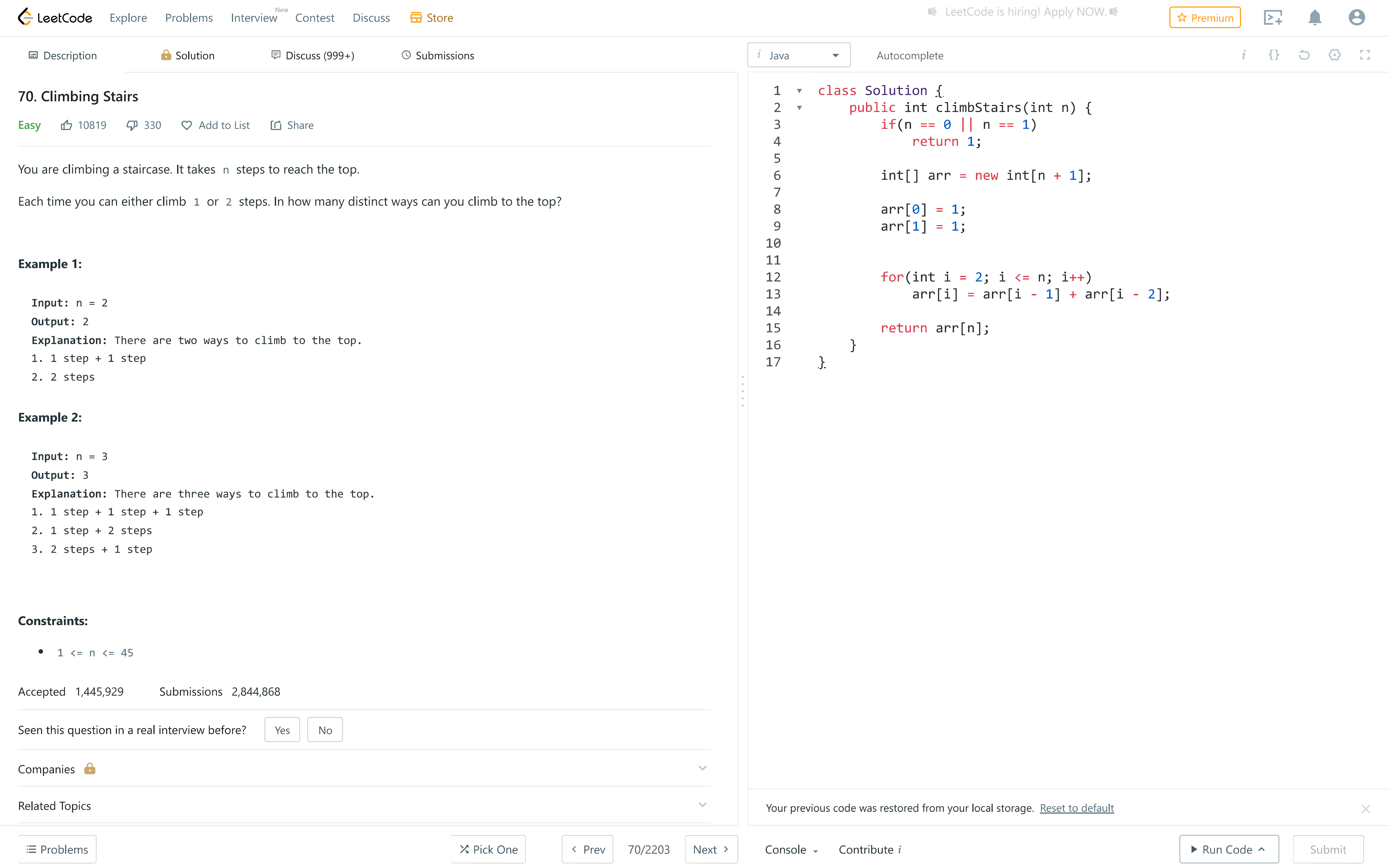

## 70. Climbing Stairs

You are climbing a staircase. It takes `n` steps to reach the top.

Each time you can either climb `1` or `2` steps. In how many distinct ways can you climb to the top?

**Example 1:**

```
Input: n = 2
Output: 2
Explanation: There are two ways to climb to the top.
1. 1 step + 1 step
2. 2 steps
```

**Example 2:**

```
Input: n = 3
Output: 3
Explanation: There are three ways to climb to the top.
1. 1 step + 1 step + 1 step
2. 1 step + 2 steps
3. 2 steps + 1 step
```

**Constraints:**

- `1 <= n <= 45`

Accepted 1,445,929 Submissions 2,844,868

Seen this question in a real interview before?

```java
class Solution {
    public int climbStairs(int n) {
        if(n == 0 || n == 1)
            return 1;

        int[] arr = new int[n + 1];

        arr[0] = 1;
        arr[1] = 1;


        for(int i = 2; i <= n; i++)
            arr[i] = arr[i - 1] + arr[i - 2];

        return arr[n];
    }
}
```

Your previous code was restored from your local storage. Reset to default