Assignment -04

Student Name: Nikhil Kumar Tiwari **UID: 22BCS10471**

Branch: BE-CSE **Section/Group:22BCS-IOT-FL-601 A**

Semester: 6th **Subject Code: 22CSP-351**

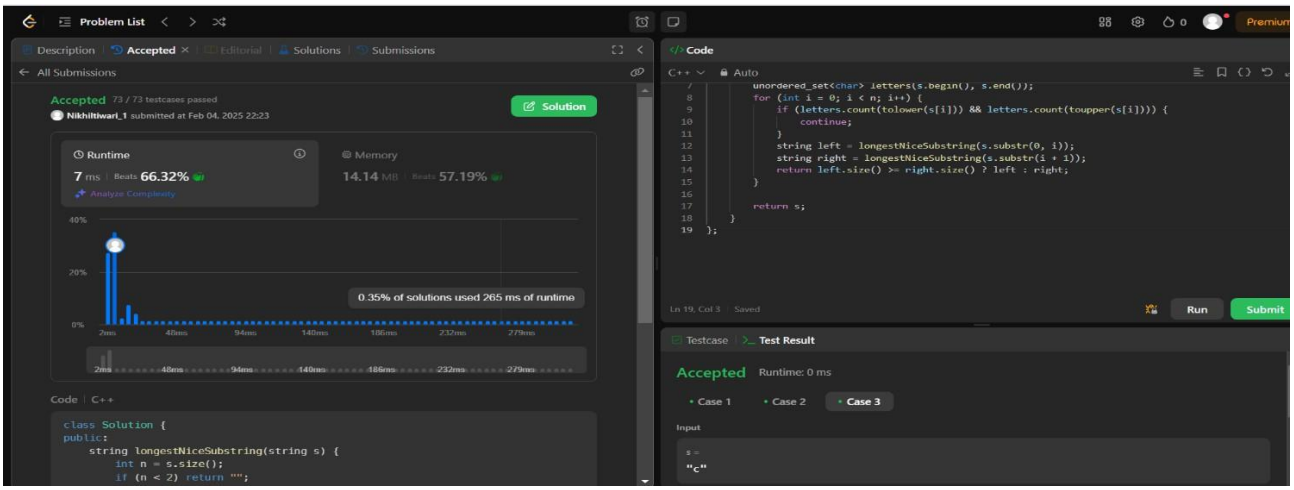
Subject Name: Advanced Programming -2

Problem 1: Longest Nice Substring { https://leetcode.com/problems/longest-nice-substring/ }

Code:

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

Screenshot:

