

Assignment 4

Advanced Programming Lab – II

Submitted By – Ratnesh Tiwari

UID – 22BCS17201

Section – 22BCS_IOT – 609

Submitted To – Pratima Sonali Horo (E18304)

1. Longest Nice Substring: <https://leetcode.com/problems/longest-nice-substring/description/>

```
class Solution {
private:
    bool isNice(string& str) {
        for (char c : str) {
            if (islower(c) && str.find(toupper(c)) == string::npos) {
                return false;
            }
            if (isupper(c) && str.find(tolower(c)) == string::npos) {
                return false;
            }
        }
        return true;
    }
public:
    string longestNiceSubstring(string s) {
        int n = s.length();
        if (n < 2) {
            return "";
        }

        bool lower[26] = {false};
```

```

bool upper[26] = {false};

for (char c : s) {
    if (islower(c)) {
        lower[c - 'a'] = true;
    } else {
        upper[c - 'A'] = true;
    }
}

for (int i = 0; i < n; i++) {
    char c = s[i];
    if (islower(c) && !upper[c - 'a']) {
        string left = longestNiceSubstring(s.substr(0, i));
        string right = longestNiceSubstring(s.substr(i + 1));
        return left.length() >= right.length() ? left : right;
    }
    if (isupper(c) && !lower[c - 'A']) {
        string left = longestNiceSubstring(s.substr(0, i));
        string right = longestNiceSubstring(s.substr(i + 1));
        return left.length() >= right.length() ? left : right;
    }
}

return s;
}
};

```

The screenshot shows a C++ IDE interface. On the left, a 'Problem List' table displays submission status: 'Accepted' (Mar 04, 2025, 3 ms, 11.7 MB), 'Wrong Answer' (Mar 04, 2025, N/A, N/A), and 'Accepted' (Mar 04, 2025, 368 ms, 124.9 MB). The main editor shows C++ code for a 'longestNiceSubstring' function. The code defines a 'Solution' class with a 'isNice' helper function and a 'longestNiceSubstring' method. The 'isNice' function checks if a string is a nice substring by verifying that for every character, its opposite case exists in the string. The 'longestNiceSubstring' method iterates through the string, using 'isNice' to find the longest valid substring. The 'Test Result' panel at the bottom shows 'Accepted' status with a runtime of 0 ms. It displays 'Case 1' with input 's = "kzakay"' and output 'aa'.

Description	Submissions	Solutions
Accepted Mar 04, 2025	C++	3 ms 11.7 MB
Wrong Answer Mar 04, 2025	C++	N/A N/A
Accepted Mar 04, 2025	C++	368 ms 124.9 MB

```

1 class Solution {
2 public:
3     bool isNice(string str) {
4         for (char c : str) {
5             if (islower(c) && str.find(toupper(c)) == string::npos) {
6                 return false;
7             }
8             if (isupper(c) && str.find(tolower(c)) == string::npos) {
9                 return false;
10            }
11        }
12        return true;
13    }
14
15 public:
16     string longestNiceSubstring(string s) {
17         int n = s.length();
18         if (n < 2) {
19             return "";
20         }
21         bool lower[26] = {false};
22         bool upper[26] = {false};
23     }
24 };

