**1763. Longest Nice Substring**

class Solution {

public:

    bool isNice(string &s) {

        unordered\_set<char> st(s.begin(), s.end());

        for (char c : s) {

            if (!st.count(tolower(c)) || !st.count(toupper(c))) {

                return false;

            }

        }

        return true;

    }

    string longestNiceSubstring(string s) {

        int n = s.size();

        string ans = "";

        for (int i = 0; i < n; ++i) {

            for (int j = i; j < n; ++j) {

                string sub = s.substr(i, j - i + 1);

                if (isNice(sub) && sub.size() > ans.size()) {

                    ans = sub;

                }

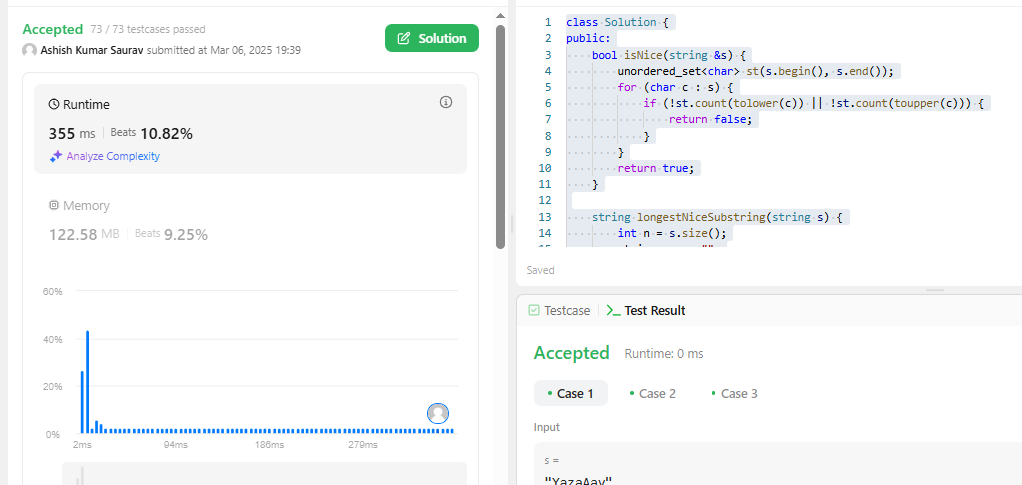
            }

        }

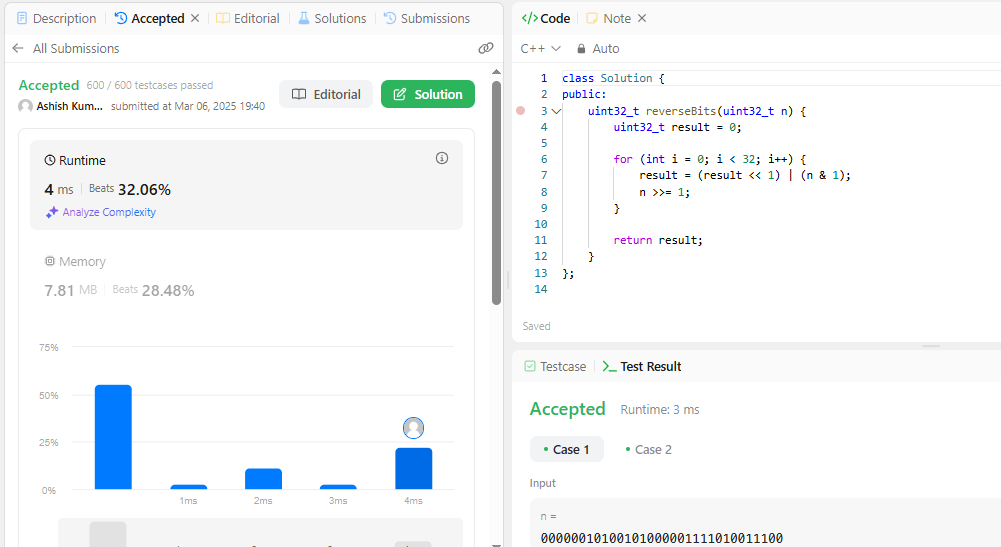
        return ans;

