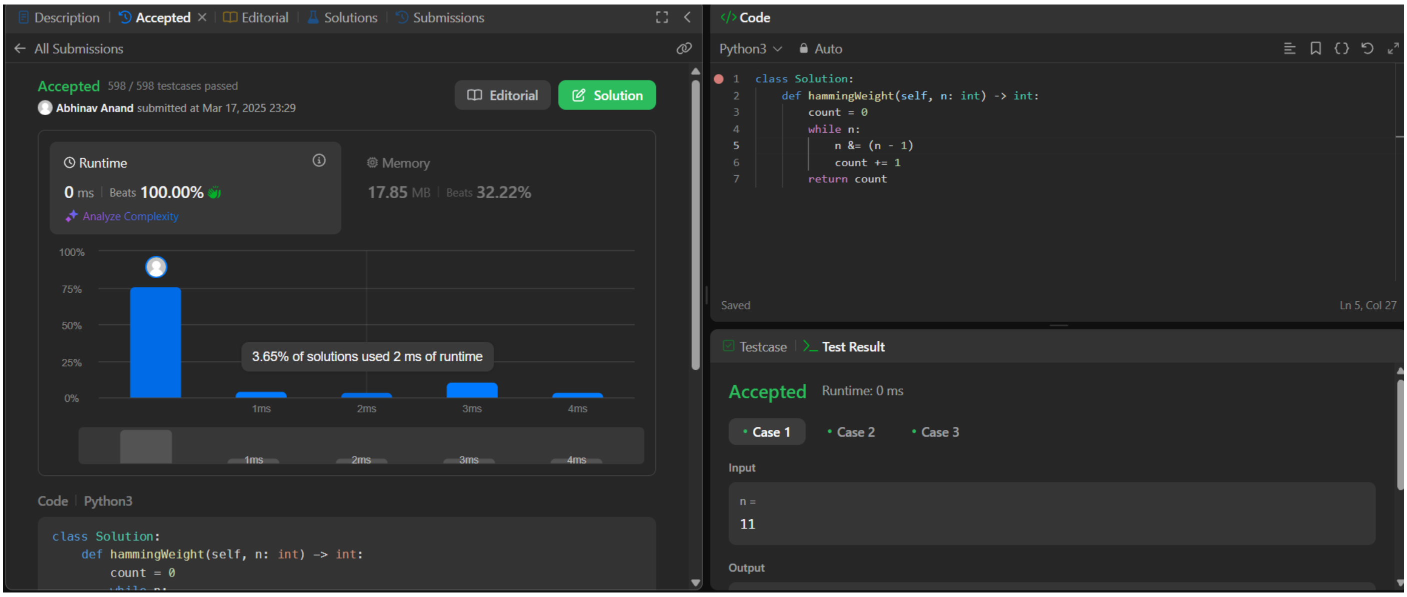
Output

Ques 3: Number of 1 Bits

Code: -

```
class Solution {
public:
    int hammingWeight(int n) {
        int count=0;
        while(n!=0){
            if(n&1){
                count++;
            }
            n=n>>1;
        }
        return count;
    }
};
```

Submission: -



Accepted 598 / 598 testcases passed

Abhinav Anand submitted at Mar 17, 2025 23:29

Runtime 0 ms | Beats 100.00%

Memory 17.85 MB | Beats 32.22%

3.65% of solutions used 2 ms of runtime

Code | Python3

```
class Solution:
    def hammingWeight(self, n: int) -> int:
        count = 0
```

```
class Solution:
    def hammingWeight(self, n: int) -> int:
        count = 0
        while n:
            n &= (n - 1)
            count += 1
        return count
```

