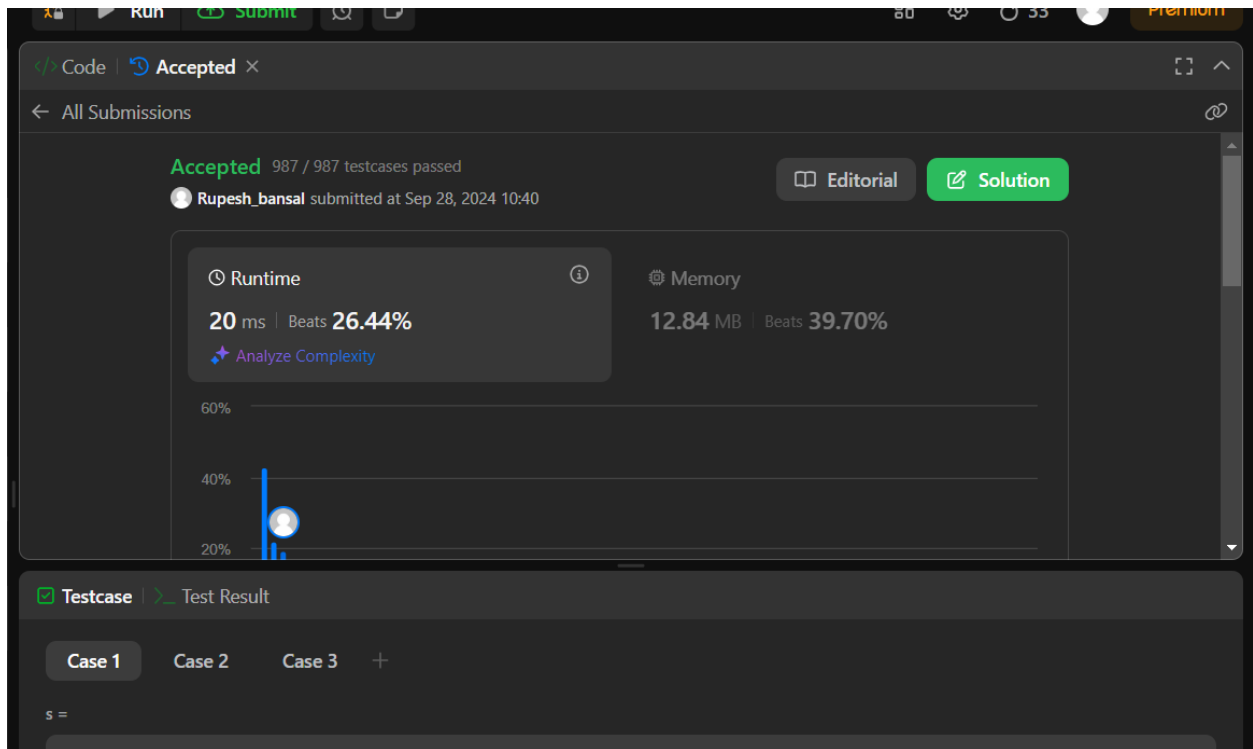


1763. [Longest Nice Substring](#)

```
class Solution {
public:
    string longestNiceSubstring(string s) {
        if (s.size() < 2) return "";
        unordered_set<char> st(begin(s), end(s));
        for (int i = 0; i < s.size(); i++) {
            if (st.find((char) toupper(s[i])) == end(st) || st.find((char) tolower(s[i])) == end(st)) {
                string s1 = longestNiceSubstring(s.substr(0, i));
                string s2 = longestNiceSubstring(s.substr(i + 1));
                return s1.size() >= s2.size() ? s1 : s2;
            }
        }
        return s;
    }
};
```



190.Reverse Bits

```
class Solution {  
public:  
    uint32_t reverseBits(uint32_t n) {  
        string bits = bitset<32>(n).to_string();  
        reverse(bits.begin(), bits.end());  
        int ans = stoll(bits, NULL, 2);  
        return ans;  
    }  
};
```

Accepted 11511 / 11511 testcases passed
 Rupesh_bansal submitted at Sep 27, 2023 23:06

Editorial Solution

Runtime: 7 ms | Beats 16.16%
 Memory: 6.54 MB | Beats 100.00%

Analyze Complexity

60%
 40%
 20%

ult

```

1 class Solution {
2 public:
3     uint32_t reverseBits(uint32_t n) {
4         string bits = bitset<32>(n).to_string();
5         reverse(bits.begin(), bits.end());
6
7         int ans = stoll(bits, NULL, 2);
8         return ans;
9     }
10 }
11 
```

