1. H-Index II

Given an array of citations **sorted in ascending order** (each citation is a non-negative integer) of a researcher, write a function to compute the researcher’s h-index.

According to the [definition of h-index on Wikipedia](https://en.wikipedia.org/wiki/H-index): “A scientist has index *h* if *h* of his/her *N* papers have **at least** *h* citations each, and the other *N − h* papers have **no more than** *h* citations each.”

**Example:**

Input: citations = [0,1,3,5,6]  
Output: 3   
Explanation: [0,1,3,5,6] means the researcher has 5 papers in total and each of them had   
 received 0, 1, 3, 5, 6 citations respectively.   
 Since the researcher has 3 papers with at least 3 citations each and the remaining   
 two with no more than 3 citations each, her h-index is 3.

**Note:**

If there are several possible values for *h*, the maximum one is taken as the h-index.

**Follow up:**

* This is a follow up problem to [H-Index](https://leetcode.com/problems/h-index/description/), where citations is now guaranteed to be sorted in ascending order.
* Could you solve it in logarithmic time complexity?

**解** h-index：和爱丁顿数一个定义。对于一个从小到大排列好的数组，h-index为：

最简单是线性搜索，逐一判断即可。在有序数组中，可以使用二分查找

class Solution {  
public:  
 int hIndex(vector<int>& citations) {  
 int l = 0, r = citations.size()-1;  
 while(l <= r){  
 int mid = (l+r)/2;  
 if(citations.size()-mid == citations[mid])return citations.size()-mid;  
 if(citations.size()-mid > citations[mid])l = mid+1;  
 else r = mid-1;  
 }  
 return citations.size()-l;  
 }  
};