Saved

Ln 5, Col 27

Accepted Runtime: 0 ms

n =
11

Ques 4: Maximum Subarray

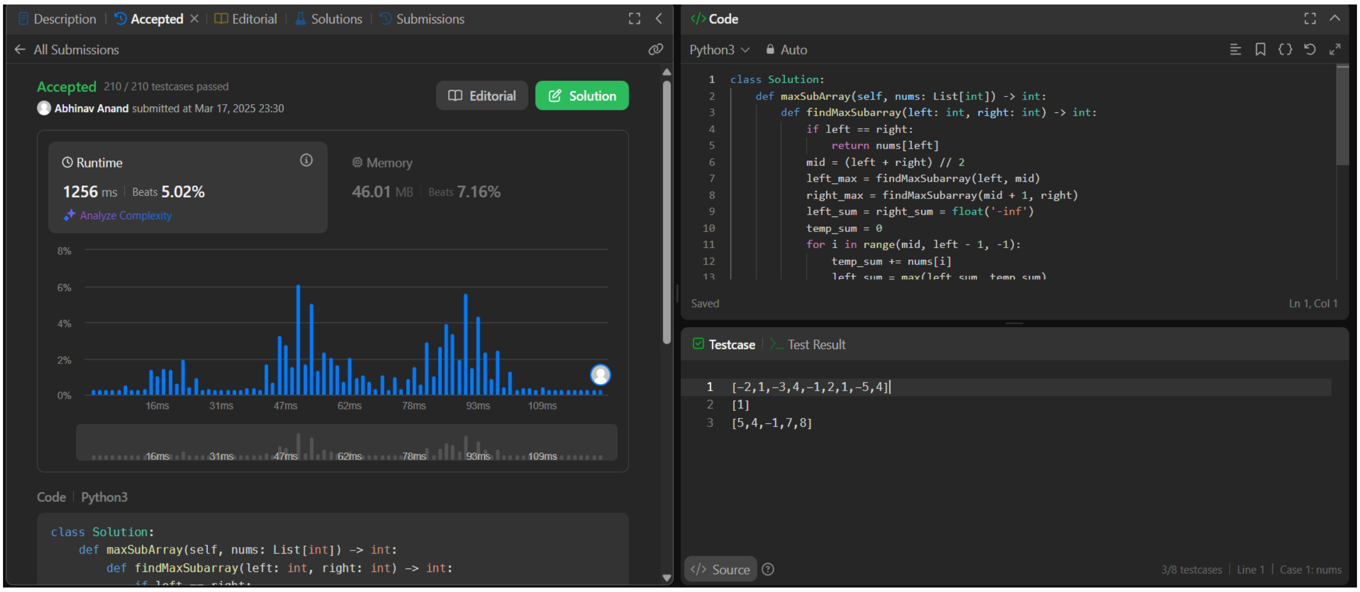
Code: -

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int n = nums.size();
        int sum = 0;
        int maxi = nums[0];

        for (int i = 0; i < n; i++) {
            sum += nums[i];
            maxi = max(sum, maxi);
            if (sum < 0) {
                sum = 0;
            }
        }

        return maxi;
    }
};
```

Submission: -



Accepted 210 / 210 testcases passed
Abhinav Anand submitted at Mar 17, 2025 23:30

Runtime
1256 ms | Beats 5.02%

Memory
46.01 MB | Beats 7.16%

Code | Python3

```
class Solution:
    def maxSubArray(self, nums: List[int]) -> int:
        def findMaxSubarray(left: int, right: int) -> int:
            if left == right:
```

```
class Solution:
    def maxSubArray(self, nums: List[int]) -> int:
        def findMaxSubarray(left: int, right: int) -> int:
            if left == right:
                return nums[left]
            mid = (left + right) // 2
            left_max = findMaxSubarray(left, mid)
            right_max = findMaxSubarray(mid + 1, right)
            left_sum = right_sum = float('-inf')
            temp_sum = 0
            for i in range(mid, left - 1, -1):
                temp_sum += nums[i]
                left_sum = max(left_sum, temp_sum)
```

Saved
Ln 1, Col 1

```
[-2,1,-3,4,-1,2,1,-5,4]
[1]
[5,4,-1,7,8]
```

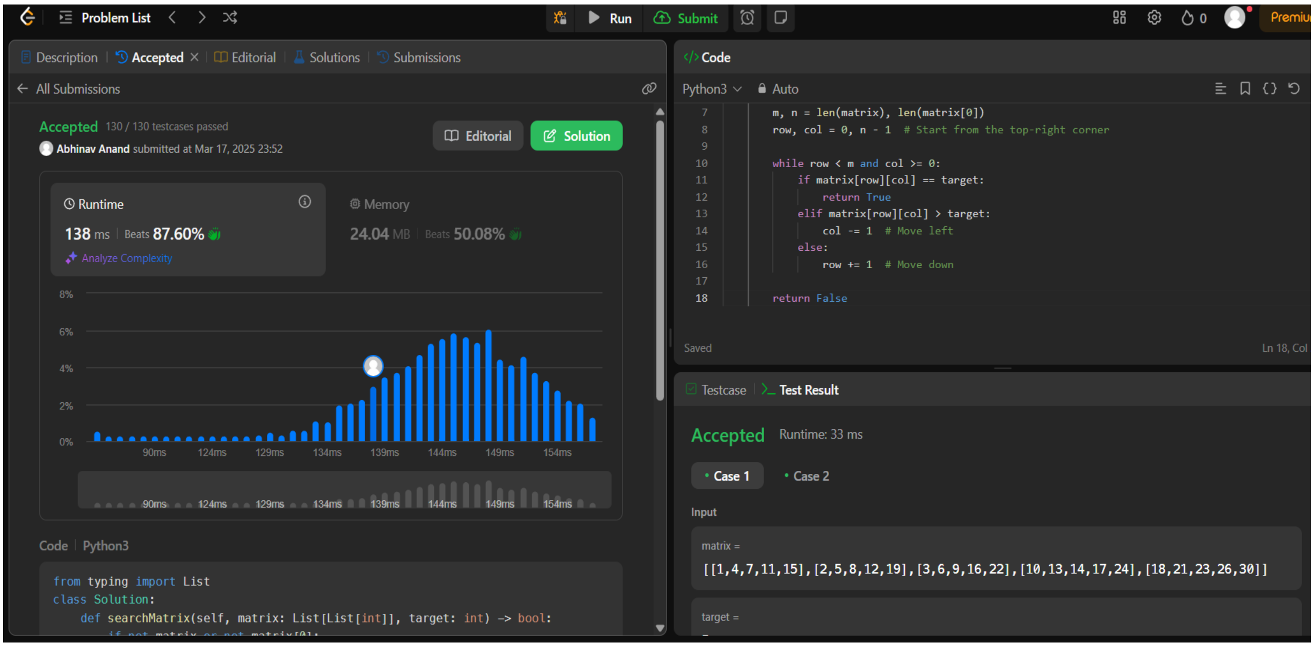
3/8 testcases | Line 1 | Case 1: nums

Ques 5: Search a 2D Matrix II

Code: -

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int r=0,c=matrix[0].size()-1;
        while (r<matrix.size() && c>=0){
            if (matrix[r][c]==target){return true;}
            else if (matrix[r][c]<target){r++;}
            else {c--;}
        }
        return false;
    }
};
```

Submission: -



Accepted 130 / 130 testcases passed

Abhinav Anand submitted at Mar 17, 2025 23:52

138 ms | Beats 87.60%

24.04 MB | Beats 50.08%

Code | Python3

```
from typing import List
class Solution:
    def searchMatrix(self, matrix: List[List[int]], target: int) -> bool:
```

```
        m, n = len(matrix), len(matrix[0])
        row, col = 0, n - 1  # Start from the top-right corner

        while row < m and col >= 0:
            if matrix[row][col] == target:
                return True
            elif matrix[row][col] > target:
                col -= 1  # Move left
            else:
                row += 1  # Move down

        return False
```

Saved

Ln 18, Col

Accepted Runtime: 33 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]]