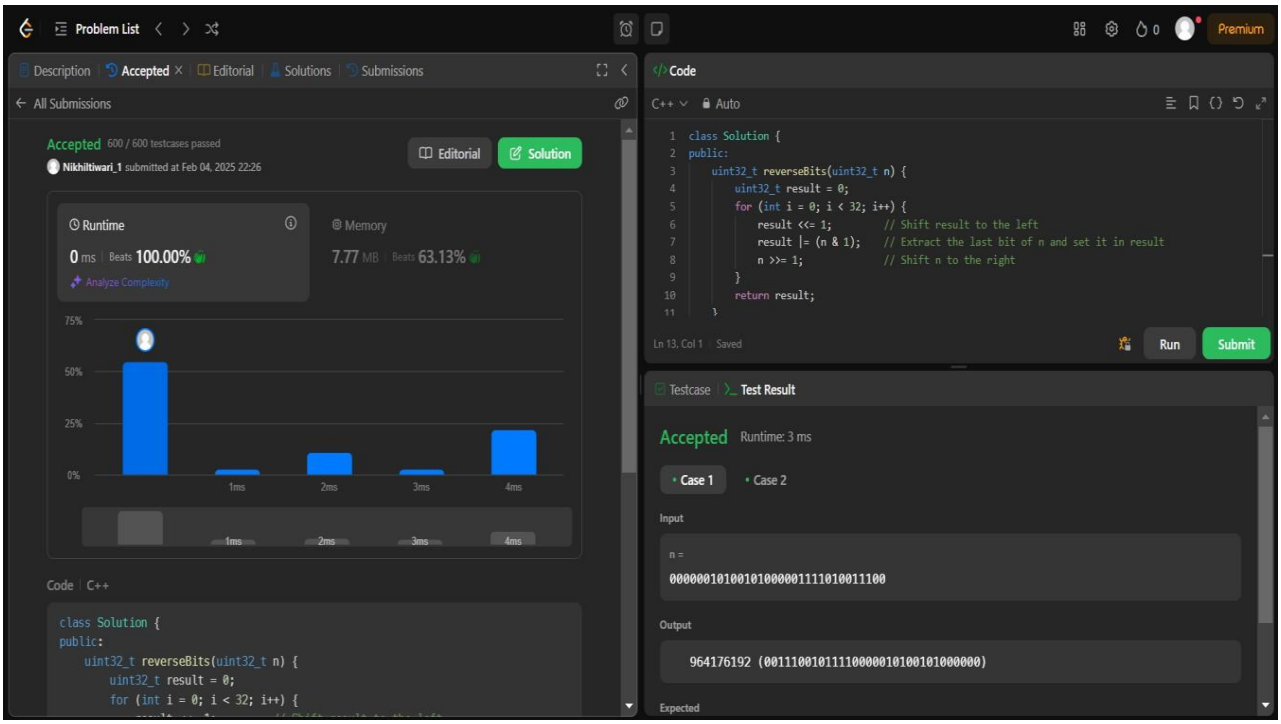


Problem 2: Reverse Bits { https://leetcode.com/problems/reverse-bits/ }

Code:

```
class Solution {

public:

  uint32_t reverseBits(uint32_t n) {

    uint32_t result = 0;

    for(int i = 0;i<32;i++) {

      uint32_t temp = n C 1;

      result = (result << 1) | temp;

      n = n >> 1;

    }

    return result;

  }

};
```

Screenshot:

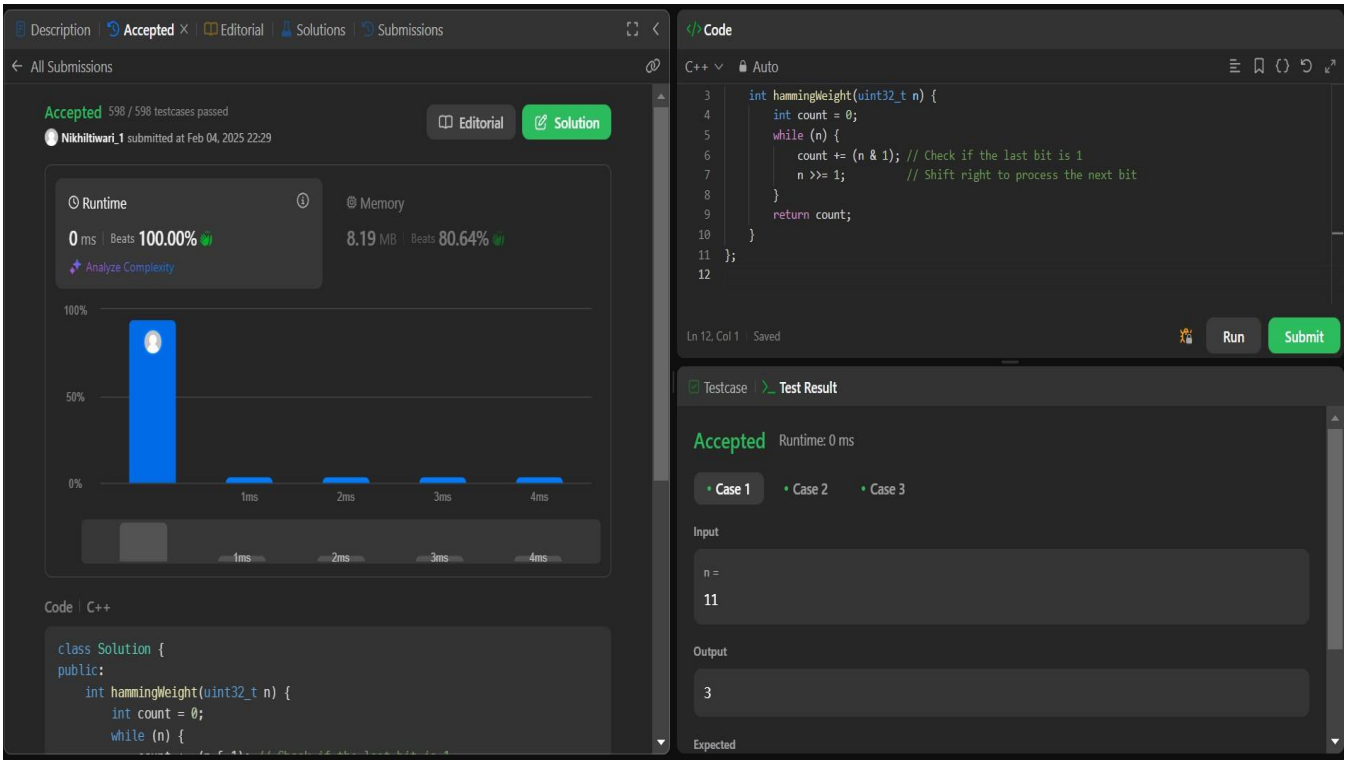


Problem 3: Number of 1 Bits { https://leetcode.com/problems/number-of-1-bits/ }

Code:

```
class Solution {

public:

  int hammingWeight(int n) {

    int count = 0;

    while(n) {

      n = n C (n-1);

      count++;

    }

    return count;

  }

};
```

Screenshot:

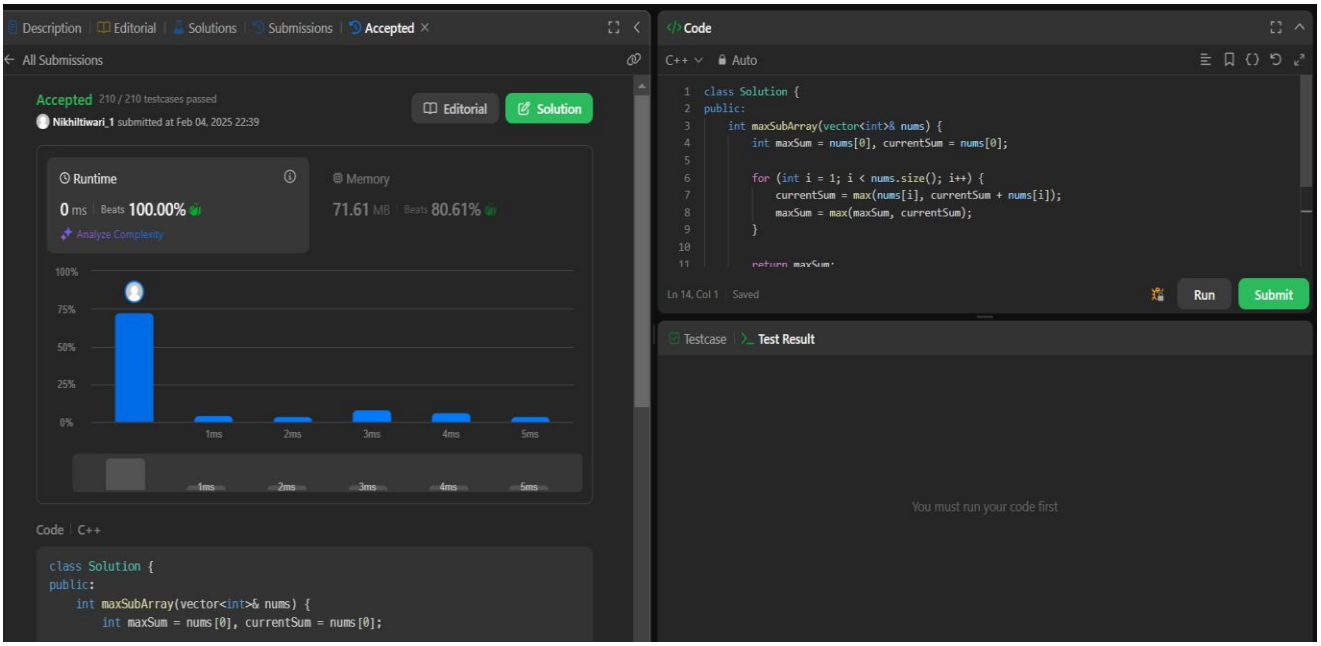


Problem 4: Maximum Subarray { https://leetcode.com/problems/maximum-subarray/ **}**

