**ASSIGNMENT – 4**

**Name – Ritvik Jindal**

**UID – 22BCS10235**

**CLASS – FL\_IOT\_604-A**

**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> uset;

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

uset.insert(s[i]);

}

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

if(uset.count(tolower(s[i]))==true && uset.count(toupper(s[i]))==true) continue;

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

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

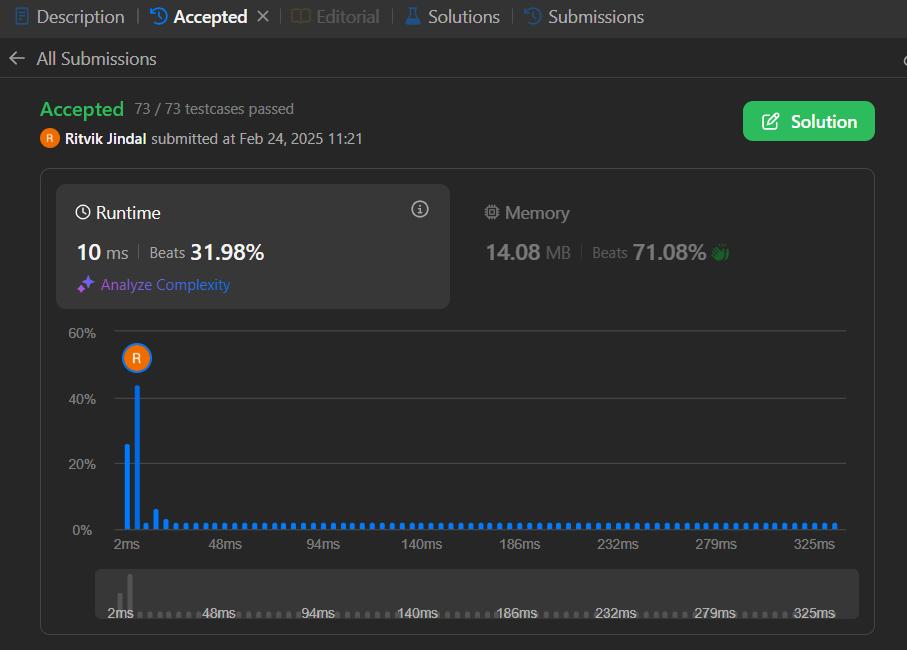
return (prev.size()>=next.size()? prev:next);

}

return s;

}

};



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

class Solution {

public:

uint32\_t reverseBits(uint32\_t n) {

uint32\_t a=0;

stack<int> stk;

int count =0;

while(stk.size()<32){

stk.push(n&1);

n=n>>1;

}

while(!stk.empty()){

if((stk.top()&1)==1){

a += pow(2,count);

}

stk.pop();

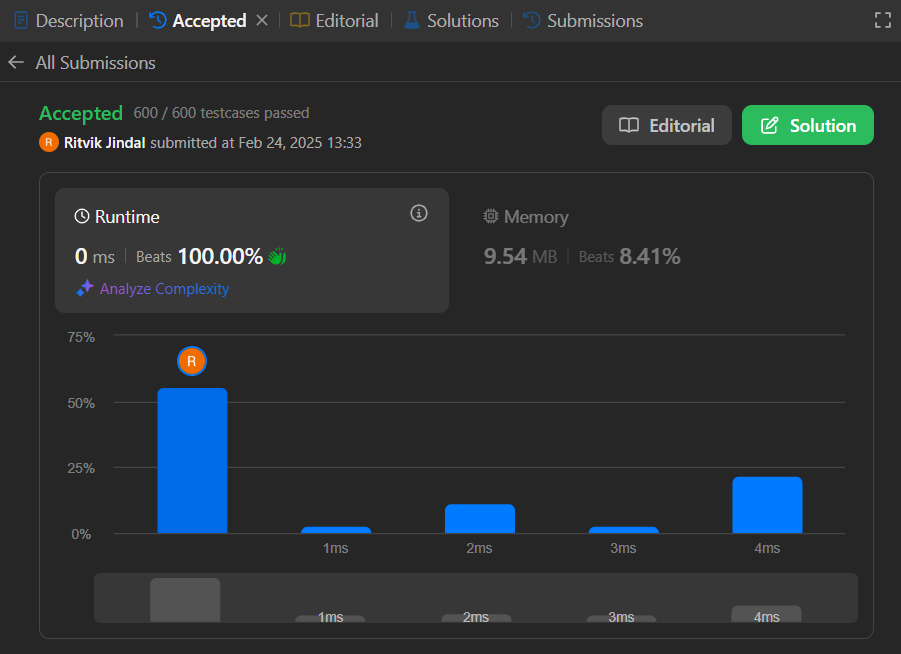
count++;