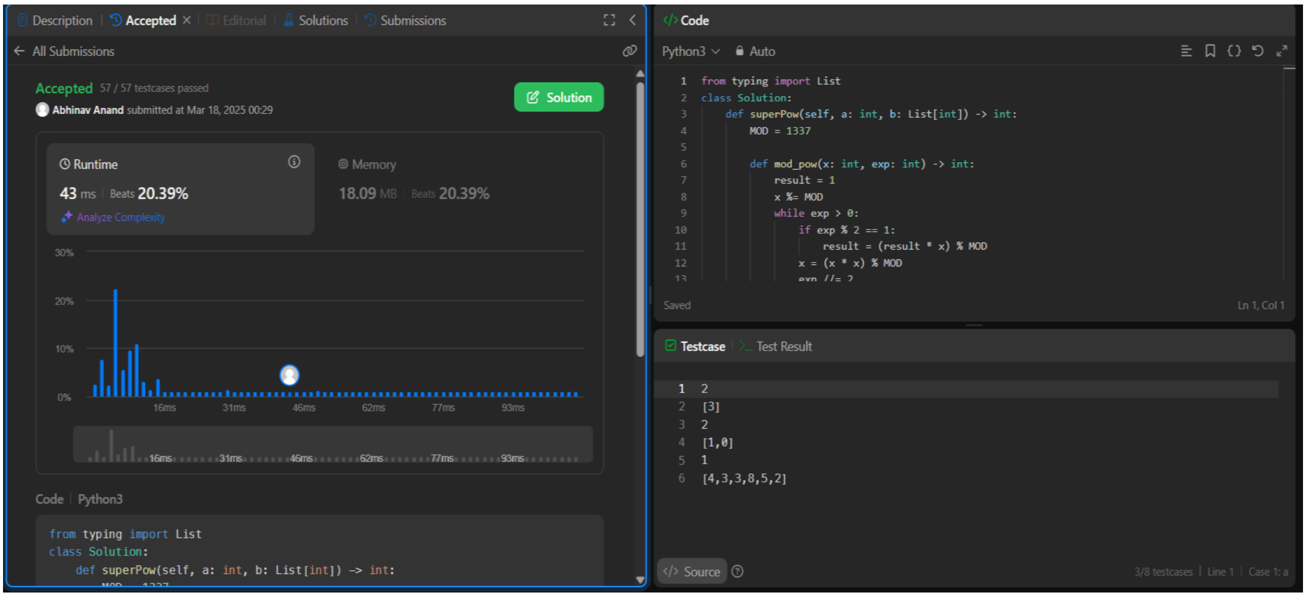
target =

Ques 6: Super Pow

Code: -

```
class Solution {
private:
    int solve(int base, int power, int mod) {
        int ans = 1;
        while (power > 0) {
            if (power & 1) {
                ans = (ans * base) % mod;}
            base = (base * base) % mod;
            power >>= 1;} return ans;}
public:
    int superPow(int a, vector<int>& b) {
        a%=1337;
        int n = b.size();
        int m = 1140;
        int expi = 0;
        for(int i : b){
            expi = (expi*10+i)%m;}
        if (expi == 0) { expi = m; }
        return solve(a,expi,1337); } };
```

Submission: -



Accepted 57 / 57 testcases passed
Abhinav Anand submitted at Mar 18, 2025 00:29

Runtime 43 ms | Beats 20.39%

Memory 18.09 MB | Beats 20.39%

Code | Python3

```
from typing import List
class Solution:
    def superPow(self, a: int, b: List[int]) -> int:
        MOD = 1337

        def mod_pow(x: int, exp: int) -> int:
            result = 1
            x %= MOD
            while exp > 0:
                if exp % 2 == 1:
                    result = (result * x) % MOD
                x = (x * x) % MOD
                exp //= 2
```