```

Test Result Runtime: 0 ms

Accepted

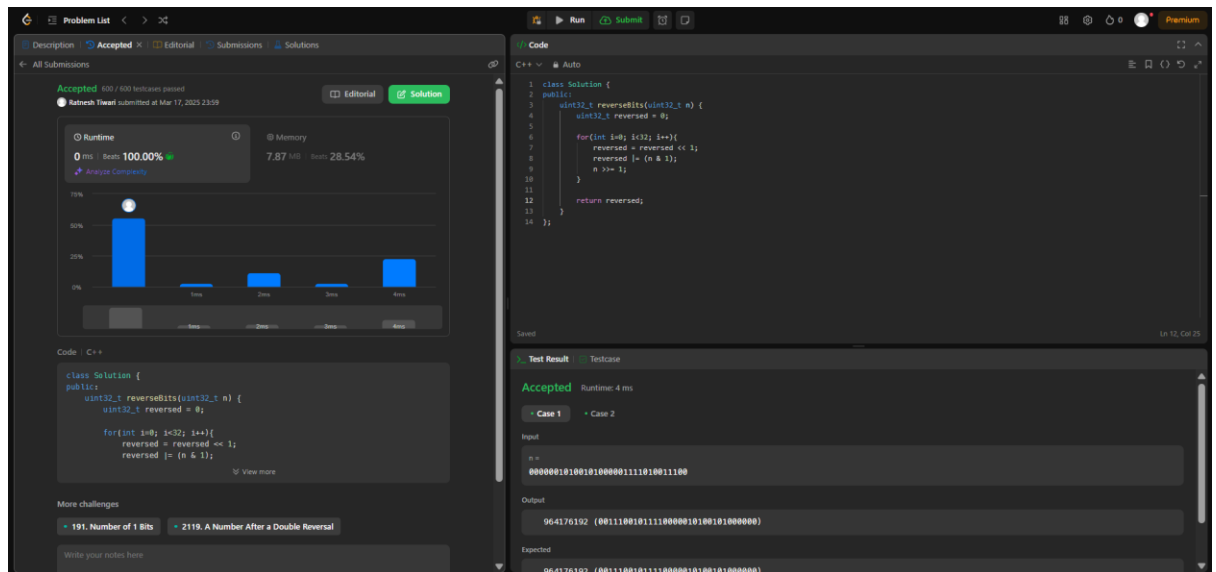
Case 1 Case 2 Case 3

Input
s =
"kzakay"

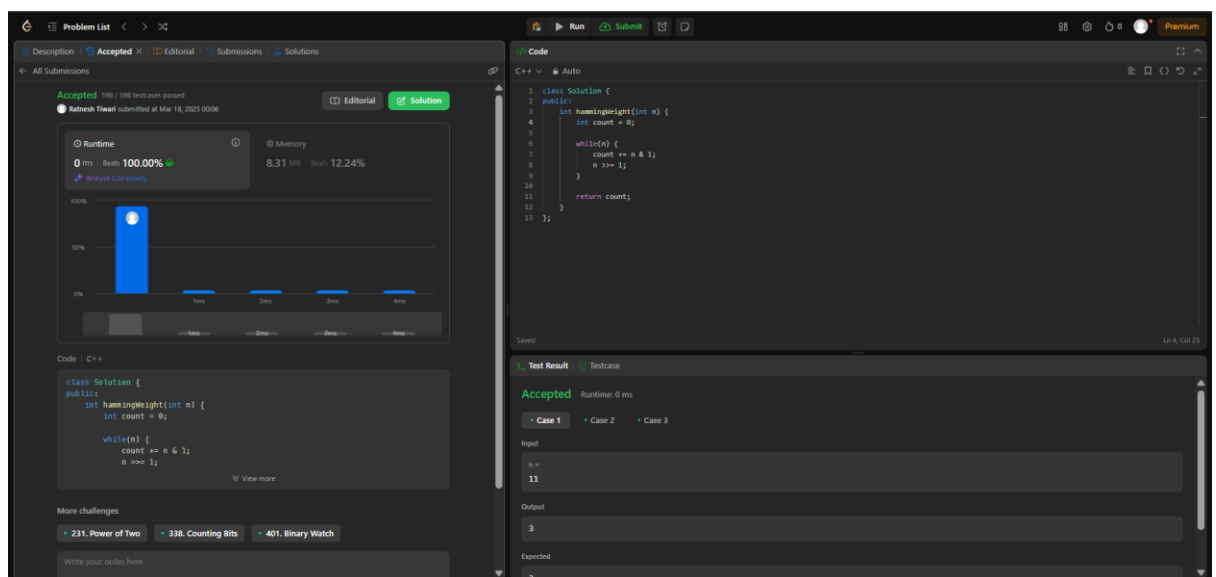
Output
"aa"

