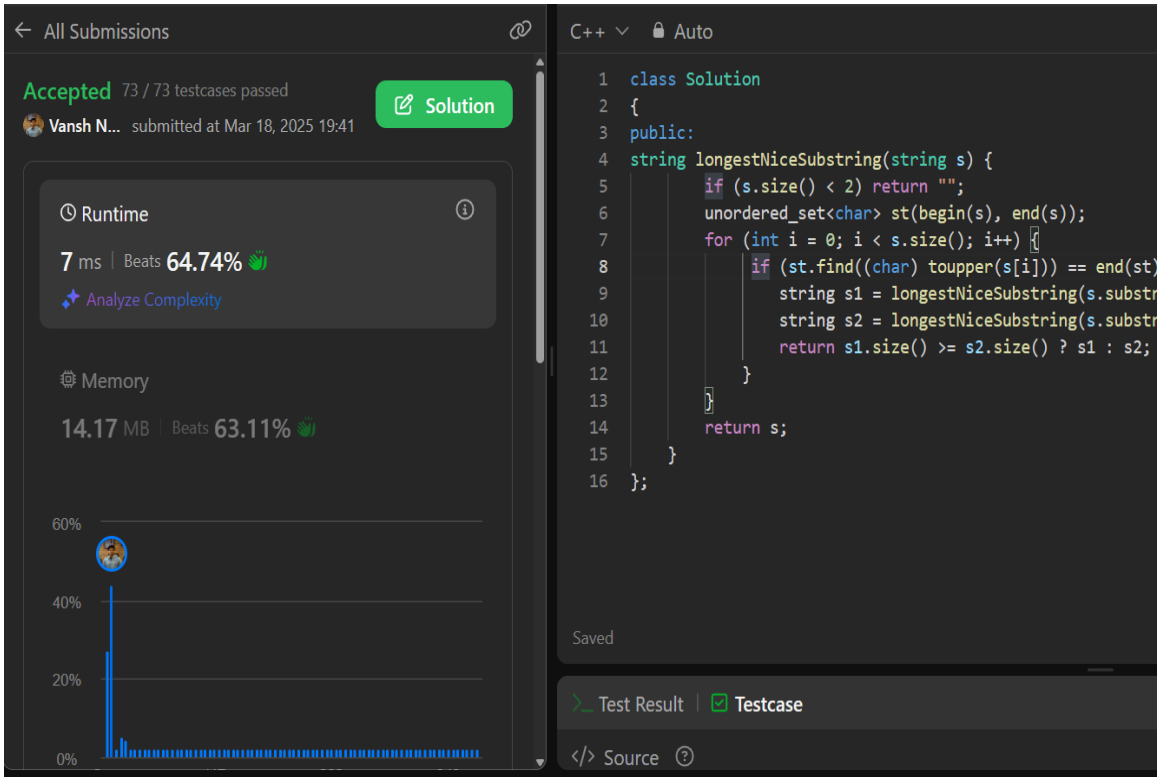
1763.Longest Nice Substring

- **Solution Code:**

```
class Solution {
public:
    string longestNiceSubstring(string s) {
        string output = "";
        int count = 0;
        for(int i = 0;i<s.length();i++){
            int smallMask=0;
            int largeMask = 0;
            char ch = s[i];
            int chint = 0;
            if(ch>=65 && ch<=90){
                chint = ch-'A';
                largeMask = 1<<chint;
            }
            else{
                chint = ch-'a';
                smallMask = 1<<chint;
            }
            for(int j = i+1;j<s.length();j++){
                ch = s[j];
                if(ch>=65 && ch<=90){
                    chint = ch-'A';
                    largeMask |= 1<<chint;
                }
                else{
                    chint = ch-'a';
                    smallMask |= 1<<chint;
                }
                //checking for nice
                if((smallMask^largeMask) == 0){
                    if(count<j-i+1){
                        count = j-i+1;
                        string
temp(s.begin()+i,s.begin()+j+1);
                        output = temp;
                    }
                }
            }
        }
        return output;
    }
};
```

- **Screenshot:**


All Submissions
Accepted 73 / 73 testcases passed
Vansh N... submitted at Mar 18, 2025 19:41
Solution
Runtime
7 ms | Beats 64.74%
Analyze Complexity
Memory
14.17 MB | Beats 63.11%
60%
40%
20%
0%
C++
Auto
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)
string s1 = longestNiceSubstring(s.substr
string s2 = longestNiceSubstring(s.substr
return s1.size() >= s2.size() ? s1 : s2;
}
}
return s;
}
};
Saved
Test Result
Testcase
Source

