AP Assignment-4

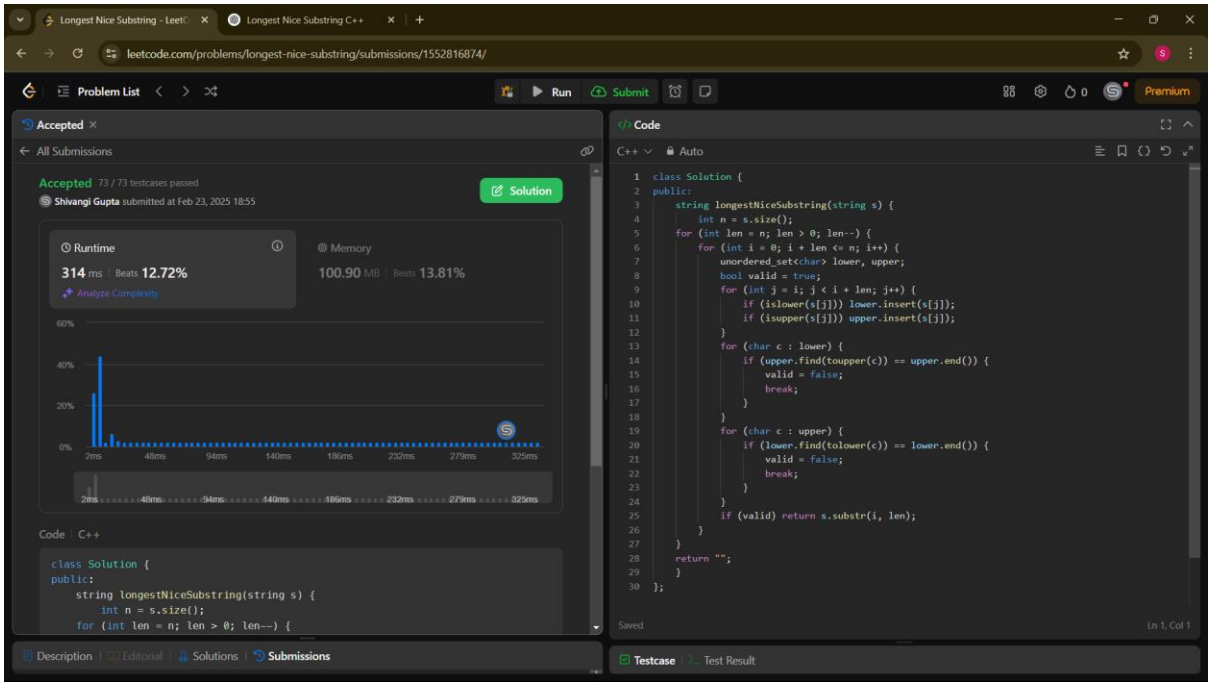
Name: Shivangi Gupta **UID: 22BCS15008** **Class: IoT_601-A**

1763. https://leetcode.com/problems/longest-nice-substring/description/

```
class Solution {
public:
    string longestNiceSubstring(string s) {
        int n = s.size();
    for (int len = n; len > 0; len--) {
        for (int i = 0; i + len <= n; i++) {
            unordered_set<char> lower, upper;
            bool valid = true;
            for (int j = i; j < i + len; j++) {
                if (islower(s[j])) lower.insert(s[j]);
                if (isupper(s[j])) upper.insert(s[j]);
            }
            for (char c : lower) {
                if (upper.find(toupper(c)) == upper.end()) {
                    valid = false;
                    break;
                }
            }
            for (char c : upper) {
                if (lower.find(tolower(c)) == lower.end()) {
                    valid = false;
                    break;
                }
            }
            if (valid) return s.substr(i, len);
        }
    }
```

```
    return "";
    }
};
```

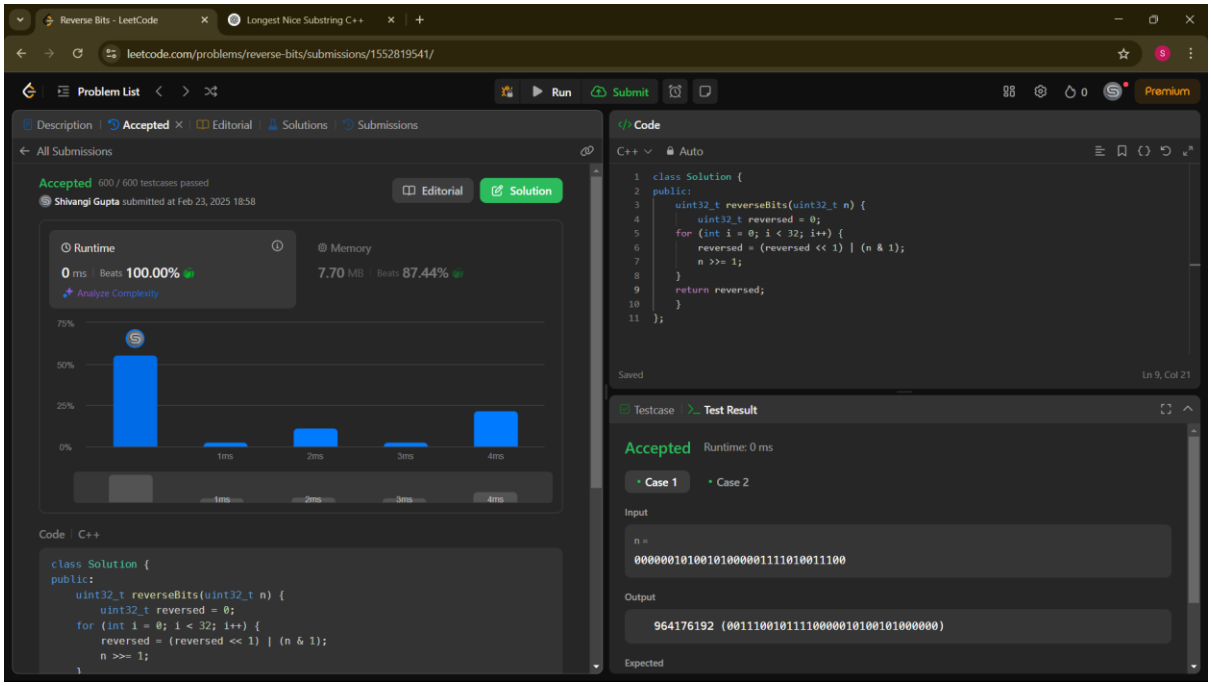
Code:



190. https://leetcode.com/problems/reverse-bits/

Code:

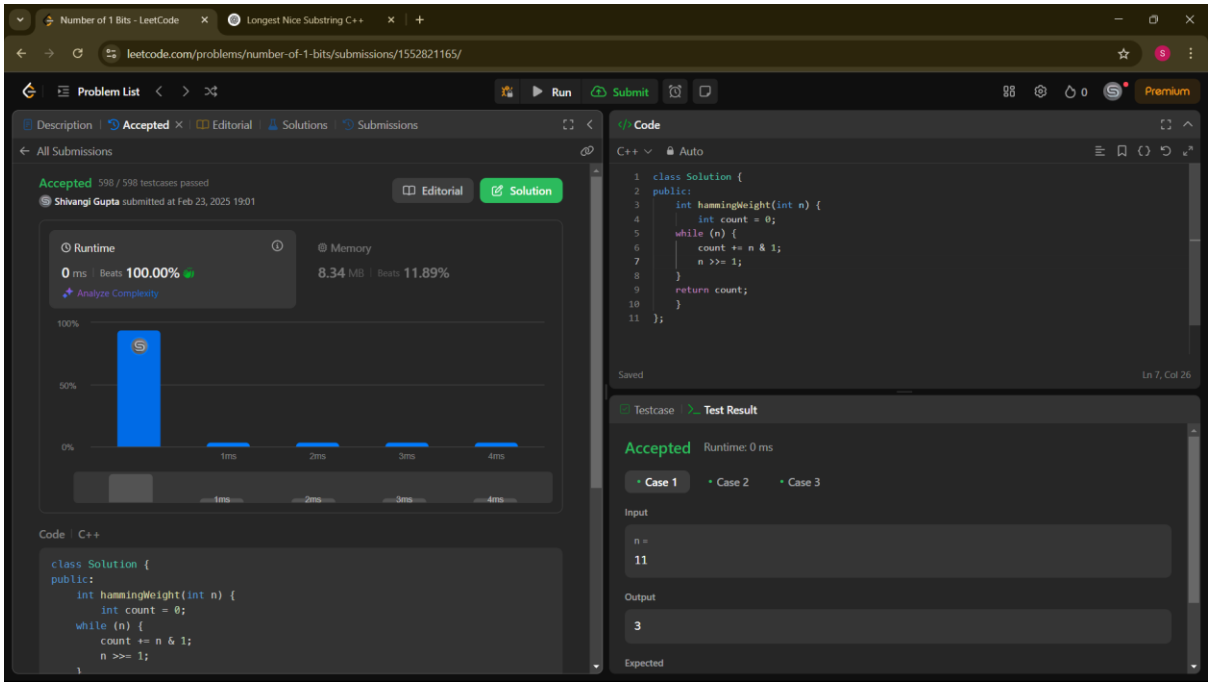
```
class Solution {
public:
    uint32_t reverseBits(uint32_t n) {
        uint32_t reversed = 0;
    for (int i = 0; i < 32; i++) {
        reversed = (reversed << 1) | (n & 1);
        n >>= 1;
    }
    return reversed;
    }
};
```