Expected

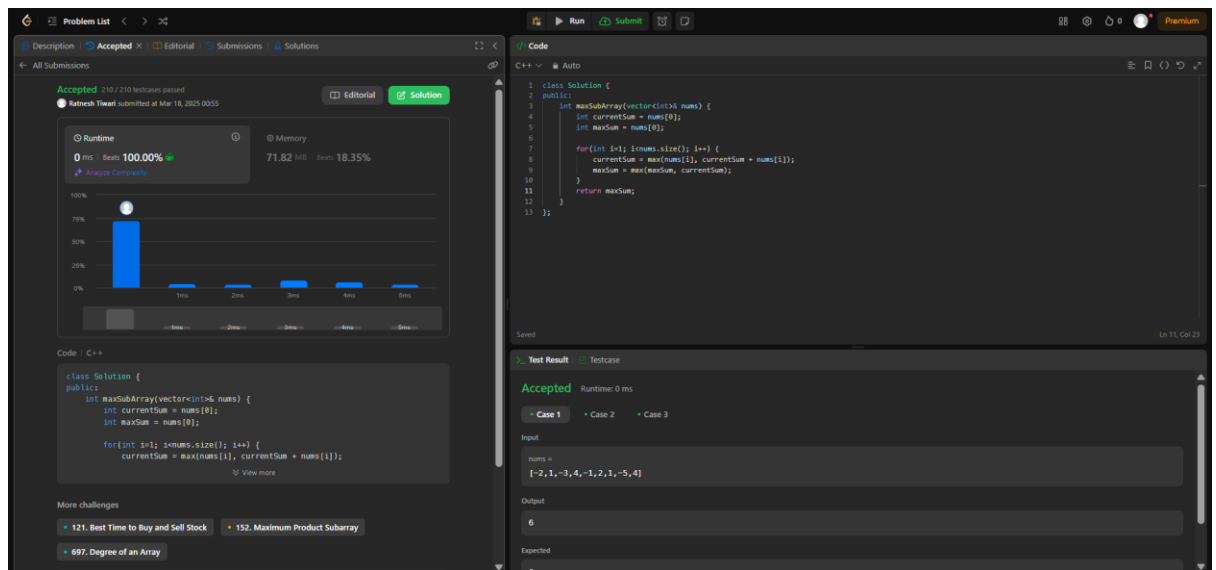
2. Reverse Bits: <https://leetcode.com/problems/reverse-bits/description/>



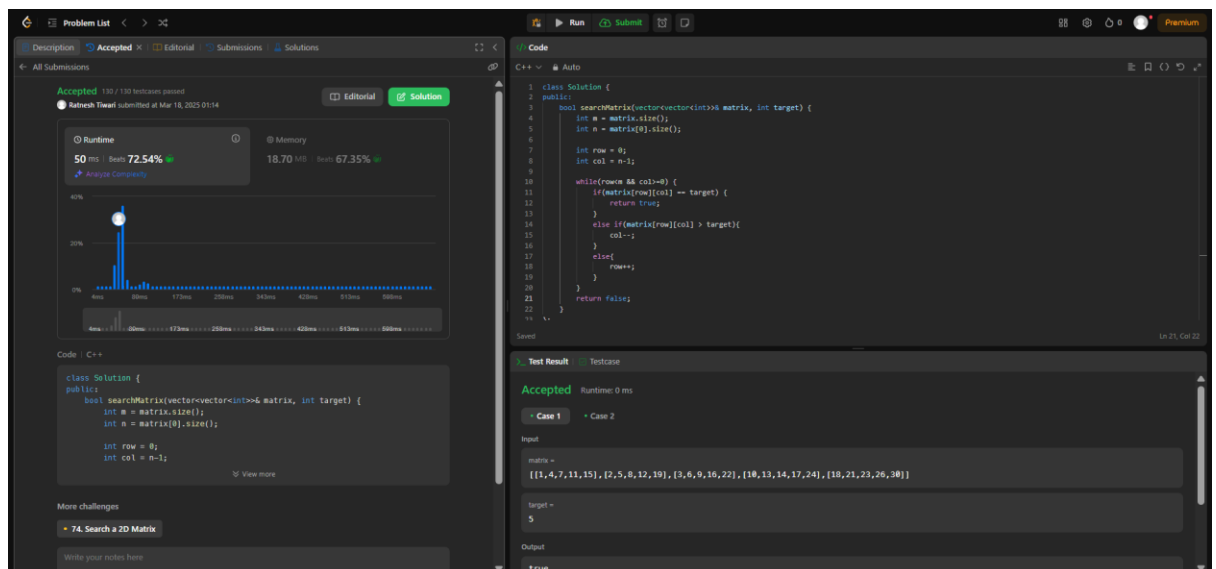
3. Number of 1 Bits: <https://leetcode.com/problems/number-of-1-bits/description/>



