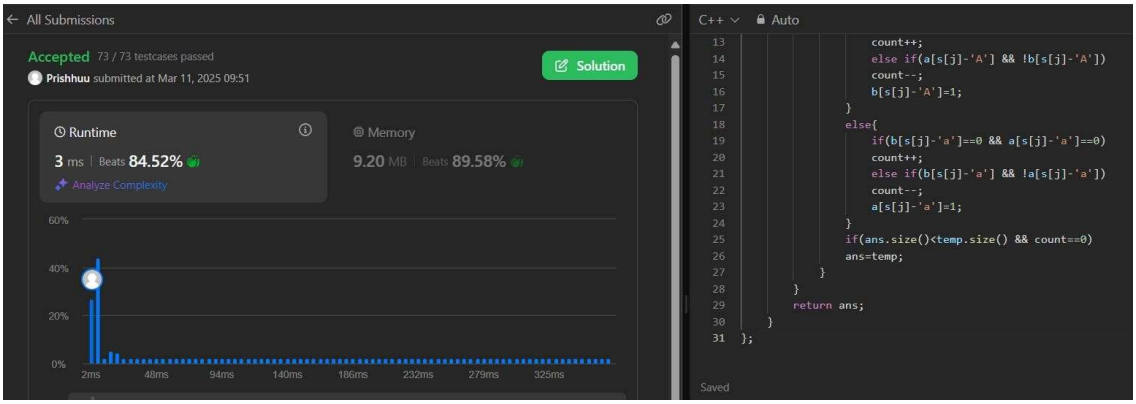
Manjot Singh 22BCS14912

1. 1763. Longest Nice Substring

Solution:

```cpp
class Solution {
public:
    string longestNiceSubstring(string s) {
        string ans="";
        for(int i=0;i<s.length();i++){
            int count=0;
            string temp="";
            vector<bool> a(26,0),b(26,0);
            for(int j=i;j<s.length();j++){
                temp.push_back(s[j]);
                if(s[j]>='A' && s[j]<='Z'){
                    if(a[s[j]-'A']==0 && b[s[j]-'A']==0)
                    count++;
                    else if(a[s[j]-'A'] && !b[s[j]-'A'])
                    count--;
                    b[s[j]-'A']=1;
                }
                else{
                    if(b[s[j]-'a']==0 && a[s[j]-'a']==0)
                    count++;
                    else if(b[s[j]-'a'] && !a[s[j]-'a'])
                    count--;
                    a[s[j]-'a']=1;
                }
                if(ans.size()<temp.size() && count==0)
                ans=temp;
            }
        }
        return ans;
    }
};
```

Screenshot:

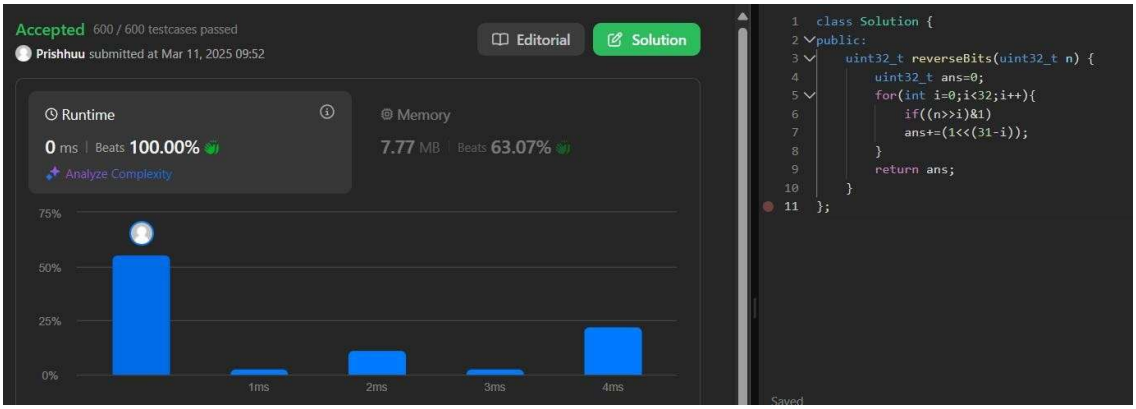


2. 190.Reverse Bits

Solution:

```
class Solution {
public:
    uint32_t reverseBits(uint32_t n)
        { uint32_t ans=0; for(int
        i=0;i<32;i++){ if((n>>i)&1)
        ans+=(1<<(31-i));
        } return
        ans;
    }
};
```

Screenshot:



3. 191.Number of 1 Bits

Solution:

```
class Solution {
public:
    int hammingWeight(int n) {
        int count=  builtin popcount(n);
        return count;
    }
};
```

Screenshot:

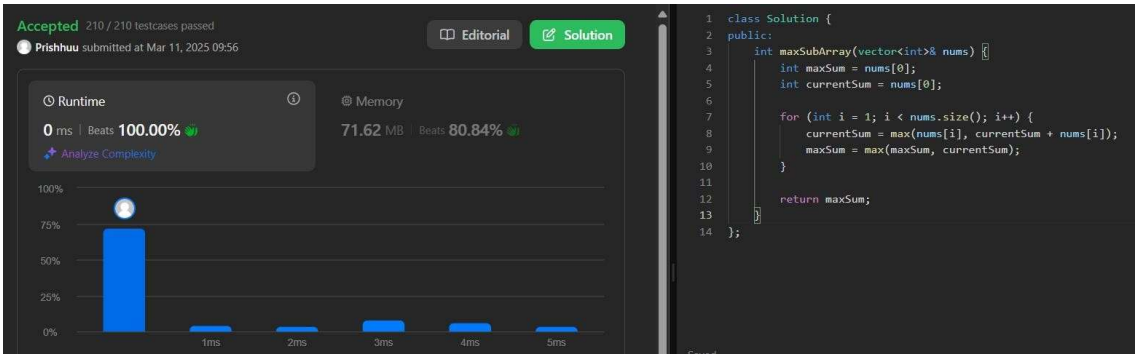
4. 53.Maximum Subarray

Solution:

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int maxSum = nums[0]; int
        currentSum = nums[0];

        for (int i = 1; i < nums.size(); i++) { currentSum
            = max(nums[i], currentSum + nums[i]); maxSum =
            max(maxSum, currentSum);
        } return
        maxSum;
    }
};
```

Screenshot:



5. 240.Search a 2D Matrix

II Solution:

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int cols = matrix[0].size() - 1; int n = matrix.size()
        - 1; int rows = 0;

        while(rows <= n && cols >= 0){ int
            toCompare = matrix[rows][cols];
            if(toCompare > target){ cols--;
            }else if(toCompare < target){
                rows++;
            }else{ return
                true;
            } }
        return
        false;
    }
};
```

Screenshot:

6. 372.Super Pow

Solution:

```
class Solution {
    const int base = 1337;
    int powmod(int a, int k) //a^k mod 1337 where 0 <= k <= 10
    {
        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;
    }
};
```

Screenshot: