ASSIGNMENT - 4 (ADVANCED PROGRAMMING)

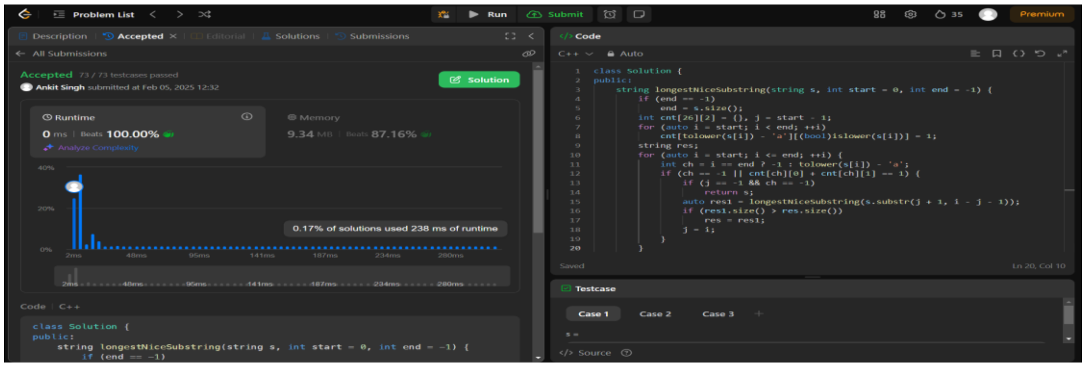
Leetcode: https://leetcode.com/u/AnkitSingh101/

1. Problem 1: Longest Nice Substring (1763)

2. Implementation/Code:

```cpp
class Solution {
public:
    string longestNiceSubstring(string s, int start = 0, int end = -1) {
        if (end == -1)
            end = s.size();
        int cnt[26][2] = {}, j = start - 1;
        for (auto i = start; i < end; ++i)
            cnt[tolower(s[i]) - 'a'][(bool)islower(s[i])] = 1;
        string res;
        for (auto i = start; i <= end; ++i) {
            int ch = i == end ? -1 : tolower(s[i]) - 'a';
            if (ch == -1 || cnt[ch][0] + cnt[ch][1] == 1) {
                if (j == -1 && ch == -1)
                    return s;
                auto res1 = longestNiceSubstring(s.substr(j + 1, i - j - 1));
                if (res1.size() > res.size())
                    res = res1;
                j = i;
            }
        }
        return res;
    }
};
```

Output:



Accepted 73 / 73 testcases passed
Ankit Singh submitted at Feb 05, 2025 12:32

Runtime
0 ms | Beats 100.00%

Memory
9.34 MB | Beats 87.16%

0.17% of solutions used 238 ms of runtime

Code | C++

```cpp
class Solution {
public:
    string longestNiceSubstring(string s, int start = 0, int end = -1) {
        if (end == -1)
```

C++ Auto

```cpp
class Solution {
public:
    string longestNiceSubstring(string s, int start = 0, int end = -1) {
        if (end == -1)
            end = s.size();
        int cnt[26][2] = {}, j = start - 1;
        for (auto i = start; i < end; ++i)
            cnt[tolower(s[i]) - 'a'][(bool)islower(s[i])] = 1;
        string res;
        for (auto i = start; i <= end; ++i) {
            int ch = i == end ? -1 : tolower(s[i]) - 'a';
            if (ch == -1 || cnt[ch][0] + cnt[ch][1] == 1) {
                if (j == -1 && ch == -1)
                    return s;
                auto res1 = longestNiceSubstring(s.substr(j + 1, i - j - 1));
                if (res1.size() > res.size())
                    res = res1;
                j = i;
            }
        }
```

Saved
Ln 20, Col 10

s =

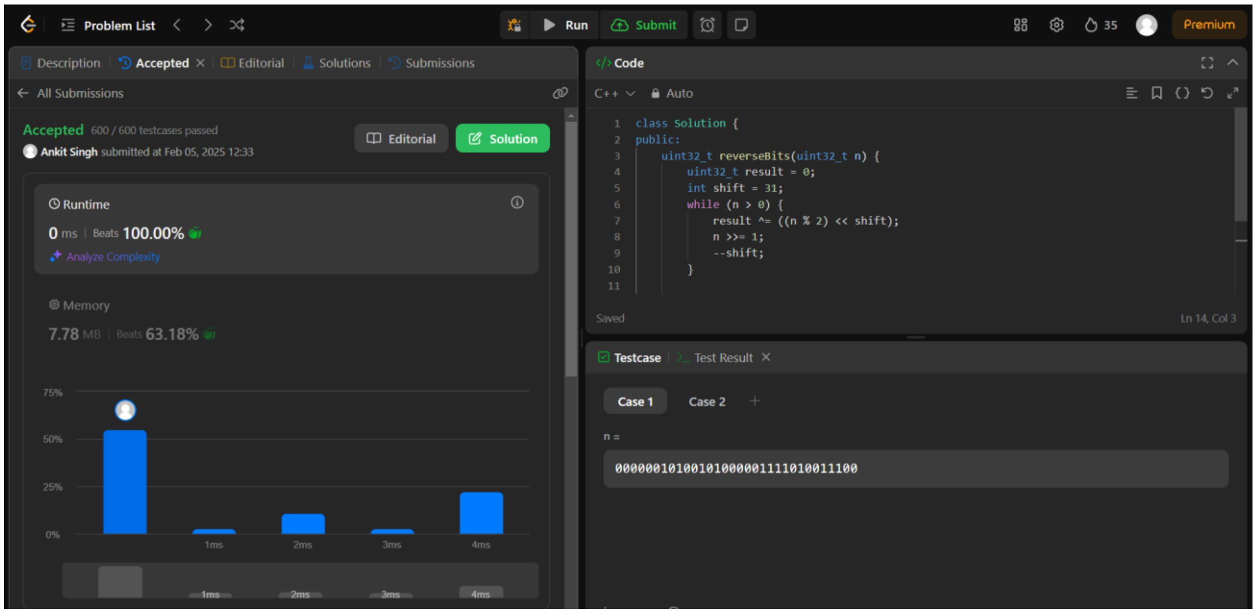
1. **Problem 2: Reverse Bits (190)**

2. **Implementation/Code:**

```cpp
class Solution {
public:
    uint32_t reverseBits(uint32_t n) {
        uint32_t result = 0;
        int shift = 31;
        while (n > 0) {
            result ^= ((n % 2) << shift);
            n >>= 1;
            --shift;
        }

        return result;
    }
};
```

3. **Output:**



Accepted 600 / 600 testcases passed
Ankit Singh submitted at Feb 05, 2025 12:33

Runtime
0 ms | Beats 100.00%
Analyze Complexity

Memory
7.78 MB | Beats 63.18%

```cpp
class Solution {
public:
    uint32_t reverseBits(uint32_t n) {
        uint32_t result = 0;
        int shift = 31;
        while (n > 0) {
            result ^= ((n % 2) << shift);
            n >>= 1;
            --shift;
        }

```

Saved
Ln 14, Col 3

Testcase | Test Result
Case 1   Case 2

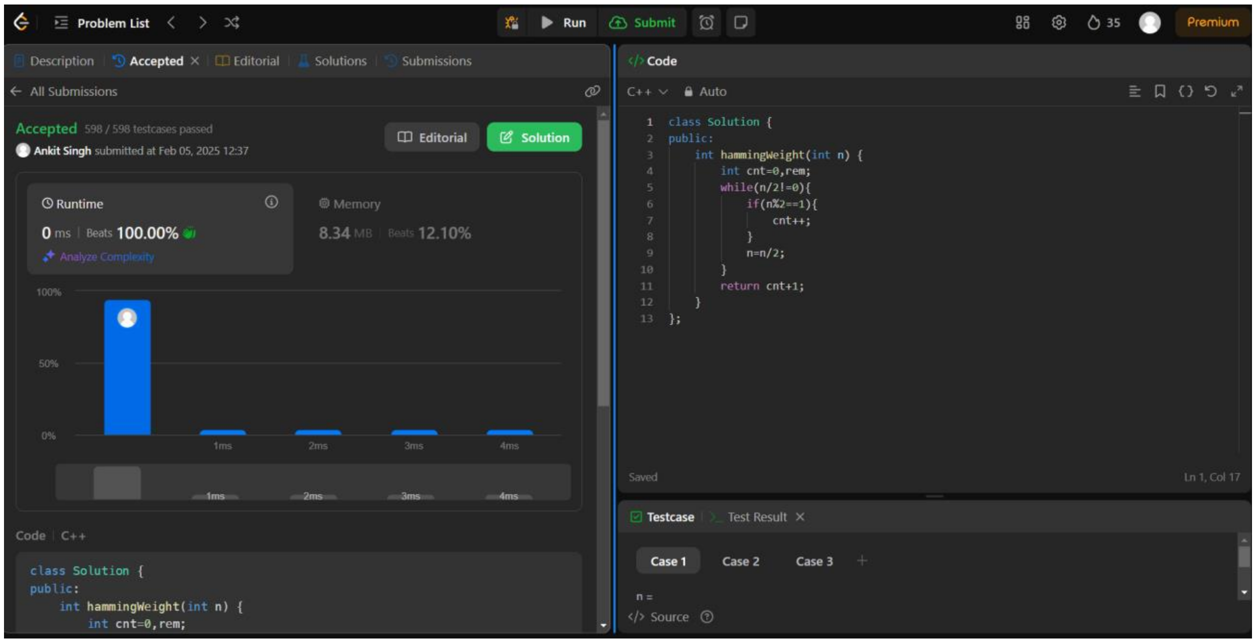
n =
```
00000010100101000001111010011100
```

1. Problem 3: Number of 1 Bits (191)

2. Implementation/code:

```
class Solution {
public:
    int hammingWeight(int n) {
        int cnt=0,rem;
        while(n/2!=0){
            if(n%2==1){
                cnt++;
            }
            n=n/2;
        }
        return cnt+1;
    }
};
```

3. Output:



Accepted 598 / 598 testcases passed
Ankit Singh submitted at Feb 05, 2025 12:37

Runtime
0 ms | Beats 100.00%

Memory
8.34 MB | Beats 12.10%

Code | C++

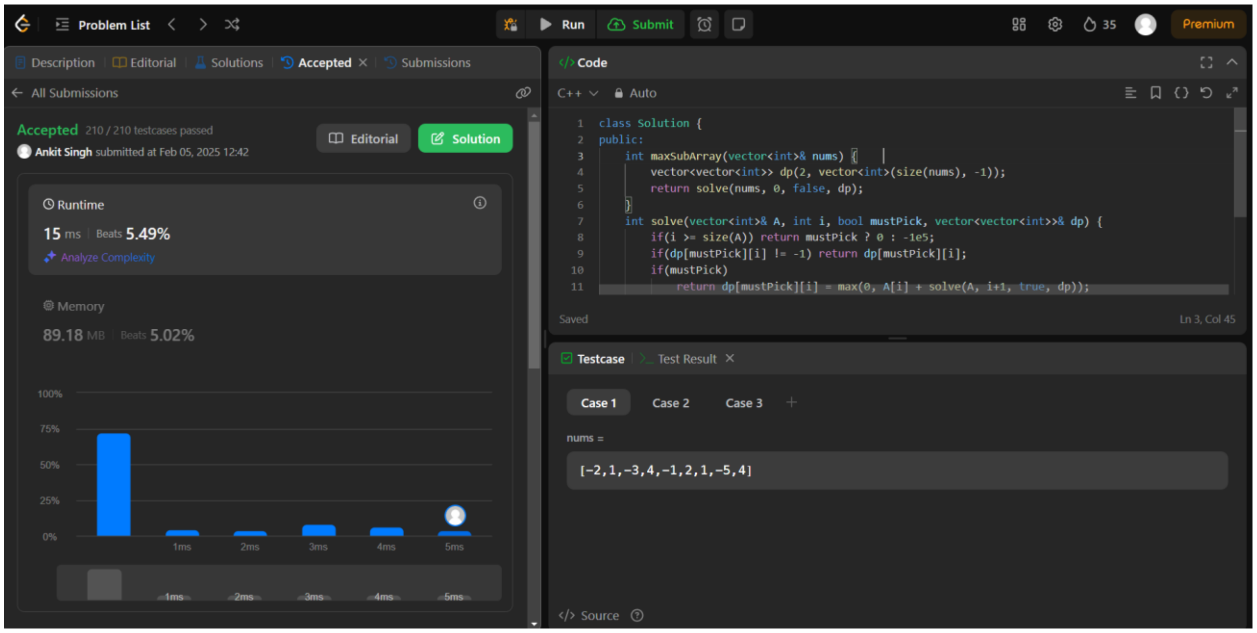
```
class Solution {
public:
    int hammingWeight(int n) {
        int cnt=0,rem;
        while(n/2!=0){
            if(n%2==1){
                cnt++;
            }
            n=n/2;
        }
        return cnt+1;
    }
};
```

1. **Problem 4: Maximum Sub array (53)**

2. **Implementation/code:**

```cpp
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        vector<vector<int>> dp(2, vector<int>(size(nums), -1));
        return solve(nums, 0, false, dp);
    }
    int solve(vector<int>& A, int i, bool mustPick, vector<vector<int>>& dp) {
        if(i >= size(A)) return mustPick ? 0 : -1e5;
        if(dp[mustPick][i] != -1) return dp[mustPick][i];
        if(mustPick)
            return dp[mustPick][i] = max(0, A[i] + solve(A, i+1, true, dp));
        return dp[mustPick][i] = max(solve(A, i+1, false, dp), A[i] + solve(A, i+1,
true, dp));
    }
};
```

3. **Output:**



Accepted 210 / 210 testcases passed
Ankit Singh submitted at Feb 05, 2025 12:42

Runtime
15 ms | Beats 5.49%

Memory
89.18 MB | Beats 5.02%

```cpp
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        vector<vector<int>> dp(2, vector<int>(size(nums), -1));
        return solve(nums, 0, false, dp);
    }
    int solve(vector<int>& A, int i, bool mustPick, vector<vector<int>>& dp) {
        if(i >= size(A)) return mustPick ? 0 : -1e5;
        if(dp[mustPick][i] != -1) return dp[mustPick][i];
        if(mustPick)
            return dp[mustPick][i] = max(0, A[i] + solve(A, i+1, true, dp));
```

Saved
Ln 3, Col 45

nums =

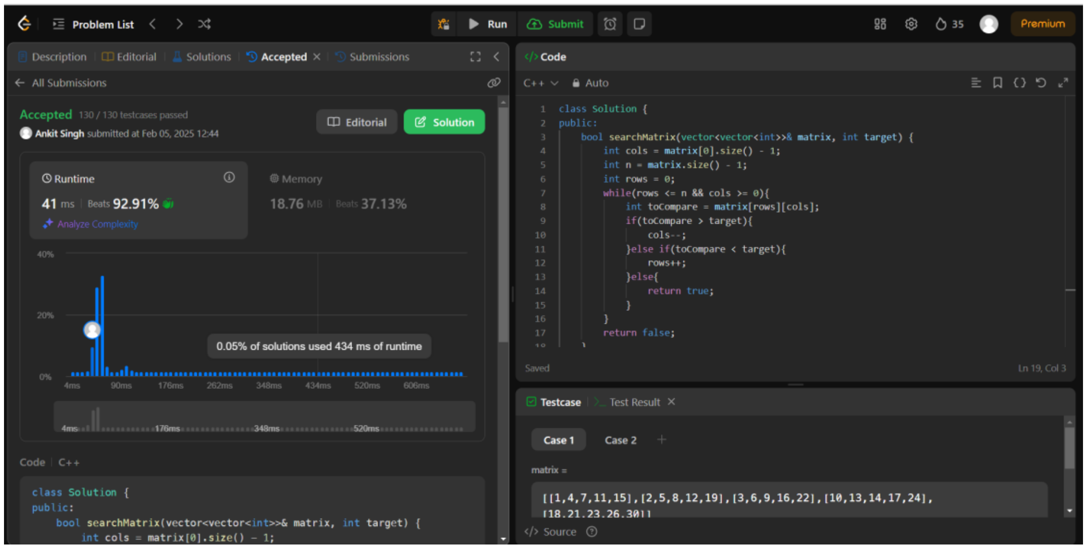
```
[-2,1,-3,4,-1,2,1,-5,4]
```

1. **Problem 5: Search a 2D Matrix II (240)**

2. **Implementation/Code:**

```cpp
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;
    }
};
```

3. **Output:**



Accepted 130 / 130 testcases passed
Ankit Singh submitted at Feb 05, 2025 12:44

Runtime
41 ms | Beats 92.91%

Memory
18.76 MB | Beats 37.13%

0.05% of solutions used 434 ms of runtime

```cpp
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;
```

Saved
Ln 19, Col 3

matrix =

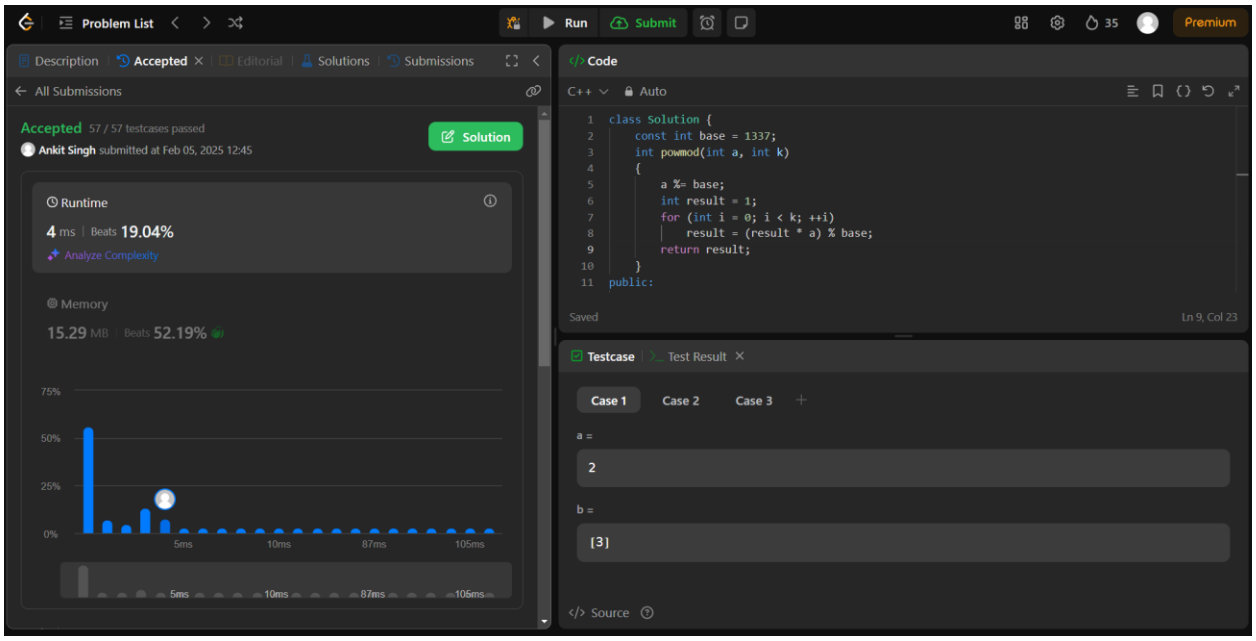
```
[[1,4,7,11,15],[2,5,8,12,19],[3,6,9,16,22],[10,13,14,17,24],
[18,21,23,26,30]]
```

1. **Problem 6: Super Pow (372)**

2. **Implementation/Code:**

```cpp
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;
    }
};
```

3. **Output:**



Accepted 57 / 57 testcases passed
Ankit Singh submitted at Feb 05, 2025 12:45

Runtime
4 ms | Beats 19.04%

Memory
15.29 MB | Beats 52.19%

```cpp
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:
```

Saved Ln 9, Col 23

a =
2

b =
[3]