Saved Ln 10, Col 6

Testcase Test Result

Accepted Runtime: 1 ms

Case 1 Case 2

Input

n =
 00000010100101000001111010011100

Output

00110010111000010100000000000000

88. Merge Sorted Array

```
class Solution {
```

```
public:
```

```
void merge(vector<int>& nums1, int m, vector<int>& nums2, int n) {
```

```
    int i=m-1;
```

```
    int j=n-1;
```

```
    int k=m+n-1;
```

```
    while(j>=0 && i>=0){
```

```
        if(nums1[i]>nums2[j]){
```

```
            nums1[k]=nums1[i];
```

```
            i--;
```

```

        k--;
    }
    else{
        nums1[k]=nums2[j];

        k--;

        j--;
    }
}

while(j>=0){
    nums1[k]=nums2[j];

    j--;

    k--;
}

}

};

```

The screenshot shows a C++ IDE with a solution for merging two sorted arrays. The code is as follows:

```

1 class Solution {
2 public:
3     void merge(vector<int>& nums1, int m, vector<int>& nums2, int n) {
4         int i=m-1;
5         int j=n-1;
6         int k=m+n-1;
7         while(j>=0 && i>=0){
8             if(nums1[i]>nums2[j]){
9                 nums1[k]=nums1[i];
10                i--;
11            }
12            else{
13                nums1[k]=nums2[j];
14                j--;
15            }
16            k--;
17        }
18    }
19 };

```

The test case shows the following input:

```

nums1 = [1, 2, 3, 0, 0, 0]
m = 3

```

372. [Super Pow](#)

```

class Solution {

    const int base = 1337;

    int powmod(int a, int k)

```

```

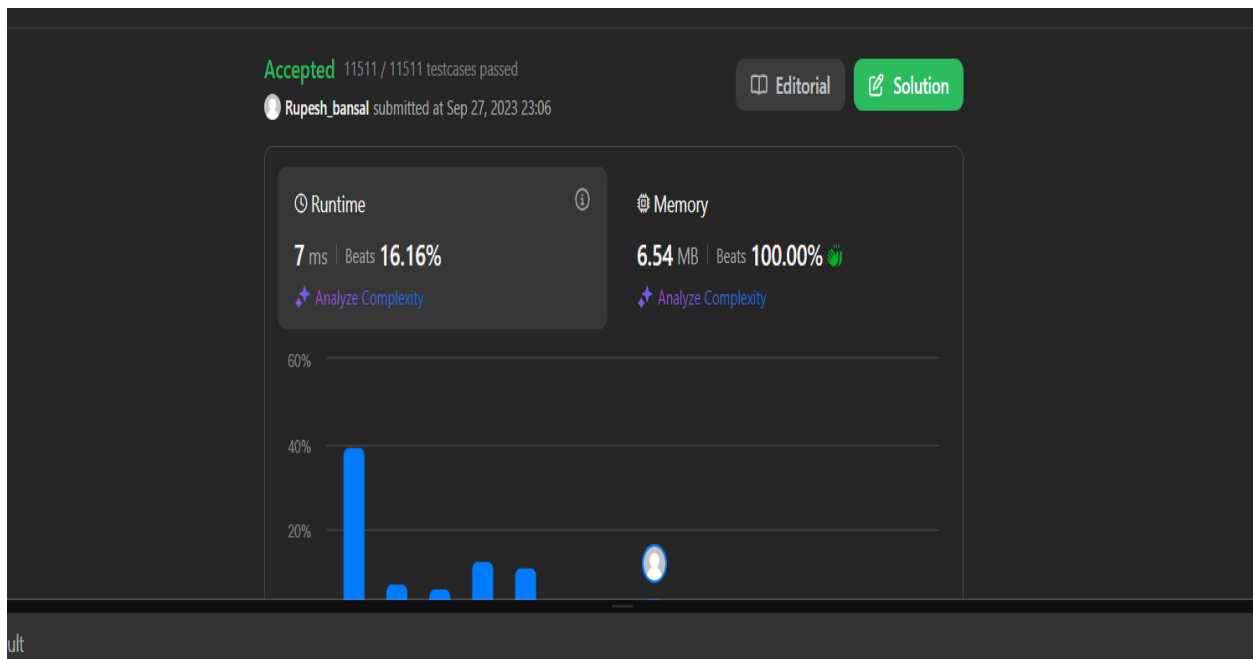
{
    a %= base;

    int result = 1;
    for (int i = 0; i < k; ++i)
        result = (result * a) % base;
    return result;
}

public:

int superPow(int a, vector<int>& b) {
    if (b.empty()) return 1;
    int last_digit = b.back();
    b.pop_back();
    return powmod(superPow(a, b), 10) * powmod(a, last_digit) % base;
}
};