Code:

```
class Solution {

public:

  int maxSubArray(vector<int>C nums) {

    int sum = nums[0];

    int max_sum = nums[0];


    for(int i = 1;i<nums.size();i++) {

      sum = max(nums[i],sum+nums[i]);

      max_sum = max(sum,max_sum);

    }

    return max_sum;

  }

};
```

Screenshot:

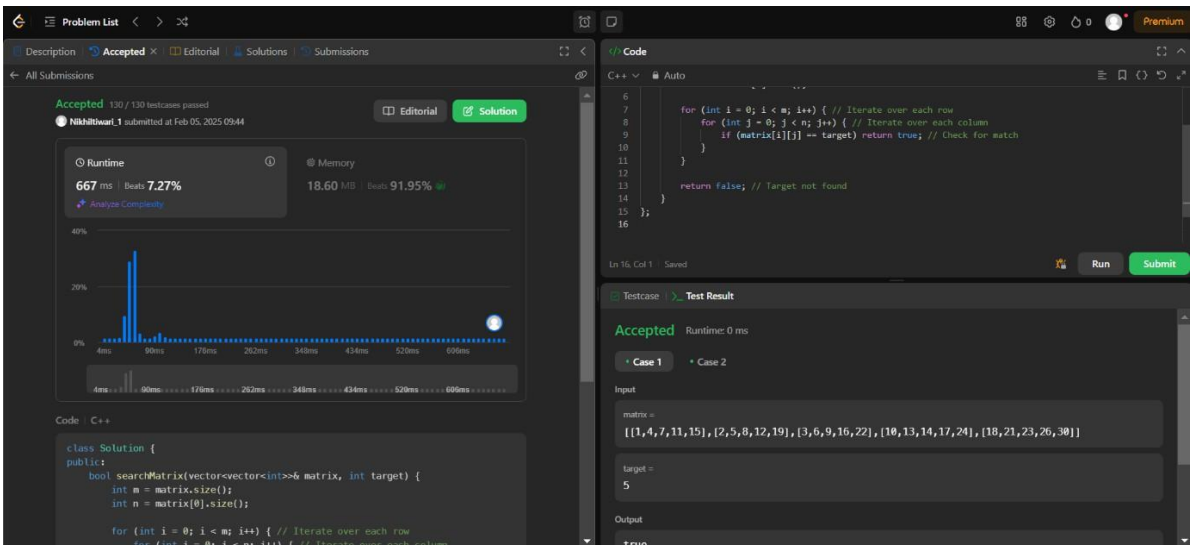


Problem 5: Search a 2D matrix { https://leetcode.com/problems/search-a-2d-matrix-ii/ }

Code:

```
class Solution {

public:

  bool searchMatrix(vector<vector<int>>C matrix, int target) {

    for(int i =0;i<matrix.size();i++) {

      for(int j = 0;j<matrix[0].size();j++) {

        if(matrix[i][j] == target) {

          return true;

        }

      }

    }

    return false;

  }

};
```

Screenshot:



Problem 6: Super Pow { https://leetcode.com/problems/super-pow/ }

Code:

```
class Solution {

public:

  const int MOD = 1337;

  int powerMod(int x, int y) {

    int result = 1;

    x %= MOD;

    while (y > 0) {

      if (y % 2 == 1) {

        result = (result * x) % MOD;

      }

      x = (x * x) % MOD;

      y /= 2;

    }

    return result;

  }

  int superPow(int a, vector<int>C b) {

    int result = 1;

    for (int digit : b) {

      result = powerMod(result, 10) * powerMod(a, digit) % MOD;

    } return result; };
```

Screenshot: