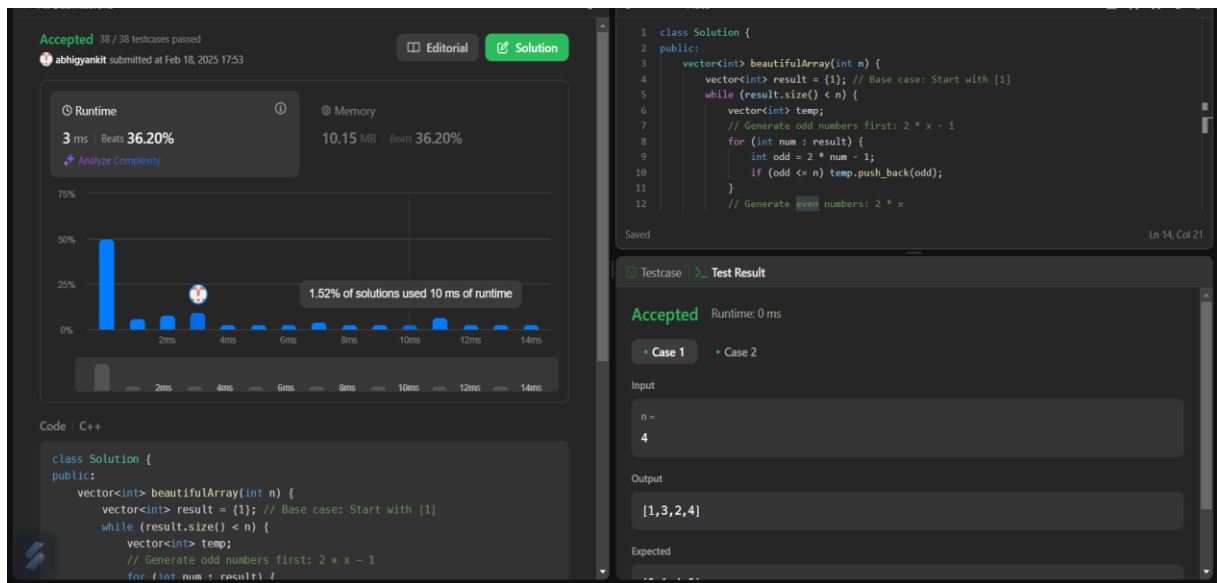


## Assignment 4

<b>Name: Agrini Mishra</b>	<b>Uid: 22BCS10821</b>
<b>Branch: BE_CSE</b>	<b>Semester: 6<sup>th</sup></b>
<b>Section: IOT_637-B</b>	<b>Subject: AP Lab II</b>

### 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;
}
};

```

The screenshot displays a C++ submission interface. On the left, the submission status is 'Accepted' with 44/44 testcases passed. The user 'abhiyankit' submitted it on Feb 18, 2025 at 17:58. The runtime statistics show 13 ms (Beats 72.56%) and 27.76 MB (Beats 71.41%). A histogram shows the distribution of runtime times, with a peak around 13 ms. The code is written in C++ and implements a solution for the 'Skyline Problem' using a sweep line algorithm. The right panel shows the test results for 'Case 1' and 'Case 2', both of which are 'Accepted' with a runtime of 0 ms. The input and output for Case 1 are shown, along with the expected output.

**Runtime Statistics:**

- Runtime: 13 ms (Beats 72.56%)
- Memory: 27.76 MB (Beats 71.41%)

**Test Results:**

Accepted Runtime: 0 ms

Case 1

Input:

```
buildings =
[[2,9,10],[3,7,15],[5,12,12],[15,20,10],[19,24,8]]
```

Output:

```
[[2,10],[3,15],[7,12],[12,0],[15,10],[20,8],[24,0]]
```

Expected:

```
[[2,10],[3,15],[7,12],[12,0],[15,10],[20,8],[24,0]]
```