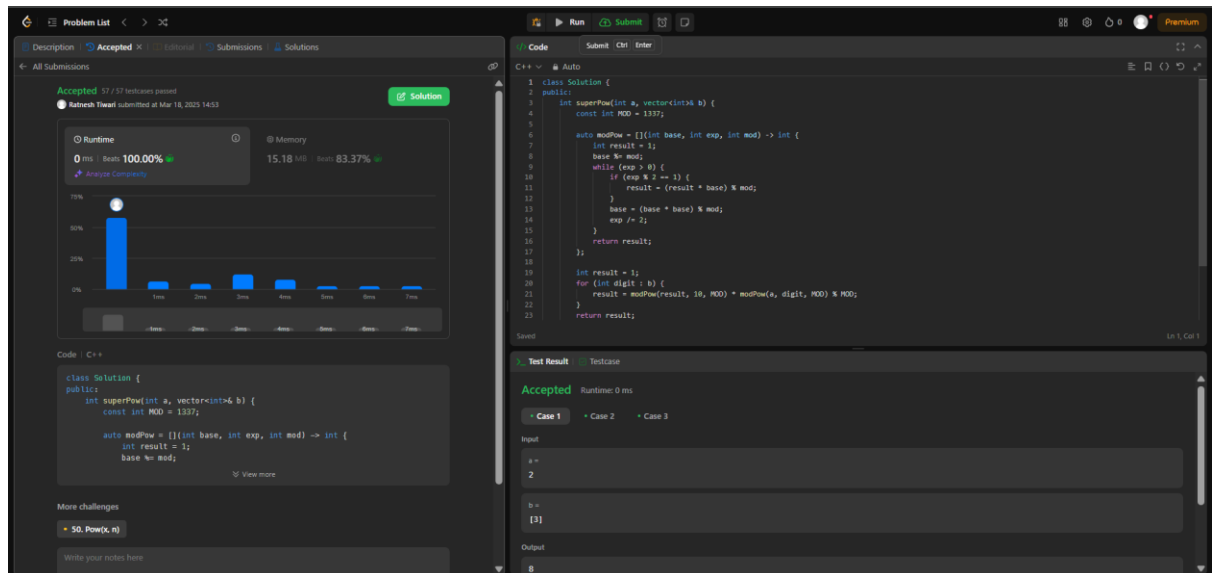
4. Maximum Subarray: <https://leetcode.com/problems/maximum-subarray/description/>