    }

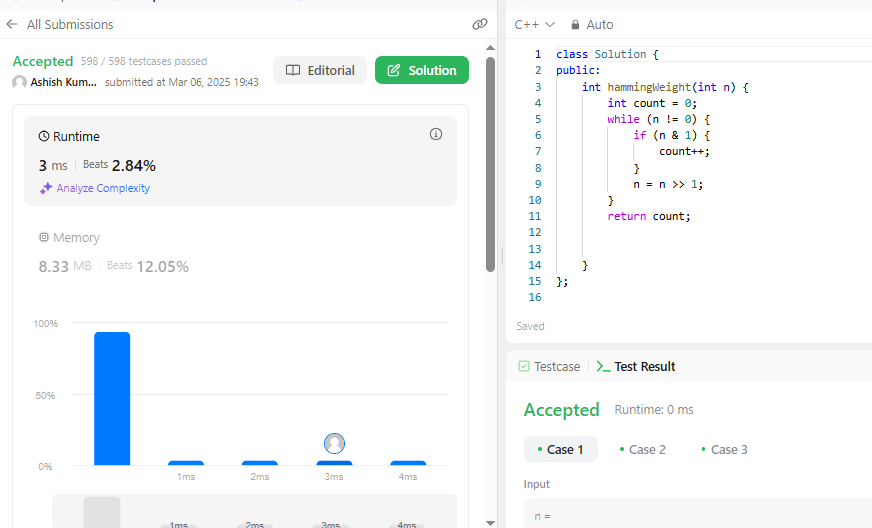
};



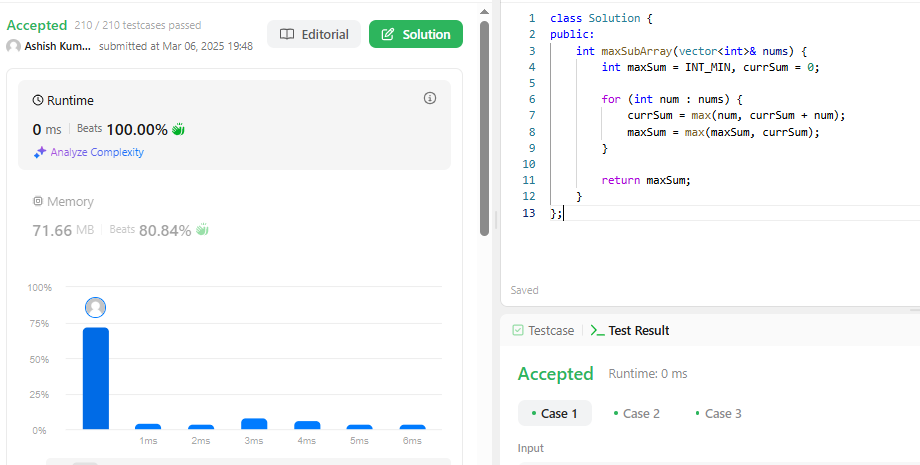
**190. Reverse Bits**



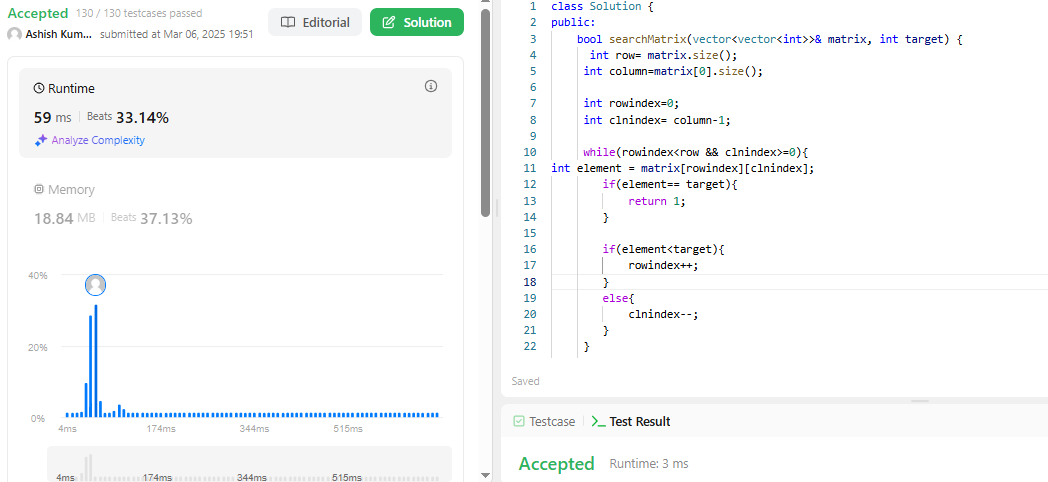
**191. Number of 1 Bits**



53. Maximum Subarray



**240. Search a 2D Matrix II**



**372. Super Pow**

class Solution {

public:

    const int MOD = 1337;

    int modPower(int a, int k) {

        int result = 1;

        a %= MOD;

        while (k > 0) {

            if (k % 2 == 1) result = (result \* a) % MOD;

            a = (a \* a) % MOD;

            k /= 2;

        }

        return result;

    }

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

        int result = 1;

        for (int digit : b) {

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

        }

        return result;

    }

};

