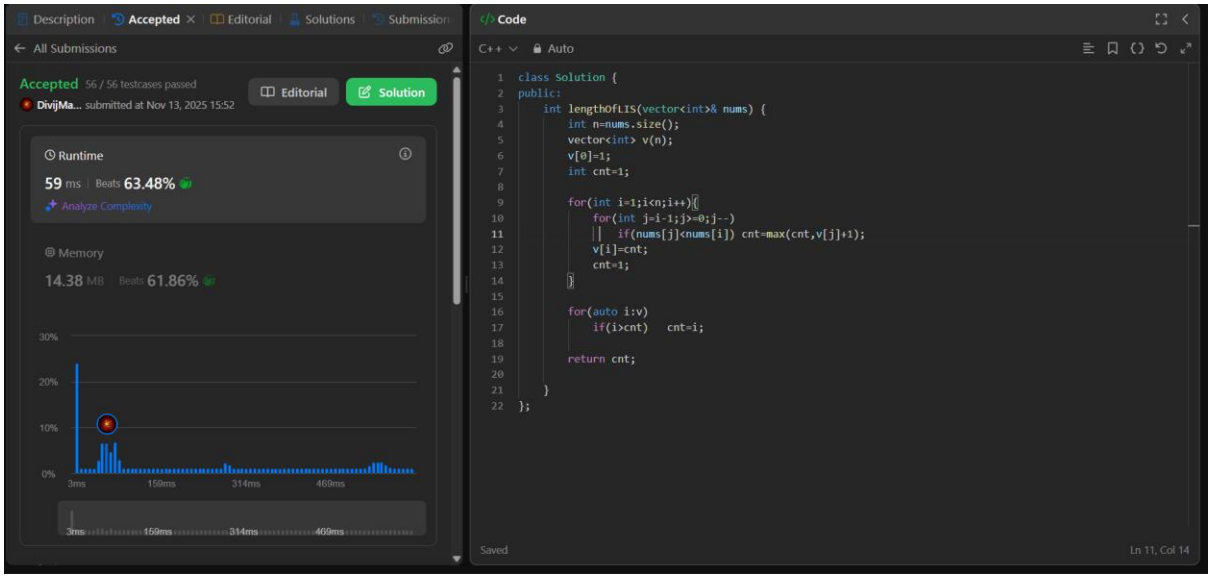
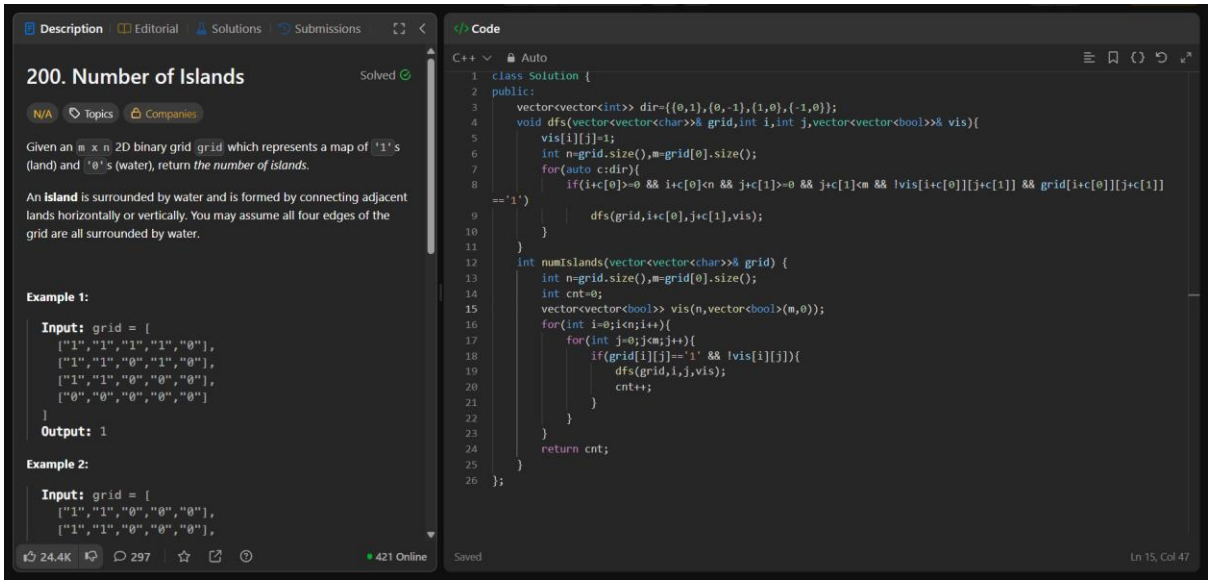

Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 56 / 56 testcases passed
DivijMa... submitted at Nov 13, 2025 15:52
Editorial
Solution
Runtime
59 ms | Beats 63.48%
Analyze Complexity
Memory
14.38 MB | Beats 61.86%
30%
20%
10%
0%
3ms
159ms
314ms
469ms
Code
C++
Auto
class Solution {
public:
int lengthOfLIS(vector<int>& nums) {
int n=nums.size();
vector<int> v(n);
v[0]=1;
int cnt=1;
for(int i=1;i<n;i++){
for(int j=i-1;j>=0;j--)
if(nums[j]<nums[i]) cnt=max(cnt,v[j]+1);
v[i]=cnt;
cnt=1;
}
for(auto i:v)
if(i>cnt) cnt=i;
return cnt;
}
};
Saved
Ln 11, Col 14


Description
Editorial
Solutions
Submissions
200. Number of Islands
Solved
N/A
Topics
Companies
Given an m x n 2D binary grid grid which represents a map of '1's (land) and '0's (water), return the number of islands.
An island is surrounded by water and is formed by connecting adjacent lands horizontally or vertically. You may assume all four edges of the grid are all surrounded by water.
Example 1:
Input: grid = [
["1","1","1","1","0"],
["1","1","0","1","0"],
["1","1","0","0","0"],
["0","0","0","0","0"]
]
Output: 1
Example 2:
Input: grid = [
["1","1","0","0","0"],
["1","1","0","0","0"],
24.4K
297
421 Online
Code
C++
Auto
class Solution {
public:
vector<vector<int>> dir={{0,1},{0,-1},{1,0},{-1,0}};
void dfs(vector<vector<char>>& grid,int i,int j,vector<vector<bool>>& vis){
vis[i][j]=1;
int n=grid.size(),m=grid[0].size();
for(auto c:dir){
if(i+c[0]>=0 && i+c[0]<n && j+c[1]>=0 && j+c[1]<m && !vis[i+c[0]][j+c[1]] && grid[i+c[0]][j+c[1]]=='1')
dfs(grid,i+c[0],j+c[1],vis);
}
}
int numIslands(vector<vector<char>>& grid) {
int n=grid.size(),m=grid[0].size();
int cnt=0;
vector<vector<bool>> vis(n,vector<bool>(m,0));
for(int i=0;i<n;i++){
for(int j=0;j<m;j++){
if(grid[i][j]=='1' && !vis[i][j]){
dfs(grid,i,j,vis);
cnt++;
}
}
}
return cnt;
}
};
Saved
Ln 15, Col 47