```



75. [Sort Colors](#)

```
class Solution {
```

public:

```
void sortColors(vector<int>& nums) {  
    int low = 0, mid = 0, high = nums.size()-1;  
    while(mid <= high){  
        if(nums[mid] == 0){  
            swap(nums[low], nums[mid]);  
            low++;  
            mid++;  
        }  
        else if(nums[mid] == 1){  
            mid++;  
        }  
        else{  
            swap(nums[mid], nums[high]);  
            high--;  
        }  
    }  
};
```

The screenshot shows a C++ solution for sorting an array of 0s, 1s, and 2s. The code is written in a C++ IDE and is being tested. The test results show that the solution is accepted and passes all test cases.

Code:

```
1 class Solution {  
2 public:  
3     void sortColors(vector<int>& nums) {  
4         int low = 0, mid = 0, high = nums.size()-1;  
5         while(mid <= high){  
6             if(nums[mid] == 0){  
7                 swap(nums[low], nums[mid]);  
8                 low++;  
9                 mid++;  
10            }  
11            else if(nums[mid] == 1){  
12                mid++;  
13            }  
14            else{  
15                swap(nums[mid], nums[high]);  
16                high--;  
17            }  
18        }  
19    }  
20 };
```

Test Results:

- Runtime: 0 ms
- Memory: 1.42 MB
- Beats: 100.00%
- Accepted
- Case 1: [2,0,2,1,1,0]
- Case 2: [2,0,2,1,1,0]

53. Maximum Subarray

```

class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int maxSum = INT_MIN;
        int currentSum = 0;
        for (int i = 0; i < nums.size(); i++) {
            currentSum += nums[i];
            if (currentSum > maxSum) {
                maxSum = currentSum;
            }
            if (currentSum < 0) {
                currentSum = 0;
            }
        }
        return maxSum;
    }
};

```

The screenshot displays a LeetCode solution for the "Maximum Subarray" problem. The interface includes a "Problem List" tab, a "Run" button, and a "Submit" button. The solution is written in C++ and is marked as "Accepted".

Performance Metrics:

- Runtime: 0 ms (Beats 100.00%)
- Memory: 76 MB (Beats 52.58%)

Test Results:

- Accepted
- Runtime: 0 ms
- Case 1, Case 2, Case 3

Input:

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

Output:

```
4
```

The code in the editor is as follows:

```

1 class Solution {
2 public:
3     int maxSubArray(vector<int>& nums) {
4         int maxSum = INT_MIN;
5         int currentSum = 0;
6         for (int i = 0; i < nums.size(); i++) {
7             currentSum += nums[i];
8             if (currentSum > maxSum) {
9                 maxSum = currentSum;
10            }
11            if (currentSum < 0) {
12                currentSum = 0;
13            }
14        }
15        return maxSum;
16    }
17 }

```

278.[First Bad Version](#)

```
class Solution {
public:
    int firstBadVersion(int n) {
        int first = 1;
        int last = n;
        while (first < last) {
            int mid = first + (last - first) / 2;
            if (isBadVersion(mid)) {
                last = mid;
            } else {
                first = mid + 1;
            }
        }
        return first;
    }
};
```

347.[Top K Frequent Elements](#)