191. Number of 1-bits https://leetcode.com/problems/number-of-1-bits/description/

Code:

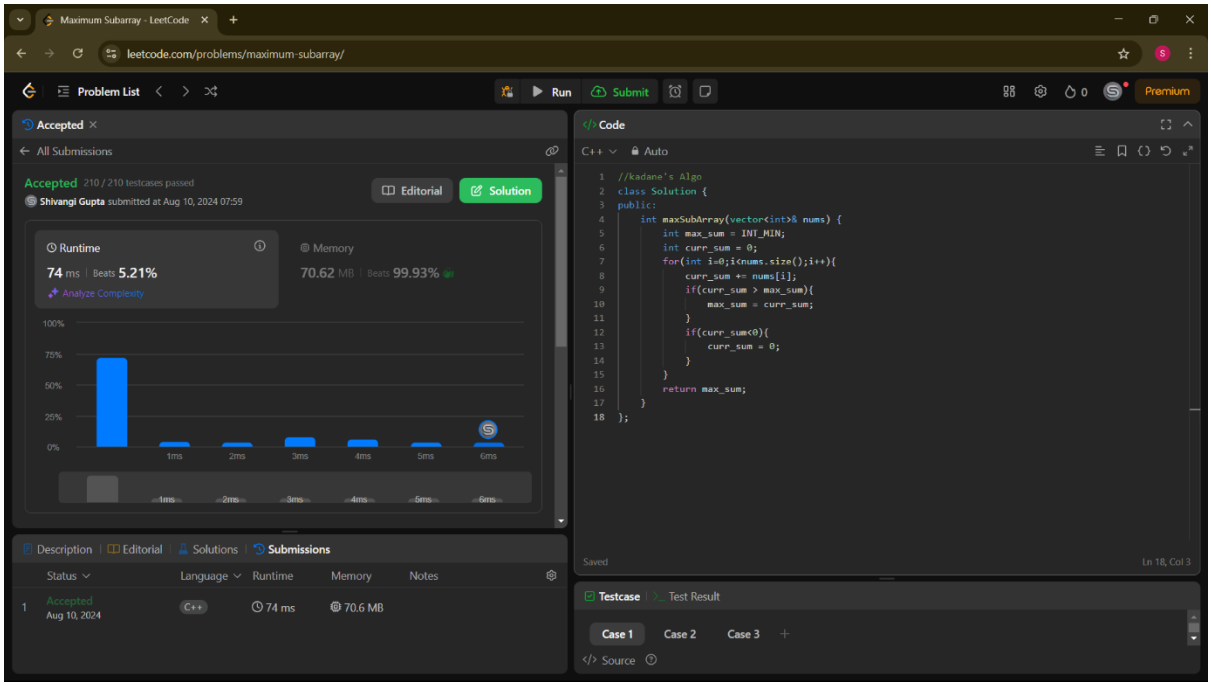
```
class Solution {
public:
    int hammingWeight(int n) {
        int count = 0;
    while (n) {
        count += n & 1;
        n >>= 1;
    }
    return count;
    }
};
```



