# 45. Jump Game II

Total Accepted: 71517
Total Submissions: 278132
Difficulty: Hard
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.
Your goal is to reach the last index in the minimum number of jumps.
For example:
Given array A = [2,3,1,1,4]
The minimum number of jumps to reach the last index is 2. (Jump 1 step from index 0 to 1, then 3 steps to the last index.)
Note:
You can assume that you can always reach the last index.
Subscribe to see which companies asked this question

```
class Solution {
public:
    int jump(vector<int>& nums) {
        int count=0, cur=0, nextMax=0, i=0;
        while(cur < nums.size()-1)
        {
            count++;
            for(; i<=cur; ++i)
            {
                if(i+nums[i]>=nextMax) nextMax = i + nums[i];
            }
            cur = nextMax;
        }
        return count;
    }
};
```

# 55. Jump Game

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.

For example:
A = [2,3,1,1,4], return true.

A = [3,2,1,0,4], return false.

Subscribe to see which companies asked this question

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