190.Reverse Bits

- **Solution Code:**

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

- **Screenshot:**

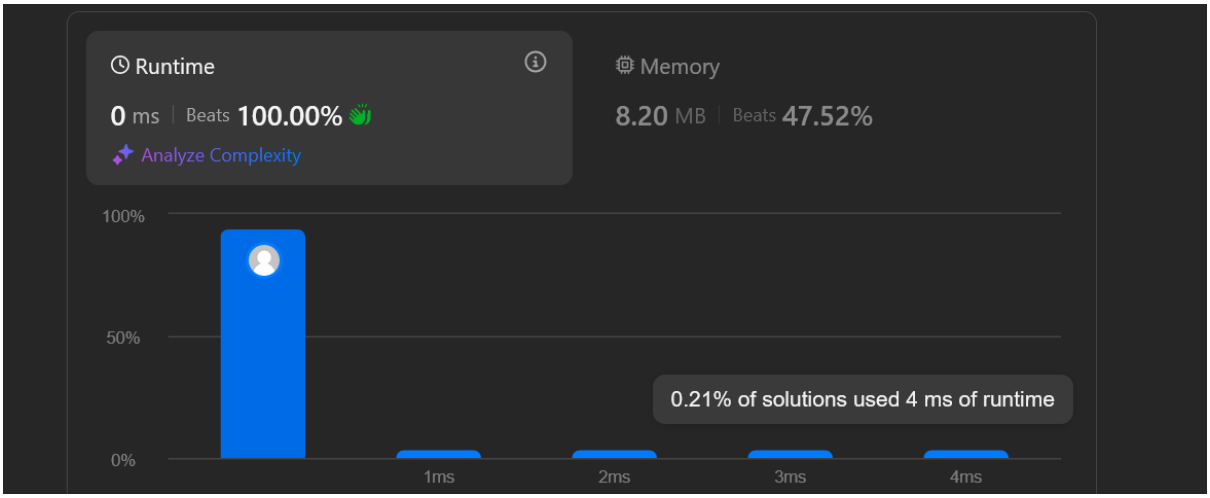
191.Number of 1 Bits

- **Solution Code:**

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

- **Screenshot:**

\



53.Maximum Subarray

- **Solution Code:**

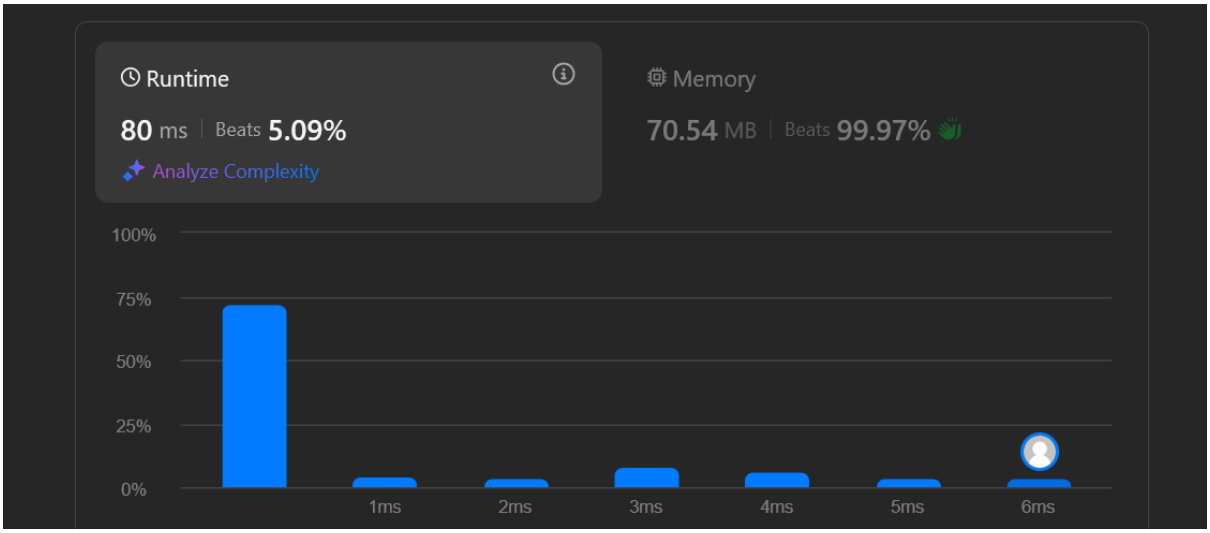
```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int res = nums[0];
        int total = 0;

        for (int n : nums) {
            if (total < 0) {
                total = 0;
            }

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

        return res;
    }
};
```