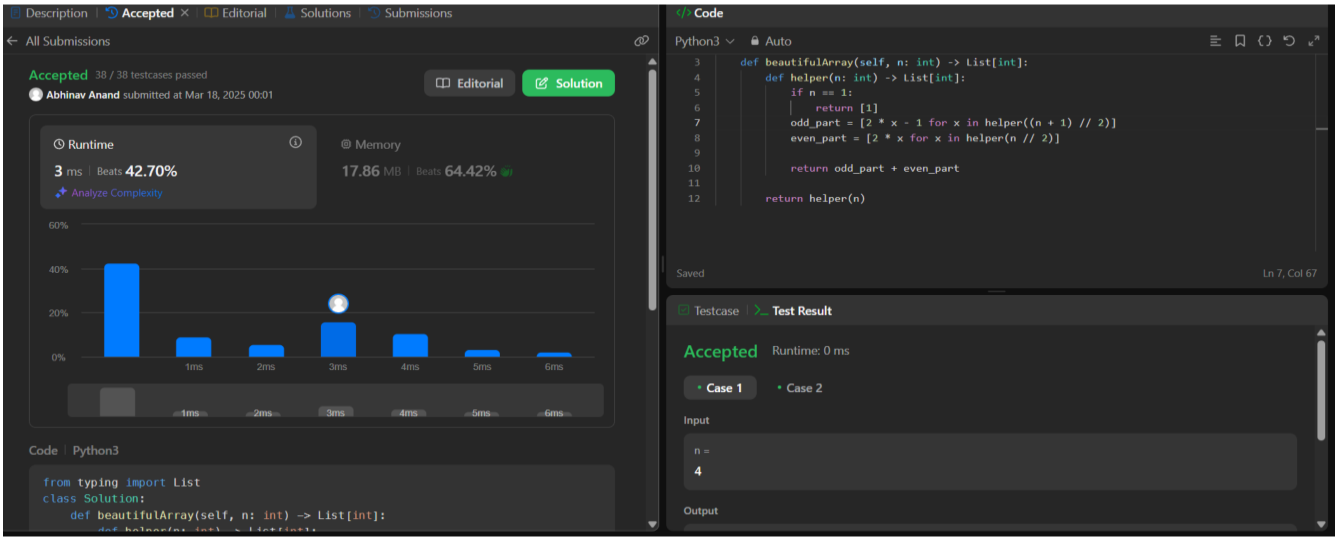
```
2
[3]
2
[1,0]
1
[4,3,3,8,5,2]
```

Ques 7: Beautiful Array

Code: -

```
class Solution {
public:
    vector<int> beautifulArray(int n)
    {
        vector<int> res = {1};
        while (res.size() < n)
        {
            vector<int> temp;
            for (int it : res)
            {
                if (it * 2 - 1 <= n)
                  temp.push_back(it * 2 - 1);
            }
            for (int it : res)
            {
                if (it * 2 <= n)
                  temp.push_back(it * 2);
            }
            res = temp;
        }
        return res; }};
```

Submission: -



Accepted 38 / 38 testcases passed
Abhinav Anand submitted at Mar 18, 2025 00:01

Runtime
3 ms | Beats 42.70%
Analyze Complexity

Memory
17.86 MB | Beats 64.42%

Code | Python3

```
from typing import List
class Solution:
    def beautifulArray(self, n: int) -> List[int]:
```

```
    def beautifulArray(self, n: int) -> List[int]:
        def helper(n: int) -> List[int]:
            if n == 1:
                return [1]
            odd_part = [2 * x - 1 for x in helper((n + 1) // 2)]
            even_part = [2 * x for x in helper(n // 2)]

            return odd_part + even_part

        return helper(n)
```

