

```

1  /*
2  学校在拍年度纪念照时，一般要求学生按照 非递减 的高度顺序排列。
3  请你返回至少有多少个学生没有站在正确位置数量。该人数指的是：能让所有学生以 非递减 高度排列
   的必要移动人数。
4  示例：
5  输入：[1,1,4,2,1,3]
6  输出：3
7  解释：
8  高度为 4、3 和最后一个 1 的学生，没有站在正确的位置。
9
10 提示：
11      1 <= heights.length <= 100
12      1 <= heights[i] <= 100
13 来源：力扣（LeetCode）
14 链接：https://leetcode-cn.com/problems/height-checker
15 著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
16  */

```

分析:

- 方法一: 另开一片空间,存储排序后的数组,再逐一遍历两个数组的每个元素进行对比,不一一对应则错误计数加1.涉及到排序和遍历,综合时间复杂度为 $O(n \log n)$;
- 方法二: 由于题设中元素的范围为 $[1, 100]$
 - 开辟100个int长度的空间,遍历一遍用来存储各个取值出现的个数;
 - 根据各个取值个数,计算出每个可能取值应该出现的范围
 - 逐一判断原数组元素是否出现在它合法的位置,不合法则错误计数加1;
 - 涉及三次遍历,综合时间复杂度为 $O(n)$;
 - 致谢:@xiaxia153 提供这种思路.

方法一_C++:排序遍历法

```

1  class solution
2  {
3      public:
4          int heightChecker(vector<int>& heights)
5          {
6              vector<int> temp(heights);
7              int ret_val = 0;
8              sort(temp.begin(),temp.end());
9
10             for(int i = 0; i < heights.size() ; i++)
11             {
12                 if(temp[i]!=heights[i])
13                 {
14                     ret_val ++;
15                 }
16             }
17             return ret_val;
18         }
19     };
20     /*
21     执行结果:

```

```
22 通过
23 显示详情
24 执行用时 :8 ms, 在所有 C++ 提交中击败了47.05% 的用户
25 内存消耗 :8.7 MB, 在所有 C++ 提交中击败了100.00%的用户
26 */
```

方法二_C++:区间遍历法

```
1  /*
2
3     1 <= heights.length <= 100
4     1 <= heights[i] <= 100
5  */
6  class Solution
7  {
8      public:
9          int heightChecker(vector<int>& heights)
10         {
11             int arr[101] = {0,} ;
12             int ret_val = 0 ;
13             int i ;
14
15             for(i = 0 ; i < heights.size() ; i++)
16             {
17                 arr[heights[i]]++;
18             }
19
20             for(i = 1 ; i < 101 ; i++)
21             {
22                 arr[i] = arr[i] + arr[i-1];
23             }
24
25             for(i = 0; i < heights.size(); i++)
26             {
27                 /*
28                 i+1必须在( arr[heights[i]-1],arr[heights[i]] ] 之间才合法
29                 ,左开右闭区
30
31                 在此区间外就是错误的高度。
32                 @xiaxia153
33                 不需要排序算法，统计下每个高度有多少人，计算该高度的人应该位于哪个
34                 区间，不在这个区间的都是站错的；
35
36                 */
37                 if( ( i+1 <= arr[heights[i] - 1])
38                     ||( i+1 > arr[heights[i]] )
39                 )
40                 {
41                     ret_val++;
42                 }
43             }
44             return ret_val;
45         }
46     };
47
48  /*
49  执行结果:
50  通过
```

47 显示详情
48 执行用时 :4 ms, 在所有 C++ 提交中击败了86.49% 的用户
49 内存消耗 :8.4 MB, 在所有 C++ 提交中击败了100.00%的用户
50 */

AlimyBreak
2019.09.18