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



6. Super Pow: <https://leetcode.com/problems/super-pow/description/>



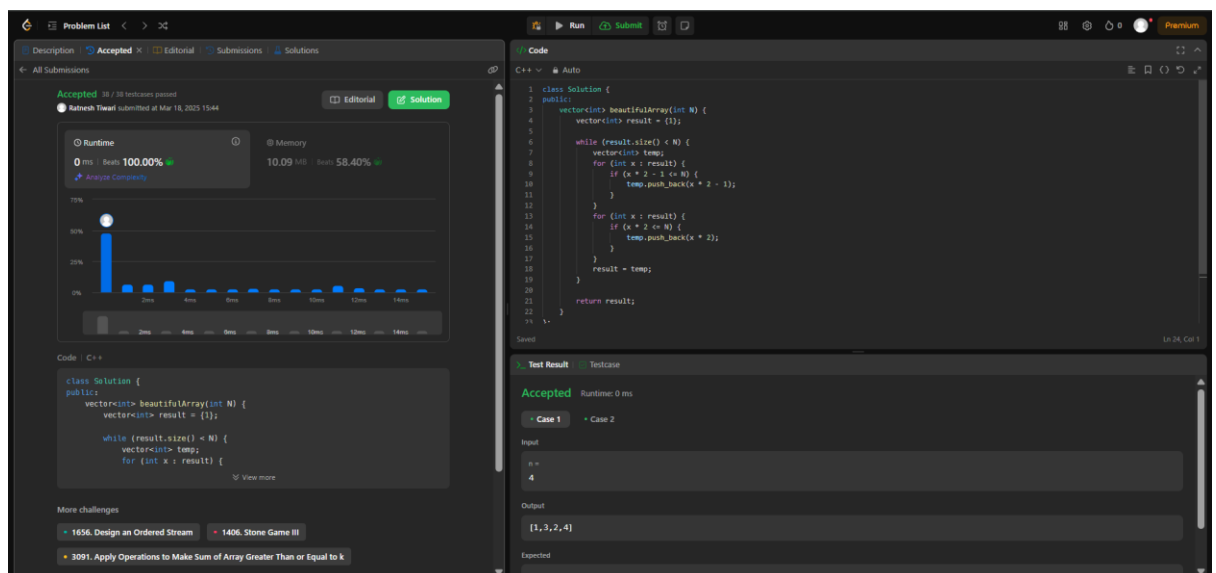
The screenshot shows the LeetCode interface for the 'Super Pow' problem. The solution is in C++ and is marked as 'Accepted'. The runtime is 0 ms, and the memory is 15.18 MB. The code implements a recursive function 'modPow' to calculate the power of a base modulo a given modulus. The test result shows the input 'a = 2' and 'b = [3]', resulting in an output of '8'.

```
class Solution {
public:
    int superPow(int a, vector<int>& b) {
        const int MOD = 1337;

        auto modPow = [&](int base, int exp, int mod) -> int {
            int result = 1;
            base %= mod;
            while (exp > 0) {
                if (exp & 1) {
                    result = (result * base) % mod;
                }
                base = (base * base) % mod;
                exp /= 2;
            }
            return result;
        };

        int result = 1;
        for (int digit : b) {
            result = modPow(result, 10, MOD) * modPow(a, digit, MOD) % MOD;
        }
        return result;
    }
};
```

