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`.

Solution 1

I just iterate and update the maximal index that I can reach

```
bool canJump(int A[], int n) {
    int i = 0;
    for (int reach = 0; i < n && i <= reach; ++i)
        reach = max(i + A[i], reach);
    return i == n;
}
```

written by alexander7 original link here

Solution 2

Idea is to work backwards from the last index. Keep track of the smallest index that can "jump" to the last index. Check whether the current index can jump to this smallest index.

```
bool canJump(int A[], int n) {
    int last=n-1,i,j;
    for(i=n-2;i>=0;i--){
        if(i+A[i]>=last)last=i;
    }
    return last<=0;
}
```

written by lucastan original link here

Solution 3

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

written by xniu original link here