Question: https://leetcode.com/problems/jump-game-ii/

Let's illustrate it using the example nums = [2, 3, 1, 1, 4] in the problem statement. We are initially at position 0. Then we can move at most nums[0] steps from it. So, after one move, we may reach nums[1] = 3 or nums[2] = 1. So these nodes are reachable in 1 move. From these nodes, we can further move to nums[3] = 1 and nums[4] = 4. Now you can see that the target nums[4] = 4 is reachable in 2 moves.

Code:  
class Solution {

public int jump(int[] nums) {

int furthest=0, current=0, jump=0;

for(int i=0; i<nums.length-1; i++){

furthest= Math.max(furthest, nums[i]+i);

if(current == i){

jump++;

current=furthest;

}

}

return jump;

}

}

Github Link :<https://lnkd.in/ecwtJeaz>