Saved
Ln 7, Col 67

Accepted Runtime: 0 ms

Input

n =
4

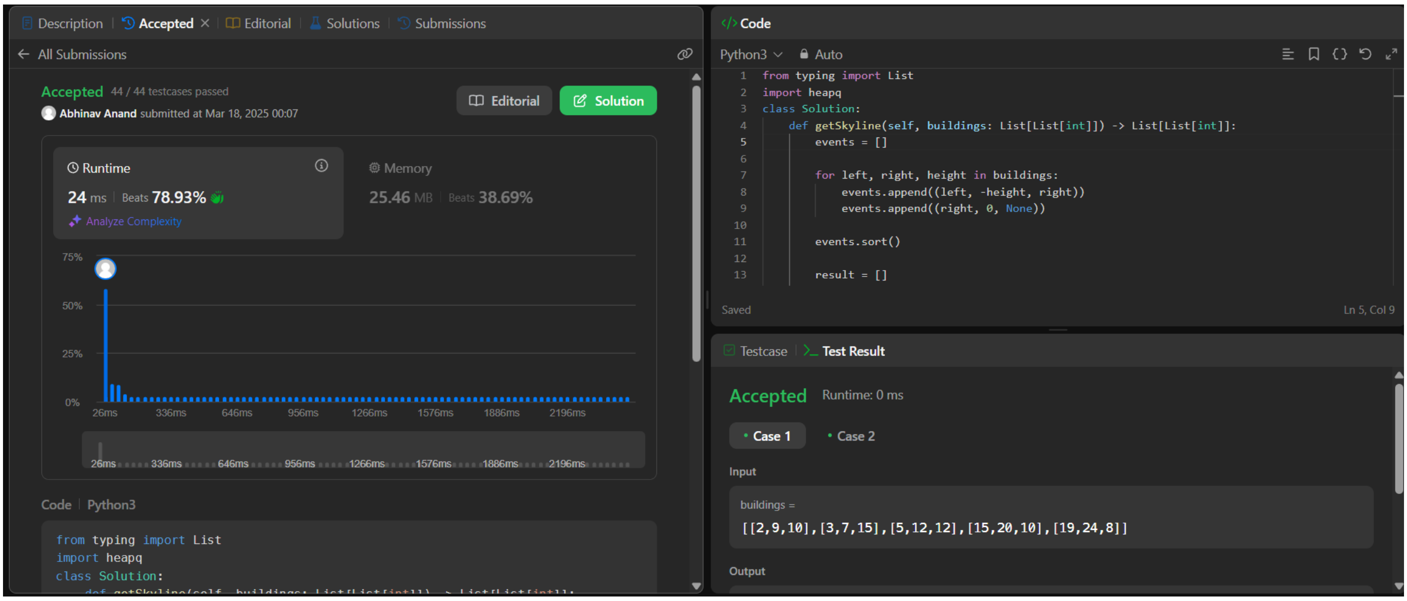
Output

Ques 8: The Skyline Problem

Code: -

```
  class Solution {
public:
  vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
    int edge_idx = 0;  vector<pair<int, int>> edges;
    priority_queue<pair<int, int>> pq;
    vector<vector<int>> skyline;
    for (int i = 0; i < buildings.size(); ++i) {
      const auto &b = buildings[i]; edges.emplace_back(b[0], i);
      edges.emplace_back(b[1], i); }
    std::sort(edges.begin(), edges.end());
    while (edge_idx < edges.size()) {
      int curr_height;
      const auto &[curr_x, _] = edges[edge_idx];
      while (edge_idx < edges.size() &&
          curr_x == edges[edge_idx].first) {
        const auto &[_, building_idx] = edges[edge_idx];
        const auto &b = buildings[building_idx];
        if (b[0] == curr_x)
          pq.emplace(b[2], b[1]);
        ++edge_idx; }
      while (!pq.empty() && pq.top().second <= curr_x)
        pq.pop();
      curr_height = pq.empty() ? 0 : pq.top().first;
      if (skyline.empty() || skyline.back()[1] != curr_height)
        skyline.push_back({curr_x, curr_height}); }
    return skyline;
    }
    };
```

Submission: -



Accepted 44 / 44 testcases passed

Abhinav Anand submitted at Mar 18, 2025 00:07

Runtime
24 ms | Beats 78.93%

Memory
25.46 MB | Beats 38.69%

Code | Python3

```
from typing import List
import heapq
class Solution:
    def getSkyline(self, buildings: List[List[int]]) -> List[List[int]]:
        events = []

        for left, right, height in buildings:
            events.append((left, -height, right))
            events.append((right, 0, None))

        events.sort()

        result = []
```

Accepted Runtime: 0 ms

Input

buildings =
[[2,9,10],[3,7,15],[5,12,12],[15,20,10],[19,24,8]]

Output

Ques 9: Reverse Pairs

Code: -

```
class Solution {
private:

int countPairs(vector<int>& arr,int low,int mid,int high){
        int cnt=0;
        int right=mid+1;
        for(int i=low;i<=mid;i++){
                while(right<=high && 0.5*arr[i]>arr[right]) right++;
                cnt+=right-(mid+1);
        }
        return cnt;
}

void merge(vector<int>& arr,int low,int mid,int high){
        int left=low;
        int right=mid+1;
        vector<int> temp;
        while(left<=mid && right<=high){
                if(arr[left]<=arr[right]){
                        temp.push_back(arr[left]);
                        left++;
                }
                else{
                        temp.push_back(arr[right]);
                        right++;
                }
        }

        while(left<=mid){
                temp.push_back(arr[left]);
                left++;
        }
        while(right<=high){
```

