



## 17、快速排序

24计算机考研成员一战成硕！



题目描述：

回忆快速排序过程及代码实现。

🕒 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. 快速排序（Quick Sort）是一种常用的排序算法，其基本思想是通过递归地将数组划分为较小和较大的两个子数组，然后对子数组进行排序，最终完成整个数组的排序。
2. 分区操作是快速排序的核心，通过选择基准元素和两个指针的移动，将数组分为两部分。

3. 可以通过递归方式实现快速排序，将左右两个子数组分别进行排序。

**时间复杂度：** $O(n\log n)$ , 其中 $n$ 是数组的长度。

**空间复杂度：** $O(\log n)$ , 每次递归调用都需要额外的栈空间。

## 2、算法题

### 思路

1. 选择一个基准元素（通常选择数组的第一个元素）。
2. 将数组分为两部分，比基准元素小的或等于的元素放在左边，比基准元素大的元素放在右边。这一步称为分区操作。
3. 对左右两个子数组递归的进行快速排序。

### 基本实现-C

```
1 #include <stdio.h>
2
3 // 快速排序函数
4 void quickSort(int* arr, int left, int right) {
5     if(left >= right) return; // 基线条件，只有一个元素时也无需操作
6
7     int pivot = arr[left]; // 选择第一个元素作为基准元素
8     int low = left, int high = right;
9
10    while (low < high) {
11        while (low < high && arr[high] >= pivot) high--; // 从右往左找到第一个小
        于基准元素的值
12        arr[low] = arr[high];
13
14        while (low < high && arr[low] <= pivot) low++; // 从左往右找到第一个小、大
        于基准元素的值
15        arr[high] = arr[low];
16    }
17
18    arr[low] = pivot; // 将基准元素放到正确的位置
19
20    // 递归的对左右两部分进行快速的排序
21    quickSort(arr, left, low - 1); // 左子数组
22    quickSort(arr, low + 1, right); // 右子数组
```

```
23
24 }
```

## 基本实现C++

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 // 快速排序函数
6 void quickSort(vector<int> &arr, int left, int right) {
7     if(left >= right) return; // 基线条件, 只有一个元素时无需排序
8
9     int pivot = arr[left];
10    int low = left, high = right;
11
12    while(low < high) {
13        while(low < high && arr[high] >= pivot) high--;
14        arr[low] = arr[high];
15        while(low < high && arr[low] <= pivot) low++;
16        arr[high] = arr[low];
17    }
18    arr[low] = pivot; // 将基准元素放到正确的位置
19
20    // 递归的对左右两部分进行快速排序
21    quickSort(arr, left, low - 1);
22    quickSort(arr, low + 1, right);
23 }
24
```

## 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。