}

return a;

}

};



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

class Solution {

public:

int hammingWeight(int n) {

int res=0;

while(n>0){

if(n & 1){

res++;

}

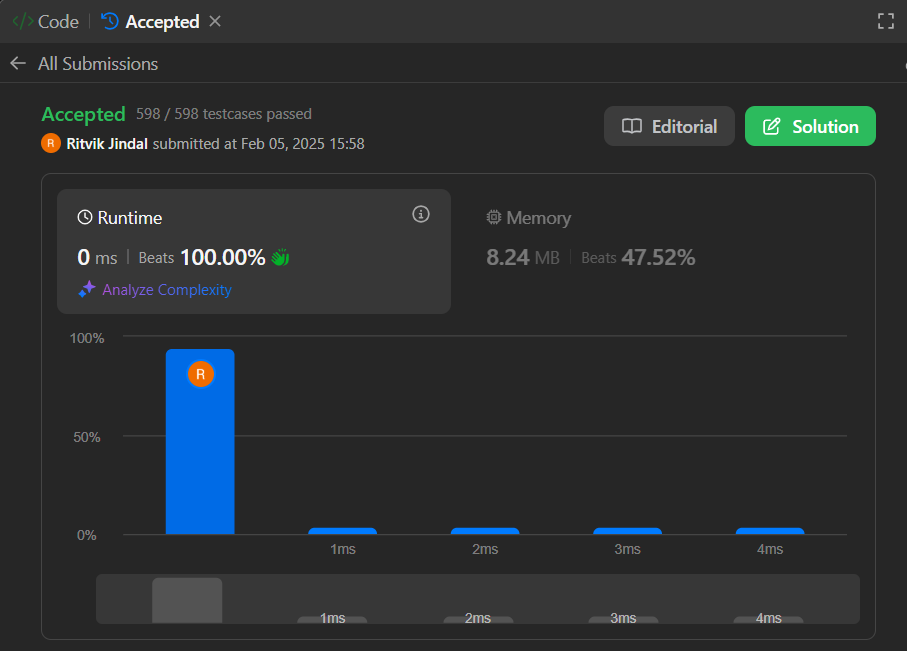
n = n>>1;

}

return res;

}

};



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

class Solution {

public:

int maxSubArray(vector<int>& nums) {

int res=nums[0],maxi = nums[0];

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

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

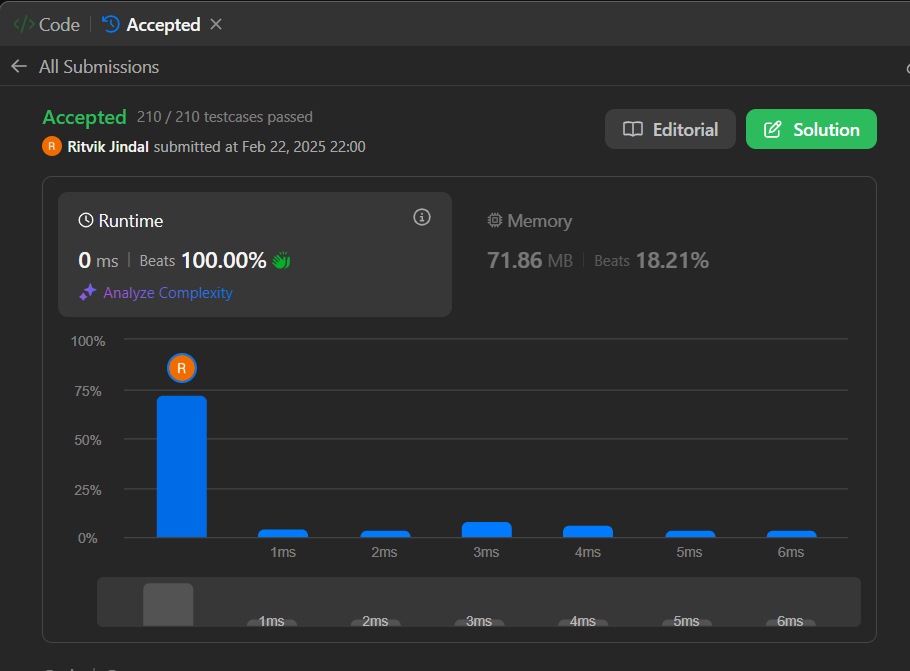
res = max(res,maxi);

