Name: Anshul

UID: 22BCS16477

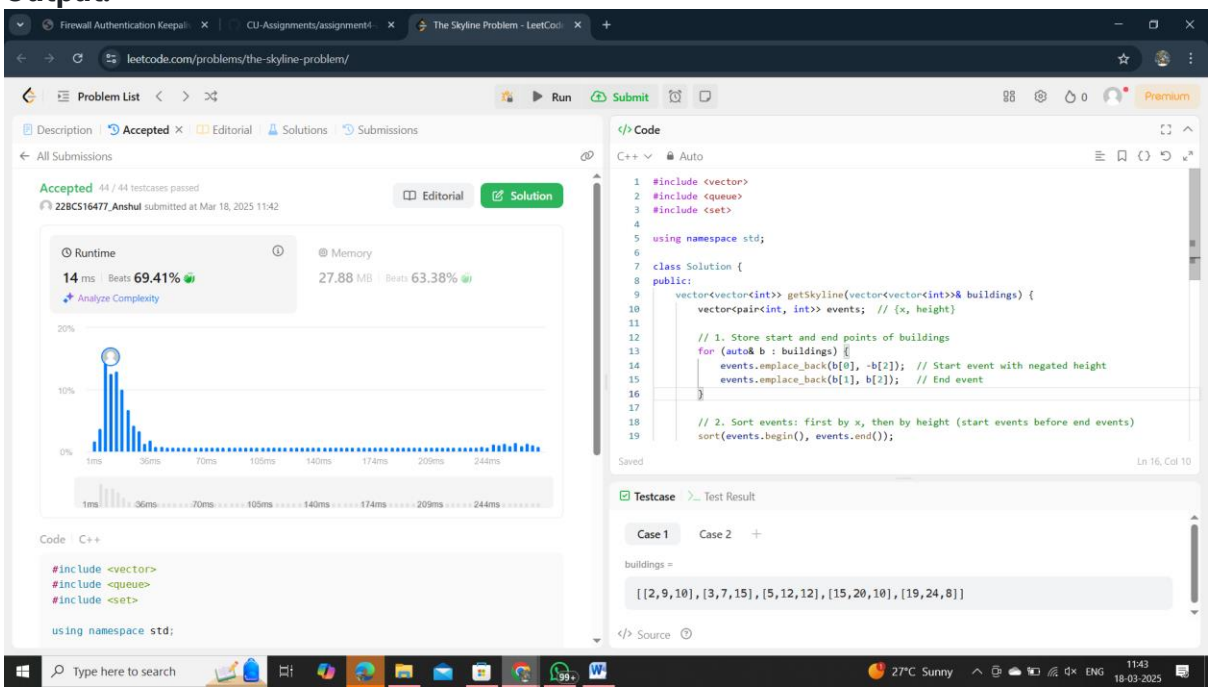
Section/Group: 609(B)

The Skyline Problem

Code:

```
#include <vector>
#include <queue>
#include <set>

using namespace std;
class Solution {
public:
    vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
        vector<pair<int, int>> events;  // {x, height}

        // 1. Store start and end points of buildings
        for (auto& b : buildings) {
            events.emplace_back(b[0], -b[2]);  // Start event with negated height
            events.emplace_back(b[1], b[2]);   // End event
        }

        // 2. Sort events: first by x, then by height (start events before end
events)
        sort(events.begin(), events.end());

        multiset<int> heights = {0};  // Using multiset to store active heights
        int prevHeight = 0;
        vector<vector<int>> result;

        // 3. Process all events
        for (auto& [x, h] : events) {
            if (h < 0)
                heights.insert(-h);   // Add building height for start events
            else
                heights.erase(heights.find(h));  // Remove height for end events

            int currentHeight = *heights.rbegin();  // Get max height
            if (currentHeight != prevHeight) {
                result.push_back({x, currentHeight});
                prevHeight = currentHeight;
            }
        }

        return result;
    }
};
```

Output:


Firewall Authentication Keepali
CU-Assignments/assignment4-
The Skyline Problem - LeetCod
leetcode.com/problems/the-skyline-problem/
Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 44 / 44 testcases passed
22BCS16477_Anshul submitted at Mar 18, 2025 11:42
Editorial
Solution
Runtime
14 ms | Beats 69.41%
Analyze Complexity
Memory
27.88 MB | Beats 63.38%
Code
C++
Auto
#include <vector>
#include <queue>
#include <set>
using namespace std;
class Solution {
public:
vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
vector<pair<int, int>> events; // {x, height}
// 1. Store start and end points of buildings
for (auto& b : buildings) {
events.emplace_back(b[0], -b[2]); // Start event with negated height
events.emplace_back(b[1], b[2]); // End event
}
// 2. Sort events: first by x, then by height (start events before end events)
sort(events.begin(), events.end());
Saved
Ln 16, Col 10
Testcase
Test Result
Case 1
Case 2
buildings =
[[2,9,10],[3,7,15],[5,12,12],[15,20,10],[19,24,8]]
Source
Type here to search
27°C Sunny
ENG
11:43
18-03-2025