**Name:Pawan Singh**

**Uid:22BCS14871**

**SEC/GRP:FL\_IOT\_602/A**

Experiment – 4(AP)

1763.[Longest Nice Substring](https://leetcode.com/problems/longest-nice-substring/description/)

class Solution {

public:

    string longestNiceSubstring(string s) {

         if (s.size() < 2) return "";

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

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

             if (st.find((char) toupper(s[i])) == end(st) || st.find((char)tolower(s[i])) == end(st)) {

                string s1 = longestNiceSubstring(s.substr(0, i));

                string s2 = longestNiceSubstring(s.substr(i + 1));

                return s1.size() >= s2.size() ? s1 : s2;

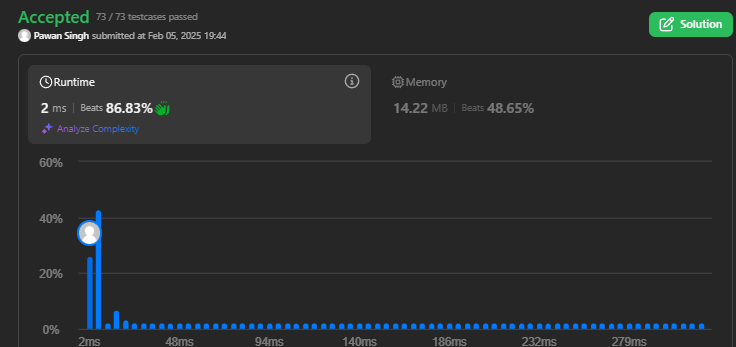
            }

        }

        return s;

    }

};



190.[Reverse Bits](https://leetcode.com/problems/reverse-bits/description/)

class Solution {

public:

    uint32\_t reverseBits(uint32\_t n) {

        uint32\_t result = 0;

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

            result = (result << 1) | (n & 1);

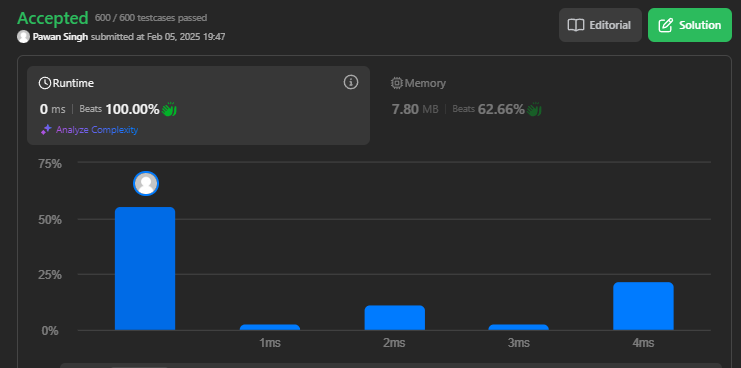
            n >>= 1;

        }

        return result;

    }

};



191.[Number of 1 Bits](https://leetcode.com/problems/number-of-1-bits/description/)

class Solution {

public:

    int hammingWeight(int n) {

         int res = 0;

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

            if ((n >> i) & 1) {

                res += 1;

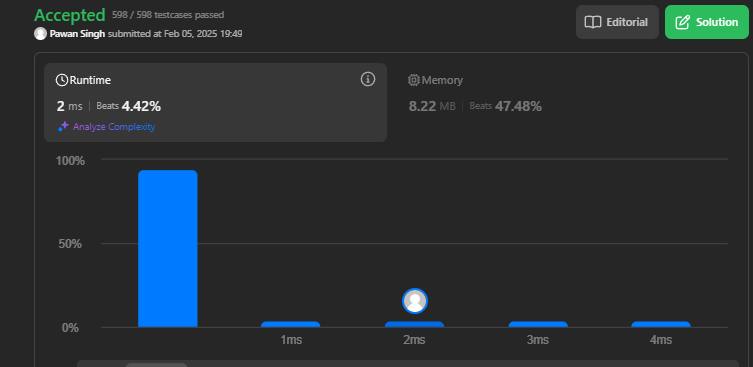
            }

        }

        return res;

    }

    };



53.[Maximum Subarray](https://leetcode.com/problems/maximum-subarray/description/)

class Solution {

public:

    int maxSubArray(vector<int>& nums) {

int maxi=INT\_MIN;

int sum=0;

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

{

    sum=sum+nums[i];

    maxi=max(sum,maxi);

    if(sum<0)

    {

        sum=0;

    }

}

return maxi;

    }

};



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

class Solution {

public:

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

        int n = matrix.size(), m = matrix[0].size();

        int row = 0, col = m - 1;

        while (row < n && col >= 0) {

            if (matrix[row][col] == target) return true;

            else if (matrix[row][col] < target) row++;

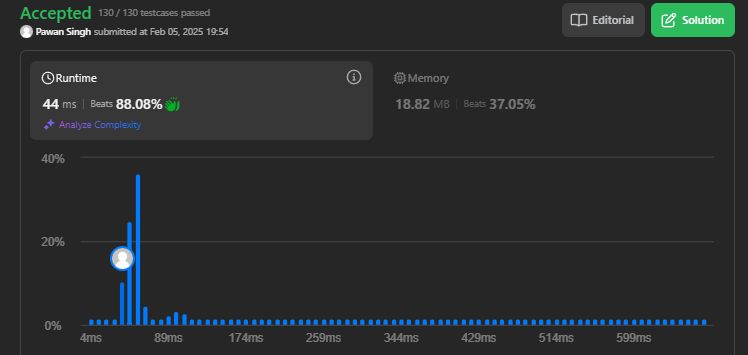
            else col--;

        }

        return false;

    }

};



372.[Super Pow](https://leetcode.com/problems/super-pow/description/)

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;

    }

};

