



## 13、折半查找

24计算机考研成员一战成硕！



题目描述：

折半查找的代码默写

🕒 倒计时

### 1、知识点及难度



解人：多动症男孩

难度：简单

知识点：

1. 折半查找（Binary Search）是一种在有序数组中快速查找目标值的算法。它的基本思想是通过将目标值与数组的中间元素进行比较，从而将搜索范围缩小一半，然后重复这个过程直到找到目标值或确定目标值不存在。

2. 折半查找要求目标数组是有序的，否则无法使用该算法进行查找。

**时间复杂度：**  $O(\log n)$ ，其中  $n$  是数组的长度。每次查找都将搜索范围缩小一半，所以最坏情况下的查找次数为  $\log n$

**空间复杂度：**  $O(1)$ ，算法只使用了常数变量来保存辅助信息，没有使用额外的辅助空间。

## 2、算法题

### 思路

1. 初始化左右指针left和right,分别指向数组的起始位置和结束位置。
2. 重复以下步骤，直到left指针大于right指针的时候停止。
  - a. 将中间元素与目标值进行比较。
  - b. 如果中间元素等于目标值，表示找到了目标值，返回中间元素的索引。
  - c. 如果中间元素大于目标值，则标指目标值可能在该中间元素的左侧，更新右指针为mid-1。
  - d. 如果中间元素小于目标值，则标指目标值可能在该中间元素的右侧，更新左指针为mid+1。
3. 循环结束仍然没有找到目标值，则表示目标值不存在于数组中。

### 基本实现-C

```
1 #include<stdio.h>
2
3 int binarySearch(int arr[], int length, int target) {
4
5     // 初始化左右指针
6     int left = 0;
7     int right = n - 1;
8
9     while (left <= right) {
10         int mid = (left + right) / 2;
11
12         if(arr[mid] == target) {
13             return mid; // 返回目标元素的下标
14         }else if (arr[mid] < target) {
15             left = mid + 1;
16         }else {
17             right = mid - 1
18         }
```

```
19     }
20
21     return -1; // 未找到, 返回一个错误表示
22 }
```

## 基本实现C++

```
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  int binarySearch(vector<int> &arr, int target) {
7
8      // 初始化左右指针
9      int left = 0;
10     int right = arr.size() - 1;
11
12     while(left <= right) {
13         int mid = (left + right) / 2;
14
15         if (arr[mid] == target) {
16             return mid; // 返回目标元素的下标
17         }else if(arr[mid] < target) {
18             left = mid + 1;
19         }else {
20             right = mid - 1;
21         }
22     }
23
24     return -1; // 未找到, 返回一个错误表示
25 }
```

## 3、总结



### 总结栏

蓝蓝B站首页: [蓝蓝希望你上岸呀B站首页](#)

蓝蓝公众号: [算法训练营9分计划](#)

蓝蓝知识星球介绍: [📖 关于知识星球的权益](#)

### 如何在星球打卡记录:

- 计算机考研数据结构算法专项day[1/60]:
- 学习内容: 最好能发出自己写的图片
- 遇到的问题: 如果无就不用写了
- 小结: 这部分一周写一次即可。