

剑指Offer（三十七）：数字在排序数组中出现的次数

🕒 2018年1月15日 21:01:16 🗨 10 🌡 3,966 °C 📄 编辑



一、前言

本系列文章为《剑指Offer》刷题笔记。

刷题平台：[牛客网](#)

书籍下载：[共享资源](#)

二、题目

统计一个数字在排序数组中出现的次数。

1、思路

既然是已经排序好的数组，那么第一个想到的就是二分查找法。

做法就是使用二分法找到数字在数组中出现的第一个位置，再利用二分法找到数字在数组中出现的第二个位置。时间复杂度为 $O(\log n + \log n)$ ，最终的时间复杂度为 $O(\log n)$ 。

举个例子，找到数字k在数组data中出现的次数。

数组data中，数字k出现的第一个位置：

我们对数组data进行二分，如果数组中间的数字小于k，说明k应该出现在中间位置的右边；如果数组中间的数字大于k，说明k应该出现在中间位置的左边；如果数组中间的数字等于k，并且中间位置的前一个数字不等于k，说明这个中间数字就是数字k出现的第一个位置。

同理，数字k出现的最后一个位置，也是这样找的。但是判断略有不同。我们使用两个函数分别获得他们。

2、代码

C++：

C+


```
1 class Solution {
2 public:
3     int GetNumberOfK(vector<int> data ,int k) {
4         int length = data.size();
5         if(length == 0){
6             return 0;
7         }
8         int first = GetFirstK(data, k, 0, length - 1);
9         int last = GetLastK(data, k, 0, length - 1);
10        if(first != -1 && last != -1){
11            return last - first + 1;
12        }
13        return 0;
14    }
15 private:
16     // 迭代实现找到第一个K
17     int GetFirstK(vector<int> data, int k, int begin, int end){
18         if(begin > end){
19             return -1;
20         }
21         int middleIndex = (begin + end) >> 1;
22         int middleData = data[middleIndex];
23         if(middleData == k){
24             return middleIndex;
25         }
26         if(middleData < k){
27             begin = middleIndex + 1;
28         }
29         else{
30             end = middleIndex - 1;
31         }
32         return GetFirstK(data, k, begin, end);
33     }
34     // 迭代实现找到最后一个K
35     int GetLastK(vector<int> data, int k, int begin, int end){
36         if(begin > end){
37             return -1;
38         }
39         int middleIndex = (begin + end) >> 1;
40         int middleData = data[middleIndex];
41         if(middleData == k){
42             return middleIndex;
43         }
44         if(middleData < k){
45             begin = middleIndex + 1;
46         }
47         else{
48             end = middleIndex - 1;
49         }
50         return GetLastK(data, k, begin, end);
51     }
52 }
```

```
25         if((middleIndex > 0 && data[middleIndex - 1] != k) || middleIndex == 0){
26             return middleIndex;
27         }
28         else{
29             end = middleIndex - 1;
30         }
31     }
32     else if (middleData > k){
33         end = middleIndex - 1;
34     }
35     else{
36         begin = middleIndex + 1;
37     }
38     return GetFirstK(data, k, begin, end);
39 }
40 // 循环实现找到最后一个K
41 int GetLastK(vector<int> data, int k, int begin, int end){
42     int length = data.size();
43     int middleIndex = (begin + end) >> 1;
44     int middleData = data[middleIndex];
45
46     while(begin <= end){
47         if(middleData == k){
48             if(middleIndex < length - 1 && data[middleIndex + 1] != k) || middleIndex == length - 1){
49                 return middleIndex;
50             }
51             else{
52                 begin = middleIndex + 1;
53             }
54         }
55         else if(middleData > k){
56             end = middleIndex - 1;
57         }
58         else{
59             begin = middleIndex + 1;
60         }
61         middleIndex = (begin + end) >> 1;
62         middleData = data[middleIndex];
63     }
64     return -1;
65 }
66 };
```

Python:

Python有现成的函数供我们使用:

<pre>1 # -*- coding:utf-8 -*- 2 class Solution: 3     def GetNumberOfK(self, data, k): 4         # write code here 5         return data.count(k);</pre>	Pytho
--	-------



微信公众号

分享技术, 乐享生活: 微信公众号搜索「JackCui-AI」关注一个在互联网摸爬滚打的潜行者。

领袖和跟风者的区别就在于创新。--- 乔布斯