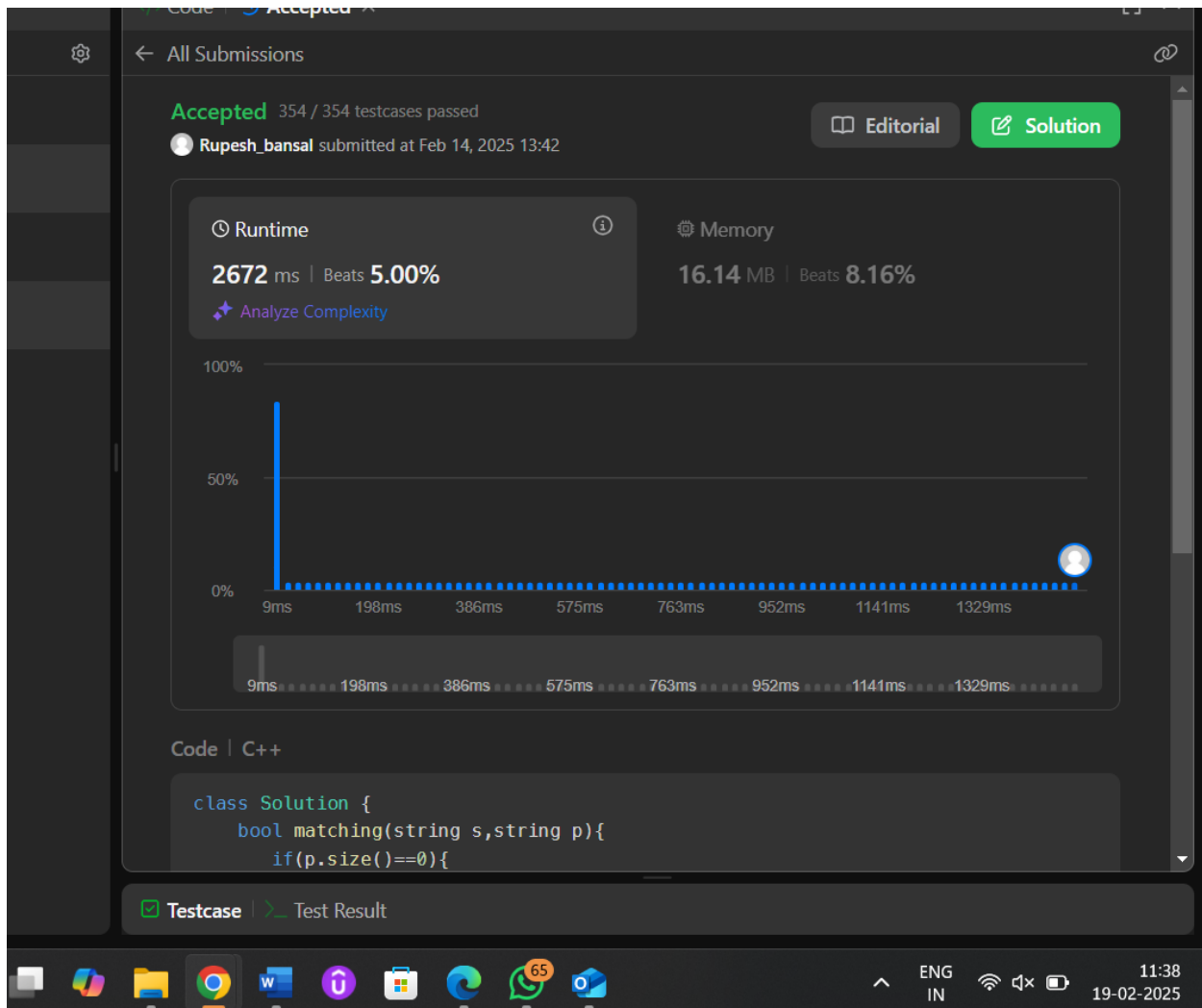
```
class Solution {
public:
    vector<int> topKFrequent(vector<int>& nums, int k) {
        unordered_map<int, int> ump;
        for(int i: nums){
            ump[i]++;
        }
        priority_queue<pair<int, int>> pq;
```



```

for(auto i: ump){
    pq.push({i.second,i.first});
}
vector<int> res;
while(k--){
    auto [elem, count] = pq.top();
    res.push_back(count);
    pq.pop();}
return res;
}
};

```



932. [Beautiful Array](#)

```

class Solution {
public:
int partition(vector<int> &v, int start, int end, int mask)
{
    int j = start;
    for(int i = start; i <= end; i++)
    {
        if((v[i] & mask) != 0)
        {
            swap(v[i], v[j]);
            j++;
        }
    }
    return j;
}

```

```

void sort(vector<int> &v, int start, int end, int mask)
{
    if(start >= end) return;
    int mid = partition(v, start, end, mask);
    sort(v, start, mid - 1, mask << 1);
    sort(v, mid, end, mask << 1);
}

