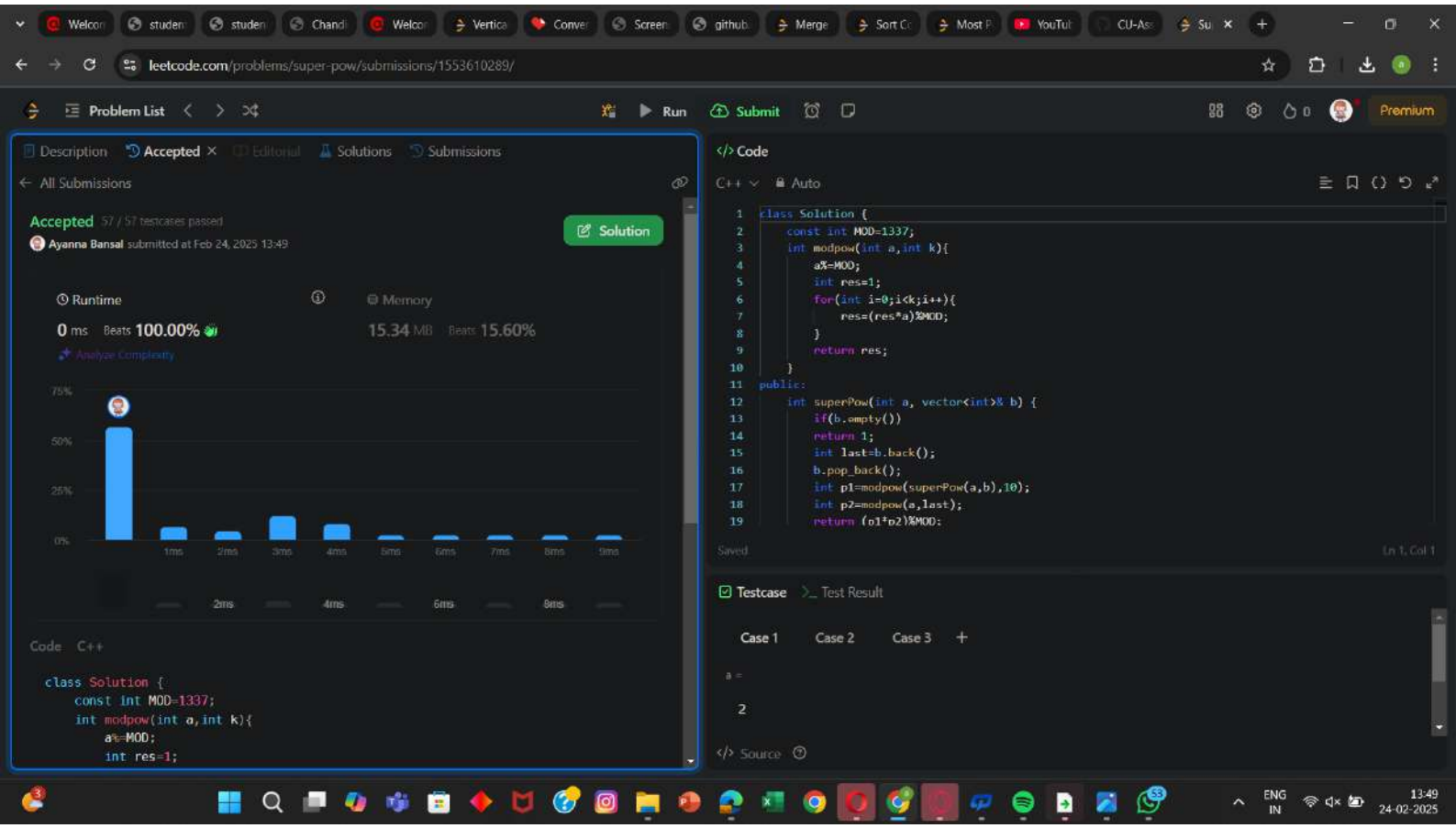
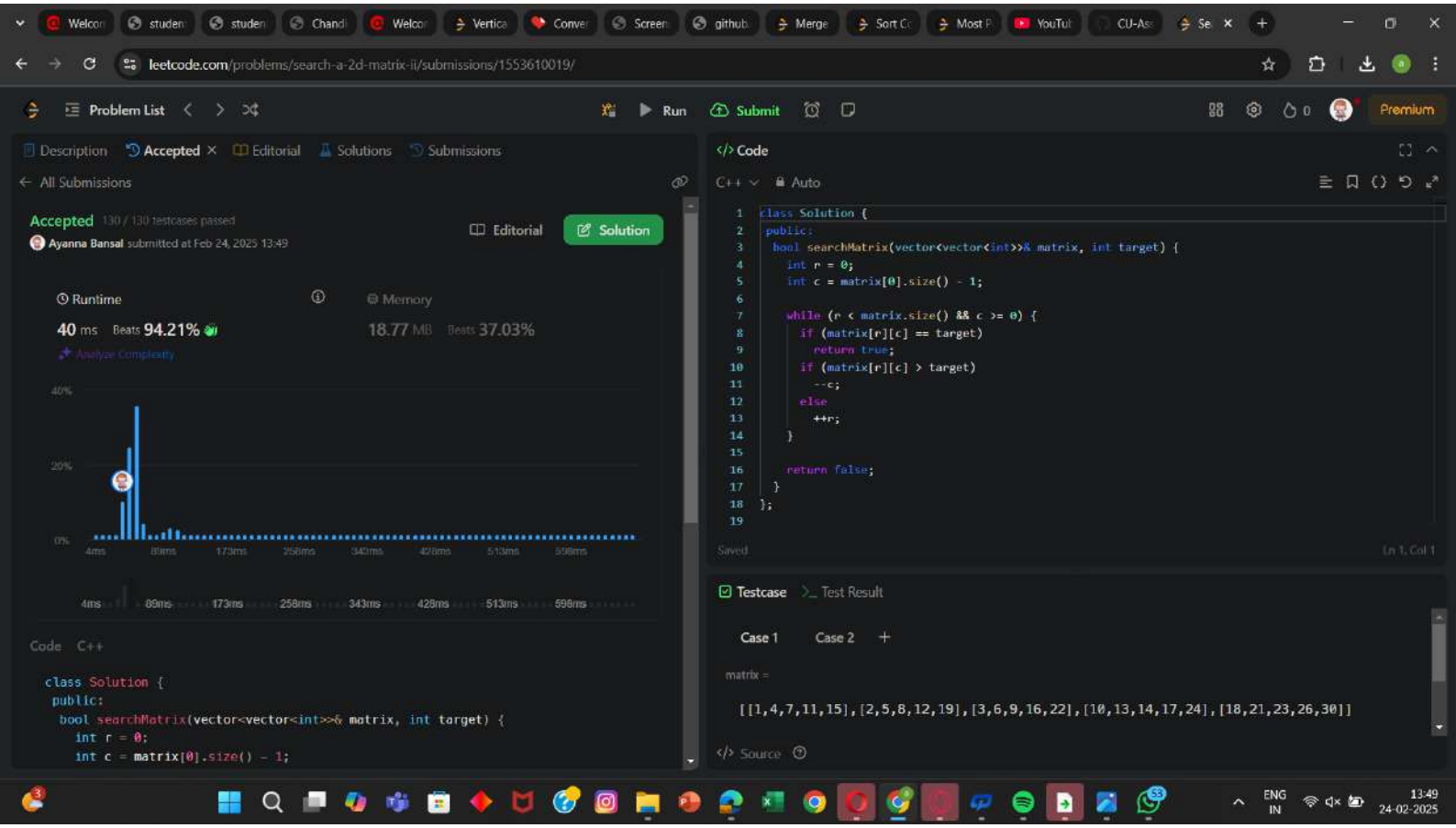
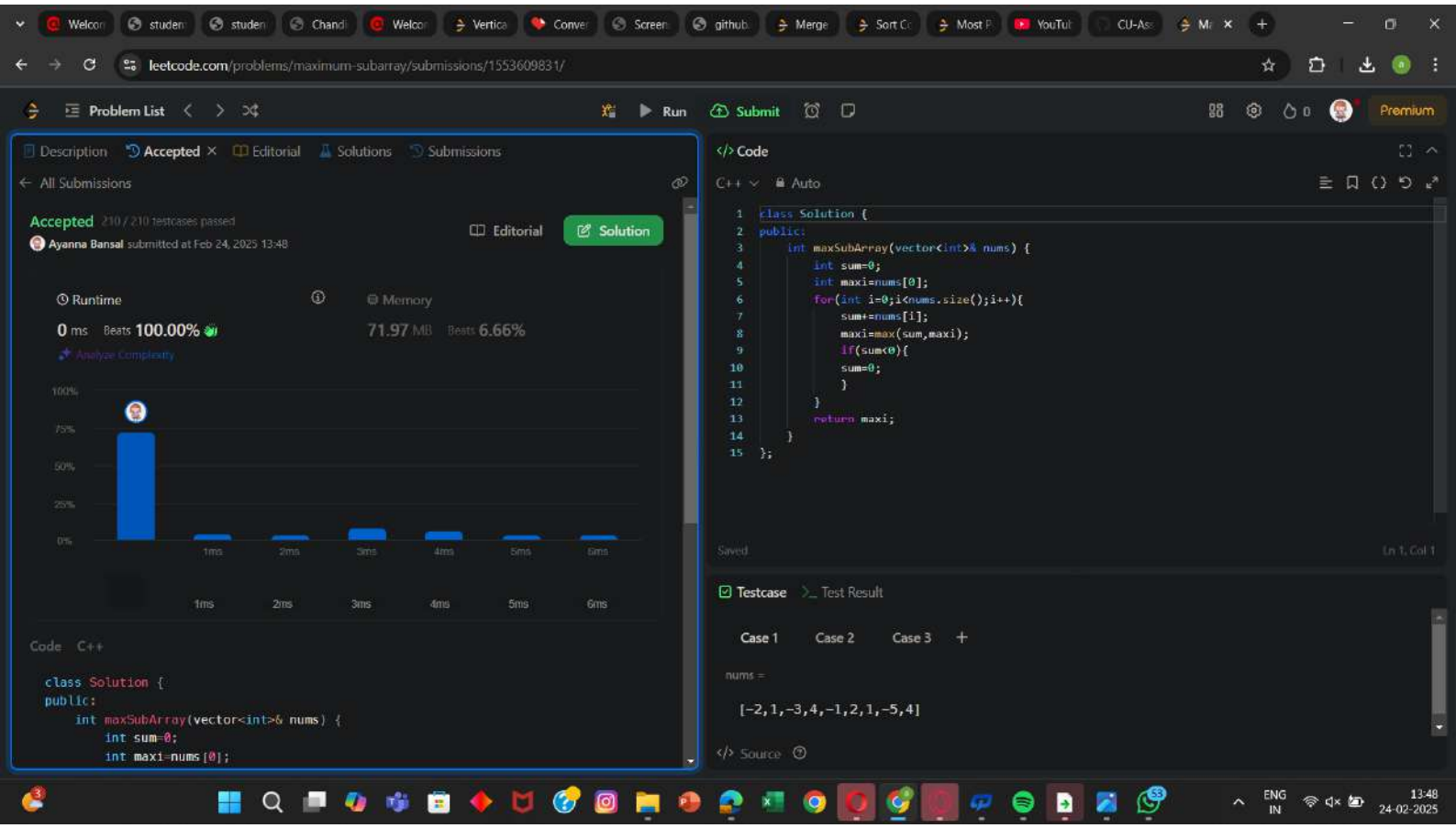
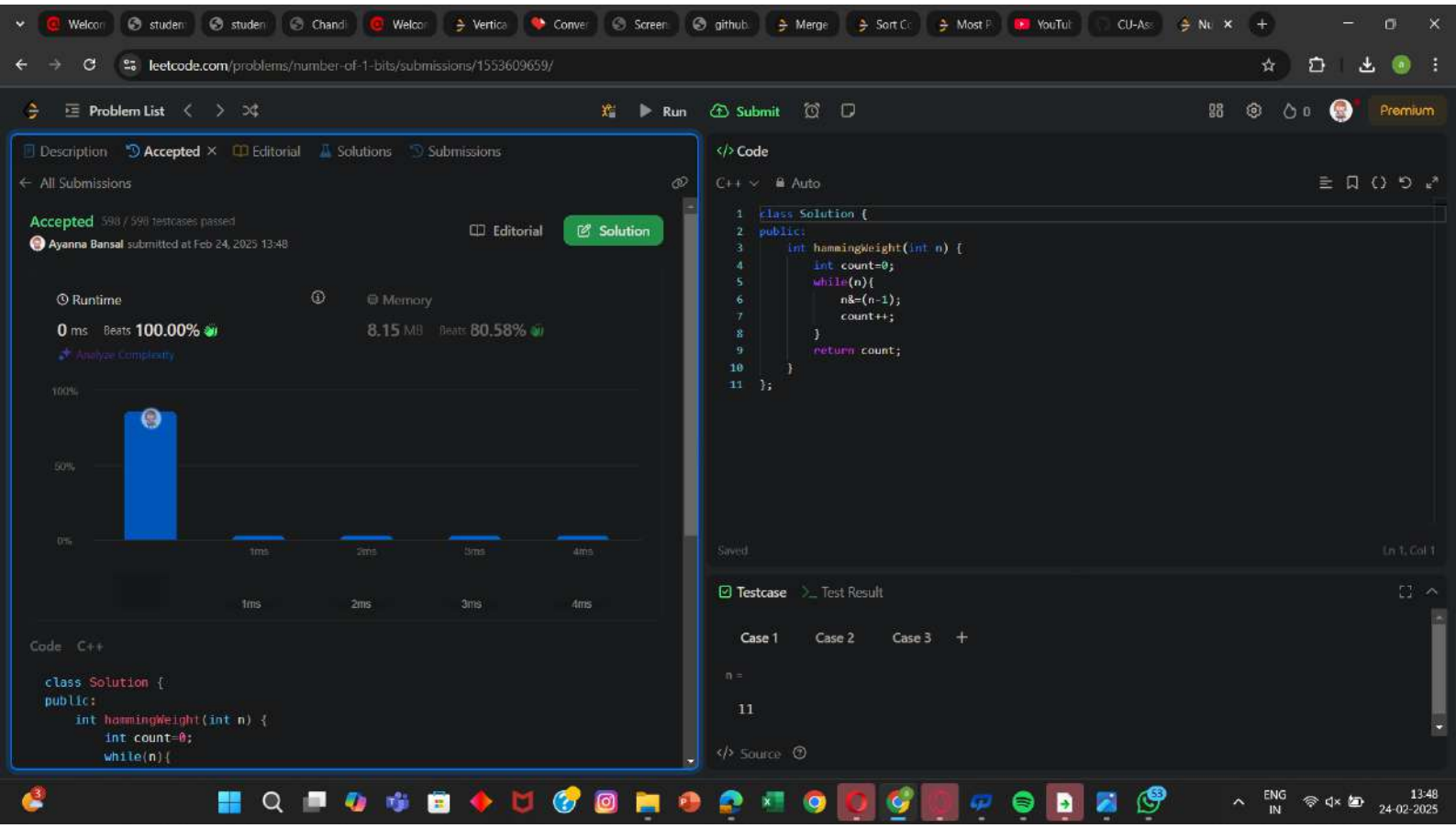
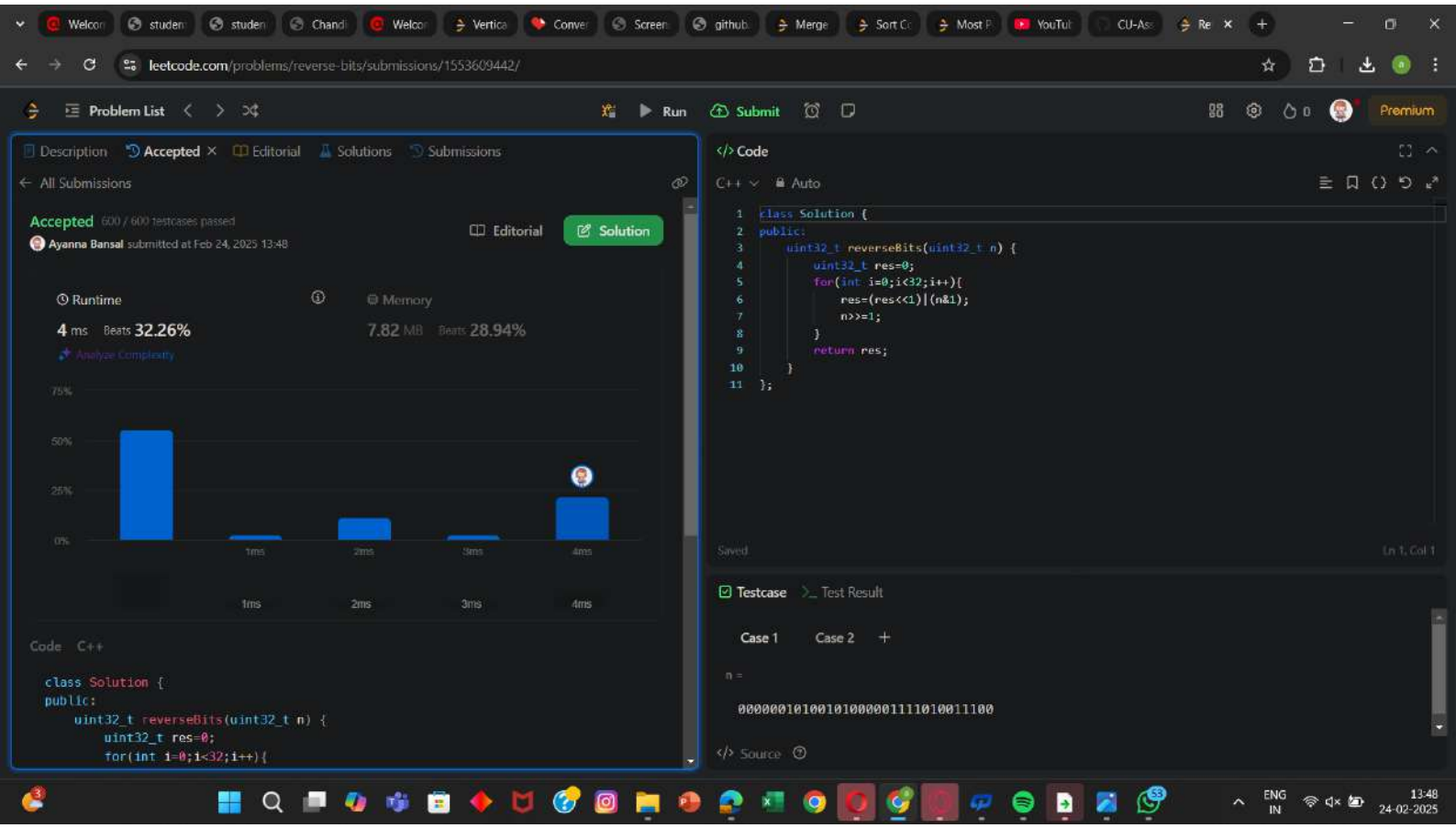
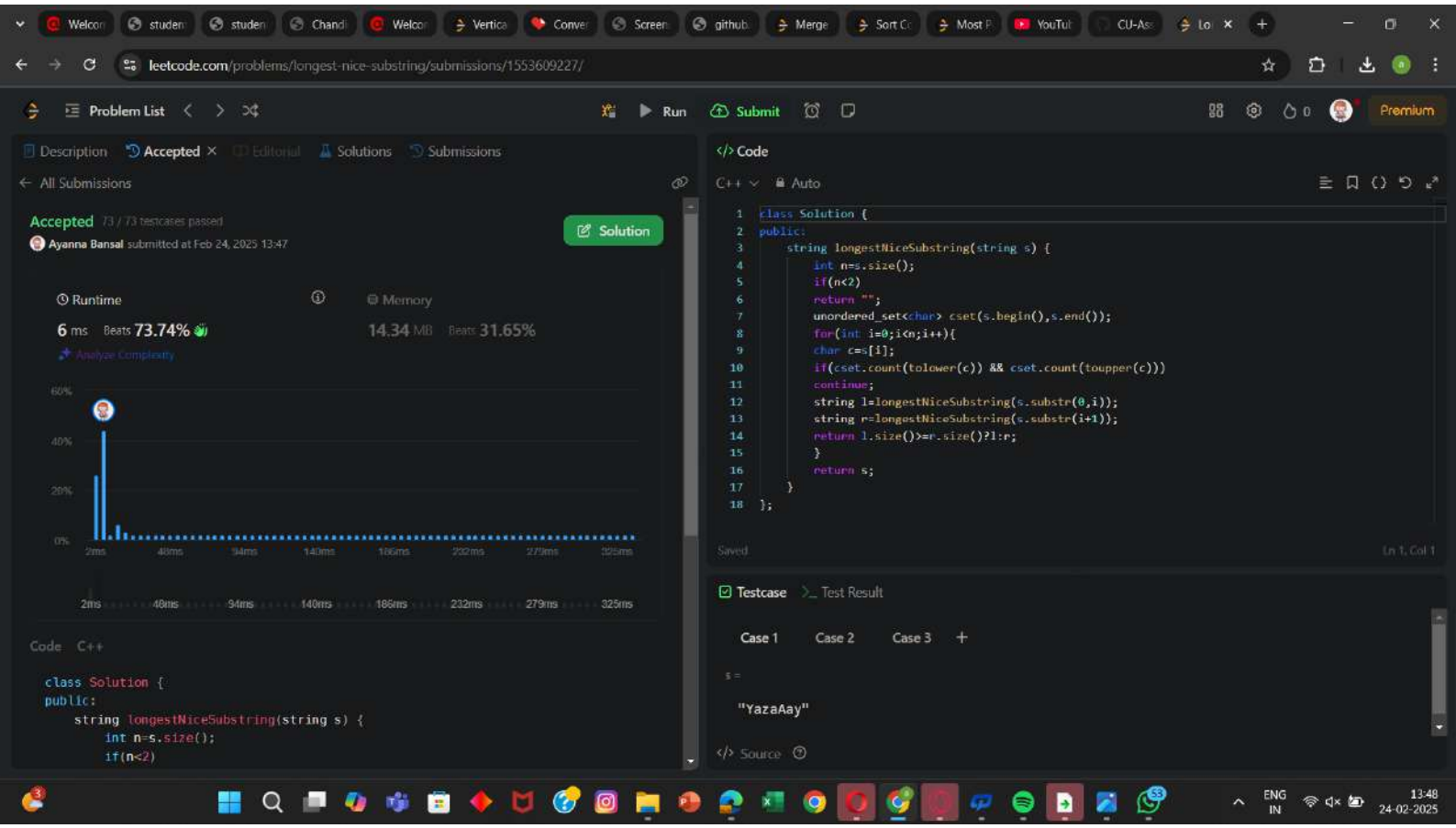

leetcode.com/problems/super-pow/submissions/1553610289/
Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 57 / 57 testcases passed
Ayanna Bansal submitted at Feb 24, 2025 13:49
Solution
Runtime
0 ms Beats 100.00%
Analyze Complexity
Memory
15.34 MB Beats 15.60%
75%
50%
25%
0%
1ms 2ms 3ms 4ms 5ms 6ms 7ms 8ms 9ms
2ms 4ms 6ms 8ms
Code C++
class Solution {
const int MOD=1337;
int modpow(int a,int k){
a%=MOD;
int res=1;
Code
C++
Auto
class Solution {
const int MOD=1337;
int modpow(int a,int k){
a%=MOD;
int res=1;
for(int i=0;i<k;i++){
res=(res*a)%MOD;
}
return res;
}
public:
int superPow(int a, vector<int>& b) {
if(b.empty())
return 1;
int last=b.back();
b.pop_back();
int p1=modpow(superPow(a,b),10);
int p2=modpow(a,last);
return (p1*p2)%MOD;
Saved
Ln 1, Col 1
Testcase
Test Result
Case 1
Case 2
Case 3
a =
2
Source
ENG
IN
13:49
24-02-2025


leetcode.com/problems/search-a-2d-matrix-ii/submissions/1553610019/
Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 130 / 130 testcases passed
Ayanna Bansal submitted at Feb 24, 2025 13:49
Editorial
Solution
Runtime
40 ms Beats 94.21%
Memory
18.77 MB Beats 37.03%
Analyze Complexity
40%
20%
0%
4ms
89ms
173ms
258ms
343ms
428ms
513ms
598ms
Code C++
class Solution {
public:
bool searchMatrix(vector<vector<int>>& matrix, int target) {
int r = 0;
int c = matrix[0].size() - 1;
Code
C++
Auto
1 class Solution {
2 public:
3 bool searchMatrix(vector<vector<int>>& matrix, int target) {
4 int r = 0;
5 int c = matrix[0].size() - 1;
6
7 while (r < matrix.size() && c >= 0) {
8 if (matrix[r][c] == target)
9 return true;
10 if (matrix[r][c] > target)
11 --c;
12 else
13 ++r;
14 }
15
16 return false;
17 }
18 };
19
Saved
Ln 1, Col 1
Testcase
Test Result
Case 1
Case 2
matrix =
[[1,4,7,11,15],[2,5,8,12,19],[3,6,9,16,22],[10,13,14,17,24],[18,21,23,26,30]]
Source
ENG
IN
13:49
24-02-2025


leetcode.com/problems/maximum-subarray/submissions/1553609831/
Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 210 / 210 testcases passed
Ayanna Bansal submitted at Feb 24, 2025 13:48
Editorial
Solution
Runtime
0 ms Beats 100.00%
Analyze Complexity
Memory
71.97 MB Beats 6.66%
100%
75%
50%
25%
0%
1ms
2ms
3ms
4ms
5ms
6ms
Code C++
class Solution {
public:
int maxSubArray(vector<int>& nums) {
int sum=0;
int maxi=nums[0];
Code
C++
Auto
class Solution {
public:
int maxSubArray(vector<int>& nums) {
int sum=0;
int maxi=nums[0];
for(int i=0;i<nums.size();i++){
sum+=nums[i];
maxi=max(sum,maxi);
if(sum<0){
sum=0;
}
}
return maxi;
}
};
Saved
Ln 1, Col 1
Testcase
Test Result
Case 1
Case 2
Case 3
nums =
[-2,1,-3,4,-1,2,1,-5,4]
Source


leetcode.com/problems/number-of-1-bits/submissions/1553609659/
Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 598 / 598 testcases passed
Ayanna Bansal submitted at Feb 24, 2025 13:48
Editorial
Solution
Runtime
0 ms Beats 100.00%
Memory
8.15 MB Beats 80.58%
Analyze Complexity
100%
50%
0%
1ms
2ms
3ms
4ms
Code C++
class Solution {
public:
int hammingWeight(int n) {
int count=0;
while(n){
Code
C++
Auto
class Solution {
public:
int hammingWeight(int n) {
int count=0;
while(n){
n&=(n-1);
count++;
}
return count;
}
};
Saved
Ln 1, Col 1
Testcase
Test Result
Case 1
Case 2
Case 3
n =
11
Source
ENG
IN
13:48
24-02-2025


leetcode.com/problems/reverse-bits/submissions/1553609442/
Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 600 / 600 testcases passed
Ayanna Bansal submitted at Feb 24, 2025 13:48
