

## 题目描述

### 55. 跳跃游戏

难度 中等

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: 我们可以先跳 1 步，从位置 0 到达 位置 1，然后再从位置 1 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: 无论如何，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。

## 解题思路

类似跳跃，利用贪心算法，从 $i=0$ 开始跳跃，每次跳跃最大距离并记下该边界  $dist = i + nums[i]$ ，从当前 $i$ 向前遍历，遍历的边界是动态的，就是 $dist$

## 代码实现

```
class Solution {
public:
    bool canJump(vector<int>& nums) {
        int m = nums.size(), dist=0, i=0;
        while (dist < m && i < m-1)
        {
            int temp = nums[i];
            dist = max(dist, i+temp);    //更新
            i++;
        }
        if (dist != m-1)
            return false;
        return true;
    }
};
```