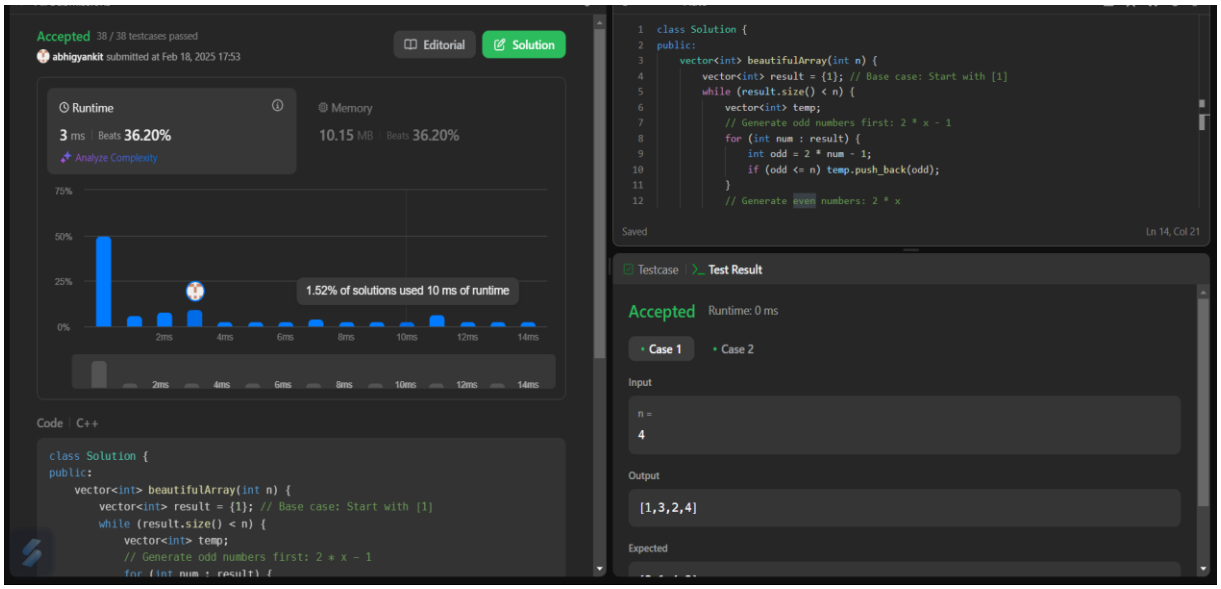
## Assignment 4

| Name: Agrini Mishra | Uid: 22BCS10821 |
|---|---|
| Branch: BE_CSE | Semester: 6th |
| Section: IOT_637-B | Subject: AP Lab II |

### 932. Beautiful Array

```
class Solution {
public:
    vector<int> beautifulArray(int n) {
        vector<int> result = {1}; // Base case: Start with [1]
        while (result.size() < n) {
            vector<int> temp;
            // Generate odd numbers first: 2 * x - 1
            for (int num : result) {
                int odd = 2 * num - 1;
                if (odd <= n) temp.push_back(odd);
            }
            // Generate even numbers: 2 * x
            for (int num : result) {
                int even = 2 * num;
                if (even <= n) temp.push_back(even);
            }
            result = temp; // Update the result
        }
        return result;
    }
};
```




## 218. The Skyline Problem

```
class Solution {
public:
    vector<vector<int>> getSkyline(vector<vector<int>>& buildings)
{
        vector<pair<int, int>> events;
        for (const auto& b : buildings) {
            events.emplace_back(b[0], -b[2]); // Start of building
            events.emplace_back(b[1], b[2]);  // End of building
        }
        sort(events.begin(), events.end());
        multiset<int> heights = {0};
        vector<vector<int>> skyline;
        int prevMax = 0;
        // Process each event
        for (const auto& [x, h] : events) {
            if (h < 0) {
                heights.insert(-h); // Add building height
            } else {
                heights.erase(heights.find(h)); // Remove building height
            }
```

```
            int currMax = *heights.rbegin(); // Get current max height
            if (currMax != prevMax) { // If height changed, add key point
                skyline.push_back({x, currMax});
                prevMax = currMax;
            }
        }
        return skyline;
    }
};
```