- **Screenshot:**



240.Search a 2D Matrix II

- **Solution Code:**

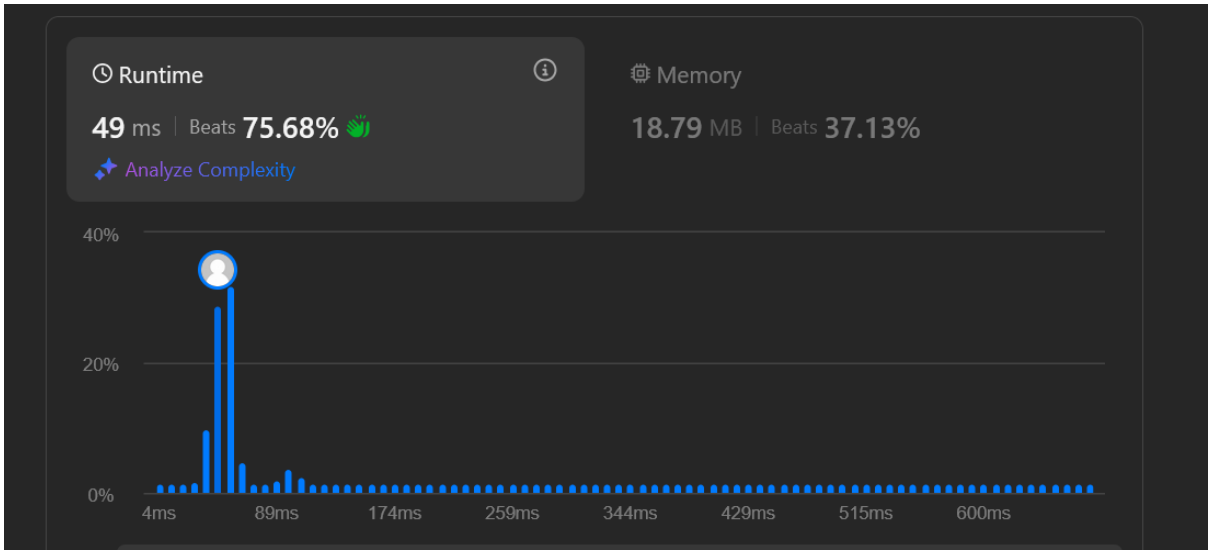
```
const auto _ = std::cin.tie(nullptr)->sync_with_stdio(false);

#define LC_HACK
#ifdef LC_HACK
const auto __ = []() {
    struct ___ {
        static void _() { std::ofstream("display_runtime.txt") << 0 <<
'\n'; }
    };
    std::atexit(&___::_);
    return 0;
}();
#endif


#define pb push_back
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int n = matrix.size(), m = matrix[0].size();
        int row = 0, col = m-1;
        while(row < n && col >= 0){
            if(matrix[row][col] == target)return true;

            else if(matrix[row][col] < target){
                row++;
            }
            else{
                col--;
            }
        }
        return false;
    }
};
```

- **Screenshot:**


Runtime
49 ms | Beats 75.68%
Analyze Complexity
Memory
18.79 MB | Beats 37.13%
40%
20%
0%
4ms
89ms
174ms
259ms
344ms
429ms
515ms
600ms

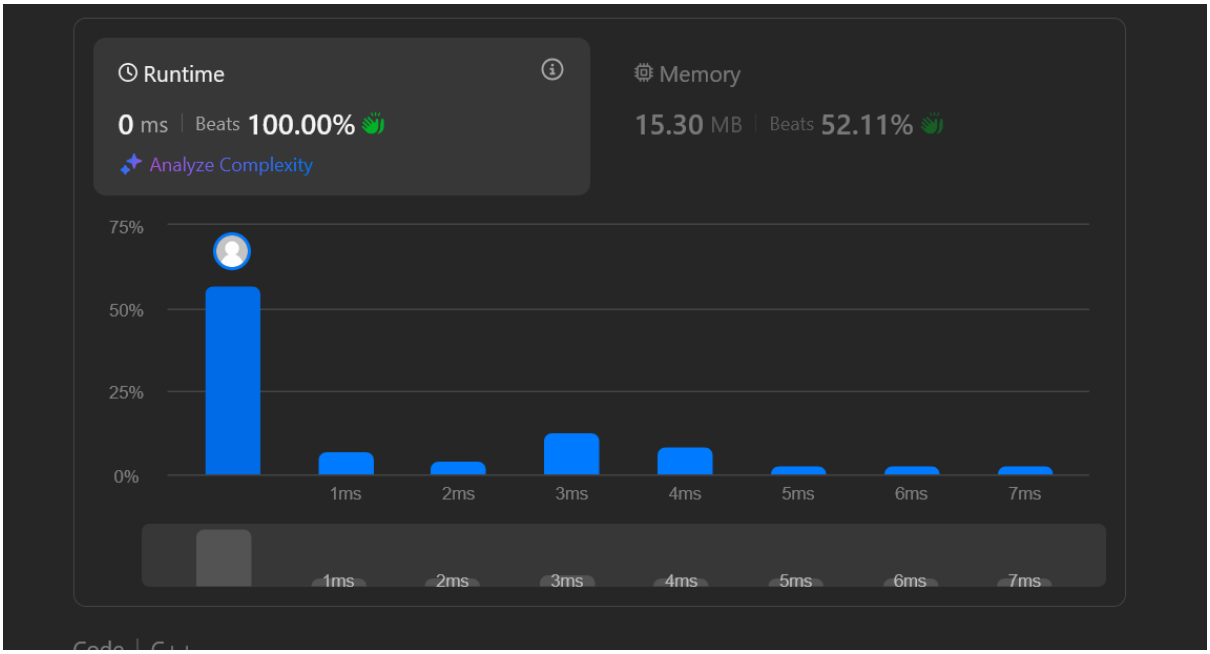
372.Super Pow

- **Solution 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);
    }
};
```

- **Screenshot:**


Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
15.30 MB | Beats 52.11%
75%
50%
25%
0%
1ms
2ms
3ms
4ms
5ms
6ms
7ms