7. Beautiful Array: <https://leetcode.com/problems/beautiful-array/description/>

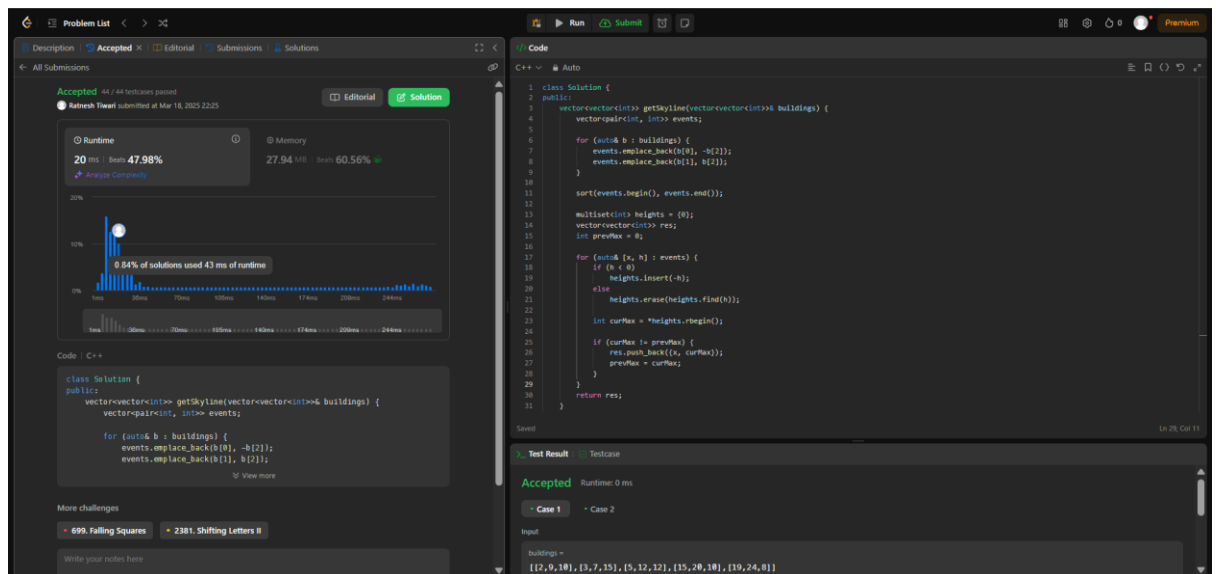


The screenshot shows the LeetCode interface for the 'Beautiful Array' problem. The solution is in C++ and is marked as 'Accepted'. The runtime is 0 ms, and the memory is 10.09 MB. The code implements a recursive function 'beautifulArray' to generate a beautiful array of size N. The test result shows the input 'n = 4', resulting in an output of '[1,2,2,4]'.

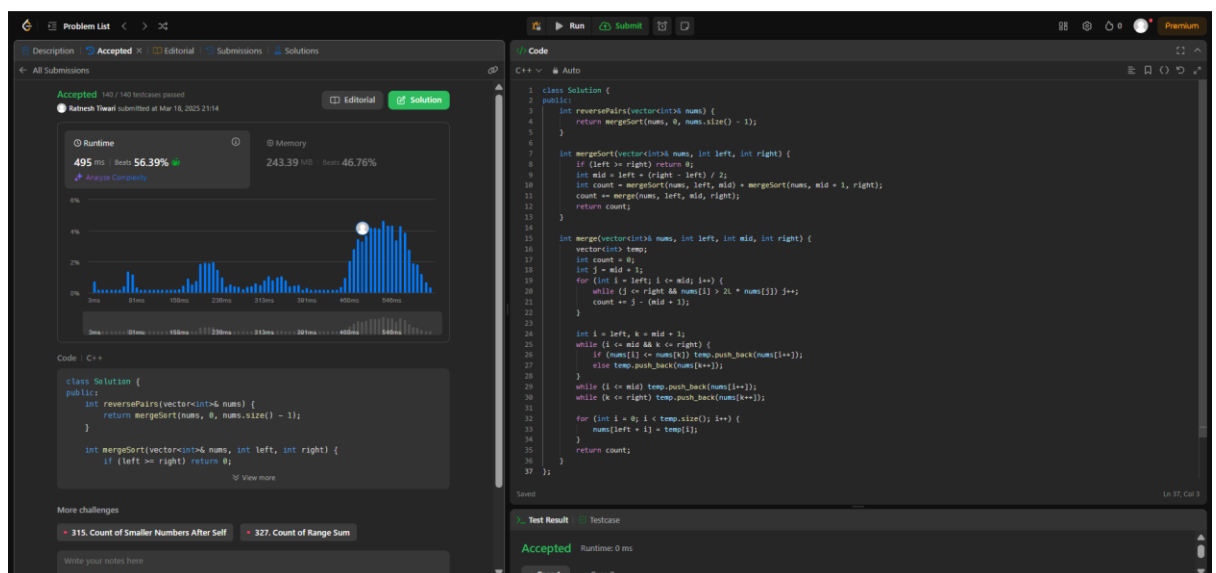
```
class Solution {
public:
    vector<int> beautifulArray(int N) {
        vector<int> result = {};

        while (result.size() < N) {
            vector<int> temp;
            for (int x : result) {
                if (x * 2 - 1 <= N) {
                    temp.push_back(x * 2 - 1);
                }
            }
            for (int x : result) {
                if (x * 2 <= N) {
                    temp.push_back(x * 2);
                }
            }
            result = temp;
        }
        return result;
    }
};
```

8. The Skyline Problem: <https://leetcode.com/problems/the-skyline-problem/description/>



9. Reverse Pairs: <https://leetcode.com/problems/reverse-pairs/description/>



10. Longest Increasing Subsequence II: <https://leetcode.com/problems/longest-increasing-subsequence-ii/description/>

Unable to solve. Still trying.