55. Jump Game

Medium 4382 334 Add to List Share

Given an array of non-negative integers, you are initially positioned at the first index of the array.

Each element in the array represents your maximum jump length at that position.

Determine if you are able to reach the last index.

Example 1:

```
Input: nums = [2,3,1,1,4]
Output: true
Explanation: Jump 1 step from index 0 to 1, then 3 steps to the last index.
```

Example 2:

```
Input: nums = [3,2,1,0,4]
Output: false
Explanation: You will always arrive at index 3 no matter what. Its maximum jump length is 0, which makes it impossible to reach the last index.
```

Constraints:

- `1 <= nums.length <= 3 * 10^4`
- `0 <= nums[i][j] <= 10^5`

Accepted 487,334 | Submissions 1,408,710


```java
class Solution {
    public boolean canJumpFromPosition(int position, int[] nums) {
     if (position == nums.length - 1) {
            return true;
        }

        int furthestJump = Math.min(position + nums[position], nums.length - 1);
        for (int nextPosition = position + 1; nextPosition <= furthestJump; nextPosition++) {
            if (canJumpFromPosition(nextPosition, nums)) {
                return true;
            }
        }
        return false;
    }
}
```

Testcase | Run Code Result | Debugger

Compile Error

```
Line 7: error: cannot find symbol [in __Driver__.java]
        boolean ret = new Solution().canJump(param_1);
                                     ^
  symbol:   method canJump(int[])
  location: class Solution
```