Editorial
Solution
Runtime
4 ms Beats 32.26%
Memory
7.82 MB Beats 28.94%
Analyze Complexity
75%
50%
25%
0%
1ms
2ms
3ms
4ms
Code C++
class Solution {
public:
uint32_t reverseBits(uint32_t n) {
uint32_t res=0;
for(int i=0;i<32;i++){
Code
C++
Auto
class Solution {
public:
uint32_t reverseBits(uint32_t n) {
uint32_t res=0;
for(int i=0;i<32;i++){
res=(res<<1)|(n&1);
n>>=1;
}
return res;
}
};
Saved
Ln 1, Col 1
Testcase
Test Result
Case 1
Case 2
n =
00000010100101000001111010011100
Source
ENG
IN
13:48
24-02-2025


leetcode.com/problems/longest-nice-substring/submissions/1553609227/
Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 73 / 73 testcases passed
Ayanna Bansal submitted at Feb 24, 2025 13:47
Solution
Runtime
6 ms Beats 73.74%
Memory
14.34 MB Beats 31.65%
Analyze Complexity
60%
40%
20%
0%
2ms
48ms
94ms
140ms
186ms
232ms
279ms
325ms
Code C++
class Solution {
public:
string longestNiceSubstring(string s) {
int n=s.size();
if(n<2)
Code
C++
Auto
class Solution {
public:
string longestNiceSubstring(string s) {
int n=s.size();
if(n<2)
return "";
unordered_set<char> cset(s.begin(),s.end());
for(int i=0;i<n;i++){
char c=s[i];
if(cset.count(tolower(c)) && cset.count(toupper(c)))
continue;
string l=longestNiceSubstring(s.substr(0,i));
string r=longestNiceSubstring(s.substr(i+1));
return l.size()>=r.size()?l:r;
}
return s;
}
};
Saved
Ln 1, Col 1
Testcase
Test Result
Case 1
Case 2
Case 3
s =
"YazaAay"
Source
ENG
IN
13:48
24-02-2025