```

```

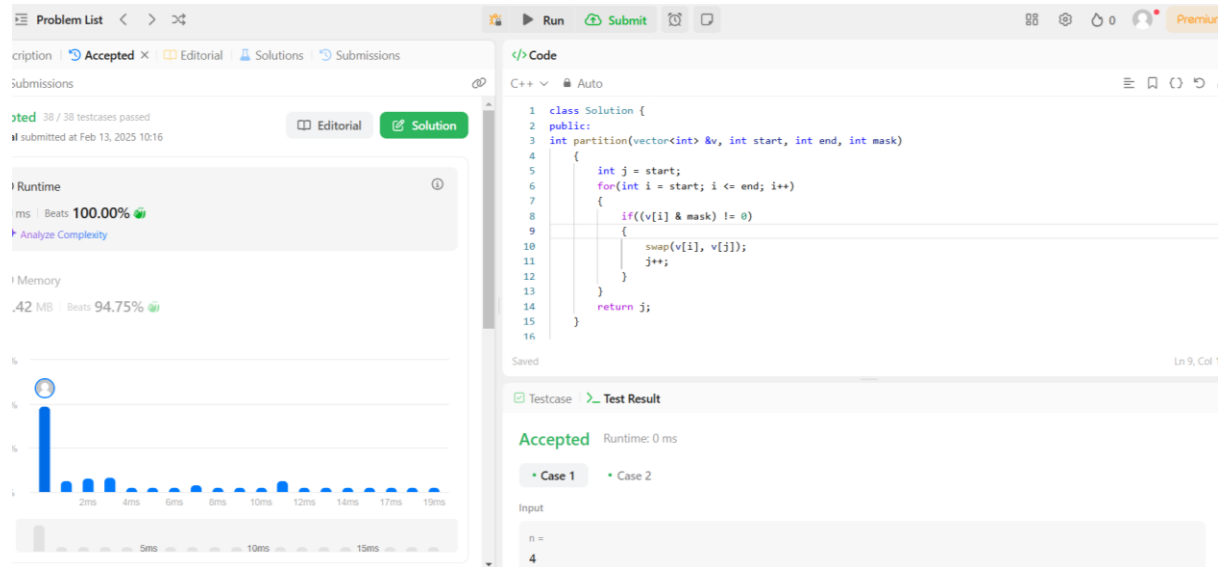
vector<int> beautifulArray(int n) {
    vector<int> ans;
    for(int i = 0; i < n; i++) ans.push_back(i + 1);
    sort(ans, 0, n - 1, 1);
    return ans;
}

```

```

    }
};

```



162. Find Peak Element

```


class Solution {
public:
    int findPeakElement(vector<int>& nums) {
        int n = nums.size();
        int r = n-1;
        int l = 0;
        while(r>l){
            int mid = (r+l)/2;
            if(nums[mid] > nums[mid+1]) r = mid;
            else l = mid + 1;
        }
        return l;
    }
}

```

};



[← All Submissions](#)

Accepted 48 / 48 testcases passed


 **Rupesh_bansal** submitted at Sep 17, 2024 22:07


[Editorial](#)


[Solution](#)

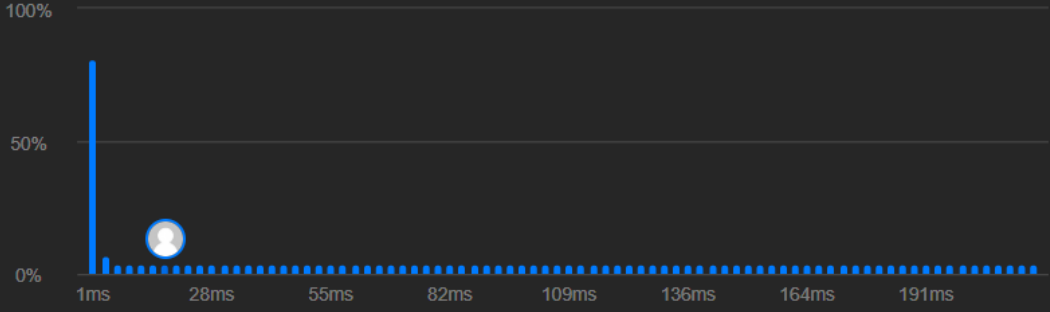
 **Runtime** 

19 ms | Beats **9.56%**

 [Analyze Complexity](#)

 **Memory**

19.77 MB | Beats **100.00%** 



100%
50%
0%

1ms 28ms 55ms 82ms 109ms 136ms 164ms 191ms

Code | C++

```
class Solution {  
public:  
    int candy(vector<int>& ratings) {
```

☒ **Testcase** | [Test Result](#)