

给定一个非负整数数组，你最初位于数组的第一个位置。  
数组中的每个元素代表你在该位置可以跳跃的最大长度。  
判断你是否能够到达最后一个位置。

#### 示例 1:

**输入:** [2,3,1,1,4]

**输出:** true

**解释:** 从位置 0 到 1 跳 1 步，然后跳 3 步到达最后一个位置。

#### 示例 2:

**输入:** [3,2,1,0,4]

**输出:** false

**解释:** 无论怎样，你总会到达索引为 3 的位置。但该位置的最大跳跃长度是 0，所以你永远不可能到达最后一个位置。

分析，这个跳跃为能跳的格子都能跳，跳过的格子，路上上的格子都能跳，动态的统计最大能跳的距离，若是最大的距离，大于等于最后一个位置，返回true，否则返回false

```
class Solution {
public:
    bool canJump(vector<int>& nums) {
        if(nums.size()<=1){
            return true;
        }
        int max=nums[0];
        if(max>=nums.size()-1){
            return true;
        }
        for(int i=1;i<=max;i++){
            if(i+nums[i]>max){
                max=i+nums[i];
                if(max>=nums.size()-1){
                    return true;
                }
            }
        }
        return false;
    }
};
```