

# Jump Game

Solution

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<sup>4</sup>
- 0 <= nums[i][j] <= 10<sup>5</sup>

Java

```
1 class Solution {
2     public boolean canJump(int[] nums) {
3         if(nums.length == 0) return false;
4
5         int balance = nums[0];
6         int i = 0;
7         while(i < nums.length-1 && balance > 0) {
8
9             if(balance + i >= nums.length-1) return true;
10
11             i++;
12             balance--;
13             balance = Math.max(nums[i], balance);
14         }
15
16         return i >= nums.length-1;
17     }
18 }
```

☐ Custom Testcase ( [Contribute](#) )

?

Run Code

Submit