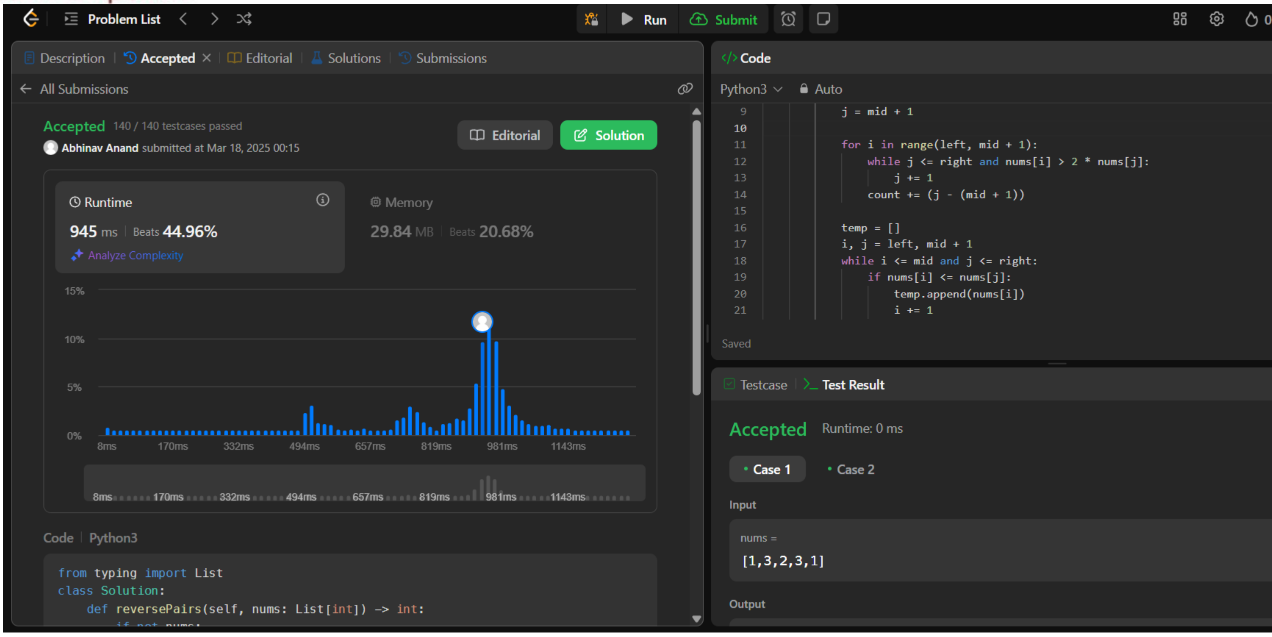
```
temp.push_back(arr[right]);
```

```
                right++;
        }
        for(int i=low;i<=high;i++){
                arr[i]=temp[i-low];
        }
}

int mergesort(vector<int>& arr,int low,int high){
        int cnt=0;
        if(low>=high) return cnt;
        int mid=(low+high)/2;
        cnt+=mergesort(arr,low,mid);
        cnt+=mergesort(arr,mid+1,high);
        cnt+=countPairs(arr,low,mid,high);
        merge(arr,low,mid,high);
        return cnt;
}

public:
    int reversePairs(vector<int>& nums) {
        return mergesort(nums,0,nums.size()-1);
    }
};
```

Submission: -



Accepted 140 / 140 testcases passed

Abhinav Anand submitted at Mar 18, 2025 00:15

Runtime 945 ms | Beats 44.96%

Memory 29.84 MB | Beats 20.68%

Code | Python3

```
from typing import List
class Solution:
    def reversePairs(self, nums: List[int]) -> int:
        if not nums:
```

```
            j = mid + 1

            for i in range(left, mid + 1):
                while j <= right and nums[i] > 2 * nums[j]:
                    j += 1
                count += (j - (mid + 1))

            temp = []
            i, j = left, mid + 1
            while i <= mid and j <= right:
                if nums[i] <= nums[j]:
                    temp.append(nums[i])
                    i += 1
```

Saved

Accepted Runtime: 0 ms

Input

nums =
[1,3,2,3,1]

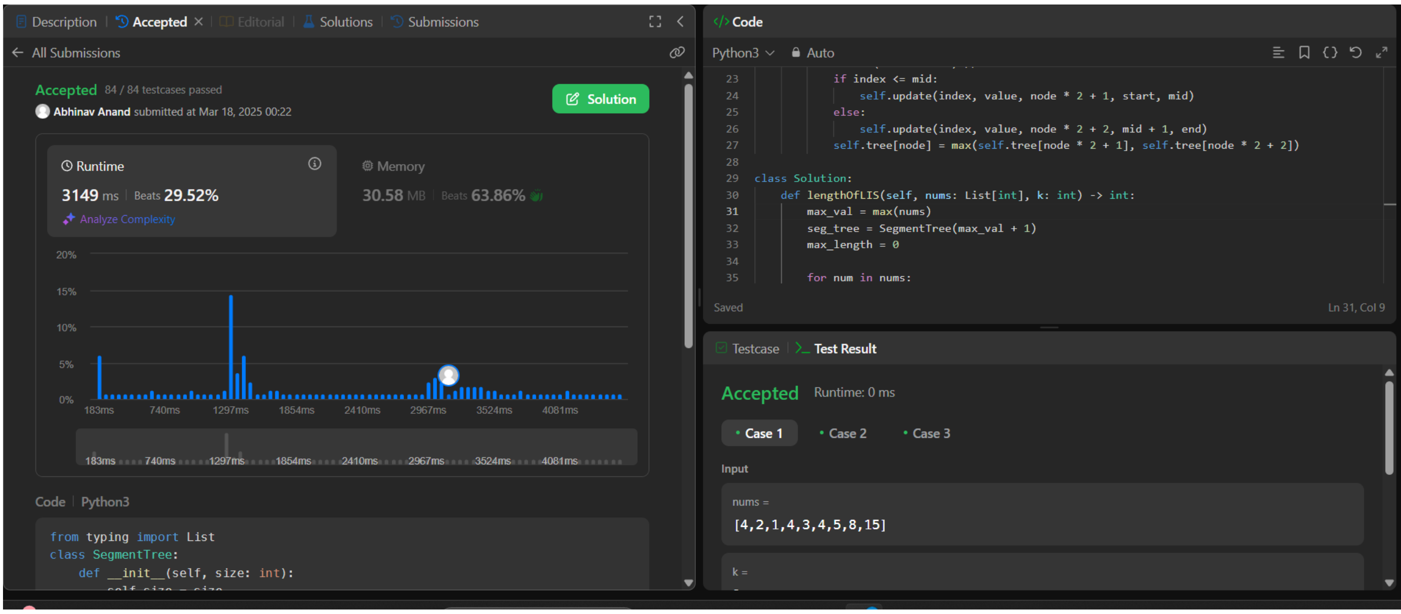
Output

Ques 10: Longest Increasing Subsequence II

Code: -

```
class Solution {
public:
    vector<int>tree;
    void update(int node,int st,int end,int i,int val){
        if(st==end){
            tree[node]=max(tree[node],val);
            return;
        }
        int mid=(st+end)/2;
        if(i<=mid){
            update(node*2,st,mid,i,val);
        }else{
            update(node*2+1,mid+1,end,i,val);
        }
        tree[node]=max(tree[node*2],tree[node*2+1]);
    }
    int query(int node,int st,int end,int x,int y){
        if(x>end || y<st) return -1e9;
        if(st>=x && end<=y){
            return tree[node];
        }
        int mid=(st+end)/2;
        int left=query(2*node,st,mid,x,y);
        int right=query(2*node+1,mid+1,end,x,y);
        return max(left,right);
    }
    int lengthOfLIS(vector<int>& nums, int k) {
        int n=nums.size();
        if(n==1) return 1;
        int m=*max_element(nums.begin(),nums.end());
        tree.clear();
        tree.resize(4*m+10);
        for(int i=n-1;i>=0;i--){
            int l=nums[i]+1,r=min(nums[i]+k,m);
            int x=query(1,0,m,l,r);
            if(x==-1e9) x=0;
            update(1,0,m,nums[i],x+1);
        }
        return tree[1];
    }
};
```

Submission: -



Accepted 84 / 84 testcases passed

Abhinav Anand submitted at Mar 18, 2025 00:22

Runtime

3149 ms | Beats 29.52%

Analyze Complexity

Memory

30.58 MB | Beats 63.86%

Code | Python3

```
from typing import List
class SegmentTree:
    def __init__(self, size: int):
```

```
            if index <= mid:
                self.update(index, value, node * 2 + 1, start, mid)
            else:
                self.update(index, value, node * 2 + 2, mid + 1, end)
            self.tree[node] = max(self.tree[node * 2 + 1], self.tree[node * 2 + 2])

class Solution:
    def lengthOfLIS(self, nums: List[int], k: int) -> int:
        max_val = max(nums)
        seg_tree = SegmentTree(max_val + 1)
        max_length = 0

        for num in nums:
```

Saved

Ln 31, Col 9

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

nums =

[4,2,1,4,3,4,5,8,15]

k =