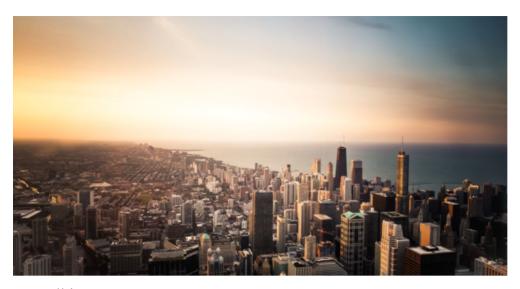
剑指Offer(三十七): 数字在排序数组中出现的次数

© 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(logn + logn),最终的l间复杂度为O(logn)。

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。但是判断少有不同。我们使用两个函数分别获得他们。

2、代码

C++:

```
C+
       class Solution {
 2
3
4
5
6
7
8
9
10
             int GetNumberOfK(vector<int> data ,int k) {
  int length = data.size();
  if(length == 0){
                           return 0;
                    int first = GetFirstK(data, k, 0, length - 1);
int last = GetLastK(data, k, 0, length - 1);
if(first != -1 && last != -1){
    return last - first + 1;
11
12
                    return 0;
14
15
16
      private:
              // 迭代实现找到第一个K
17
18
19
                    GetFirstK(vector<int> data, int k, int begin, int end){
                    if(begin > end){
    return -1;
20
21
22
                    int middleIndex = (begin + end) >> 1;
int middleData = data[middleIndex];
                    if(middleData == k){
24
```

```
if((middleIndex > 0 && data[middleIndex - 1] != k) || middleIndex == 0){
262728333333333334444454474544755555555566616263465
                              return middleIndex;
                        else{
                             end = middleIndex - 1;
                 else if (middleData > k){
                       end = middleIndex - 1;
                 else{
                       begin = middleIndex + 1;
                 return GetFirstK(data, k, begin, end);
          }
// 循环实现找到最后一个K
int GetLastK(vector<int> data, int k, int begin, int end){
    int length = data.size();
    int middleIndex = (begin + end) >> 1;
    int middleData = data[middleIndex];
                 while(begin <= end){
   if(middleData == k){
      if((middleIndex < length - 1 && data[middleIndex + 1] != k) || middleIndex == length - 1){</pre>
                                    return middleIndex;
                                    begin = middleIndex + 1;
                       else if(middleData > k){
   end = middleIndex - 1;
                        else{
                             begin = middleIndex + 1;
                       middleIndex = (begin + end) >> 1;
middleData = data[middleIndex];
                  return -1;
66
```

Python:

Python有现成的函数供我们使用:

```
Pytho

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



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