You are given an integer array nums. You are initially positioned at the array's first index, and each element in the array represents your maximum jump length at that position.

Return true *if you can reach the last index, or* false *otherwise*.

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 <= 10^4**
- **0 <= nums[i] <= 10^5**

Solution:

```
class Solution {
   public boolean canJump(int[] nums) {
       int boundary = 0;
    for(int i =0;i<=boundary;i++){
       boundary = Math.max(boundary,i+nums[i]);
       if(boundary >=nums.length-1)
       return true;
    }
    return false;
   }
}
```