53. https://leetcode.com/problems/maximum-subarray/description/

Code:

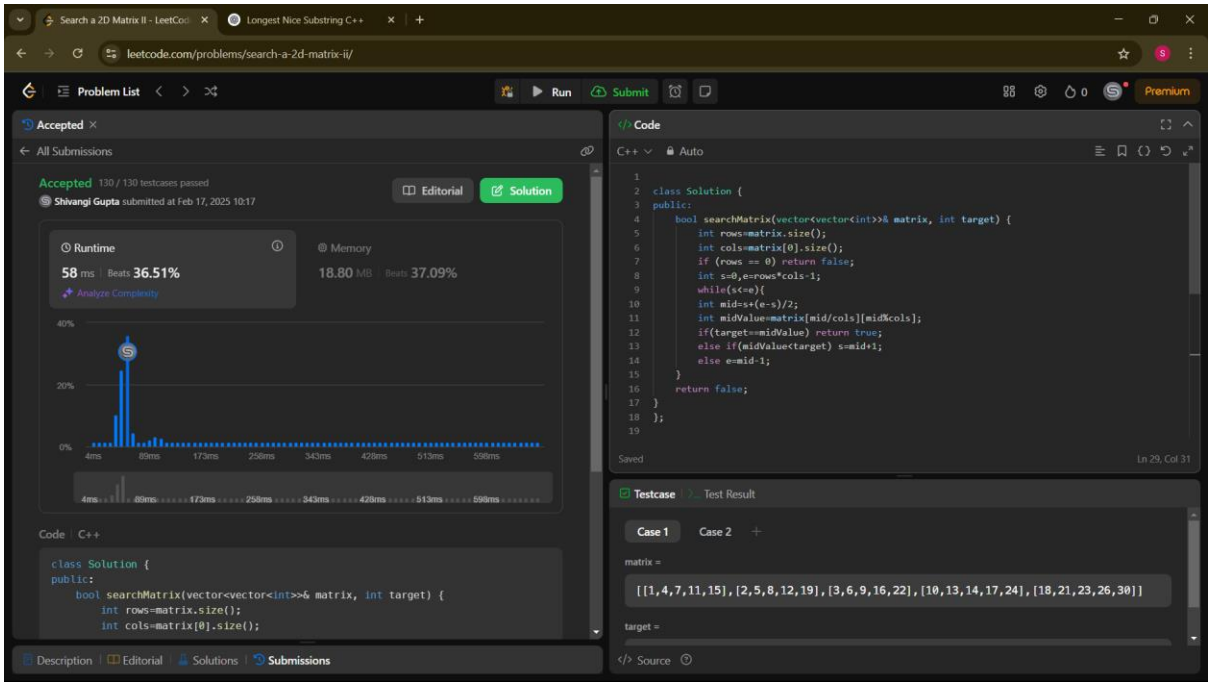
```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int max_sum = INT_MIN;
        int curr_sum = 0;
        for(int i=0;i<nums.size();i++){
            curr_sum += nums[i];
            if(curr_sum > max_sum){
                max_sum = curr_sum;
            }
            if(curr_sum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



240. Search a 2D Matrix-II https://leetcode.com/problems/search-a-2d-matrix-ii/description/

Code:

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int rows=matrix.size();
        int cols=matrix[0].size();
        if (rows == 0) return false;
        int s=0,e=rows*cols-1;
        while(s<=e){
        int mid=s+(e-s)/2;
        int midValue=matrix[mid/cols][mid%cols];
        if(target==midValue) return true;
        else if(midValue<target) s=mid+1;
        else e=mid-1;
    }
    return false;
}
};
```



372. Super Pow https://leetcode.com/problems/super-pow/description/

Code:

```
class Solution {
public:
int mod = 1337;


int quickPow(int a, long long b) {
        int result = 1;
        a = a % mod;
        while (b > 0) {
            if (b % 2 == 1) {
                result = (result * a) % mod;
            }
            a = (a * a) % mod;
            b /= 2;
        }
        return result;
    }
    int superPow(int a, vector<int>& b) {
```

```
        long long exponent = 0;
        for (int i = 0; i < b.size(); i++) {
            exponent = (exponent * 10 + b[i]) % (mod - 1);
        }
        return quickPow(a, exponent);
    }
};
```