



17、快速排序

24计算机考研成员一战成硕！

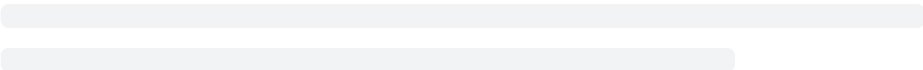


题目描述：

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  /*
2  快速排序思想：
3  一共三步
4  1. 找任意一个元素x作为标准，可以是l，可以是r，（可以是l+r）/2，可以随机。
5  2. 把小于等于x的放在左边，大于等于x的放在右边，等于x既有可能在左边也有可能右边
6  3. x元素把数组分割成任意两部分，然后在对这两部分进行递归
7  */
8  void Quicksort(int q[] , int l , int r) {
9      //如果只有一个元素直接返回
10     if( l >= r )return;
11     //第一部分：选取最左边的元素为x，因为边界不能动，并且使用do...while语句，所以把i和j
    分别设成边界元素的后一个位置
12     int x = q[l], i = l + 1, j = r + 1;
13     //第二部分：
14     while( i < j ) {
15         //从左边寻找一个不满足小于x的元素
16         do i++; while( q[i] < x );
17         //从右边寻找一个不满足大于x的元素
18         do j--; while( q[j] > x );
19         //并且在i < j情况在再交换
20         //如果当不满足i<j说明此时以j为下标的分界线把数组分为了两个部分
21         if( i < j )swap(q[i] , q[j]);
22     }
23     //第三部分：
```

```

24 //通过上面的循环把利用元素x把数组分成了逻辑上的两个部分，物理上是用下标j分为了两个部
    分，然后再分别对这两部分进行递归就行
25 // 例如：原数组：5, 2, 7,9,24,-1,7,354
26 //一次quicksort之后[-1, 2 (j) , 7, 9, 24, 5 (x) , 7, 354], j前面的都是小于x的, j
    后面都是大于x的，但是前后不保证有序，需要递归
27 Quicksort(q , l , j);
28 Quicksort(q , j + 1 , r);
29 }

```

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。