}

return res;

}

};



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

class Solution {

public:

bool binarySearch(vector<int> &arr, int low, int high, int x)

{

while (low <= high) {

int mid = low + (high - low) / 2;

if (arr[mid] == x)

return true;

if (arr[mid] < x)

low = mid + 1;

else

high = mid - 1;

}

return false;

}

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

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

if(target>=matrix[i][0] && target <= matrix[i].back()){

int low =0;

int high = matrix[0].size() -1;

if(binarySearch(matrix[i],low,high,target))

return true;

}

else{

return false;

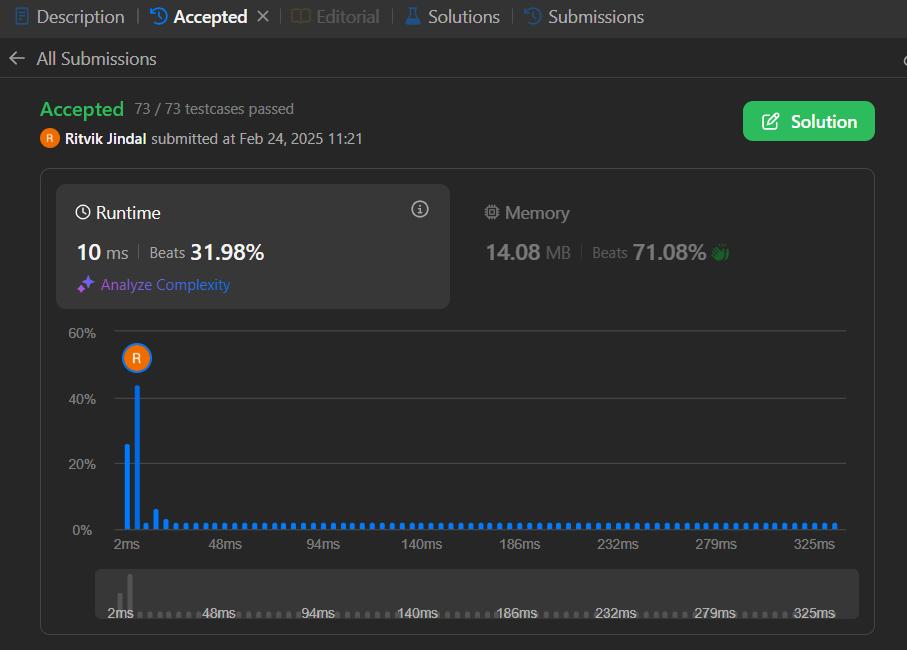
}

}

return false;

}

};



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

class Solution {

public:

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

long long pow = b[0];

int n=b.size();

if(n<1 || a==1){

return 1;

}

for(int i=1;i<n;i++){

pow \*= 10;

pow += b[i];

pow %= 1140;

}

if(pow ==0){

pow = 1140;

}

return power(a,pow)%1337;

}

long long power(int &a,long long &pow){

if(pow==0){

return 1;

}

long long temp = pow/2;

long long hpow = power(a,temp);

if(pow%2==0){

return hpow\*hpow%1337;

}

else{

return hpow\*hpow\*a %1337;

}

}

};

