

第13课 快速排序

快速排序的基本思想

快速排序（Quicksort）是对冒泡排序的一种改进。

它的基本思想是：通过一趟排序将要排序的数据分割成独立的两部分，其中一部分的所有数据都比另外一部分的所有数据都要小，然后再按此方法对这两部分数据分别进行快速排序，整个排序过程可以递归进行，以此达到整个数据变成有序序列。

快速排序的算法分析

快速排序算法通过多次比较和交换来实现排序，其排序流程如下：

- (1)首先设定一个分界值，通过该分界值将数组分成左右两部分。
- (2)将大于或等于分界值的数据集中到数组右边，小于分界值的数据集中到数组的左边。此时，左边部分中各元素都小于或等于分界值，而右边部分中各元素都大于或等于分界值。
- (3)然后，左边和右边的数据可以独立排序。对于左侧的数组数据，又可以取一个分界值，将该部分数据分成左右两部分，同样在左边放置较小值，右边放置较大值。右侧的数组数据也可以做类似处理。
- (4)重复上述过程，可以看出，这是一个递归定义。通过递归将左侧部分排好序后，再递归排好右侧部分的顺序。当左、右两个部分各数据排序完成后，整个数组的排序也就完成了。

快速排序的排序步骤

设要排序的数组是 $A[0].....A[N-1]$ ，首先任意选取一个数据（通常选用数组的第一个数）作为关键数据，然后将所有比它小的数都放到它左边，所有比它大的数都放到它右边，这个过程称为一趟快速排序。

值得注意的是，快速排序不是一种稳定的排序算法，也就是说，多个相同的值的相对位置也许会在算法结束时产生变动。

一趟快速排序的算法是：

- 1) 设置两个变量*i*、*j*，排序开始的时候： $i=0$ ， $j=N-1$ ；
- 2) 以第一个数组元素作为关键数据，赋值给key，即 $key=A[0]$ ；
- 3) 从*j*开始向前搜索，即由后开始向前搜索($j--$)，找到第一个小于key的值A[j]，将A[j]和A[i]的值交换；
- 4) 从*i*开始向后搜索，即由前开始向后搜索($i++$)，找到第一个大于key的A[i]，将A[i]和A[j]的值交换；
- 5) 重复第3、4步，直到*i=j*；（3,4步中，没找到符合条件的值，即3中A[j]不小于key,4中A[i]不大于key的时候改变j、i的值，使得 $j=j-1$ ， $i=i+1$ ，直至找到为止。找到符合条件的值，进行交换的时候i，j指针位置不变。另外， $i==j$ 这一过程一定正好是*i+*或*j-*完成的时候，此时循环结束）。

青少年软件编程等级考试Python标准公益课 (3级)

假设一开始序列a是：5，3，7，6，4，1，0，2，9，10，8。此时， $\text{ref}=5$ ， $i=0$ ， $j=10$ ，从后往前找，第一个比5小的数是 $a[7]=2$ ，因此序列为：2，3，7，6，4，1，0，5，9，10，8。此时 $i=0$ ， $j=7$ ，从前往后找，第一个比5大的数是 $a[2]=7$ ，因此序列为：2，3，5，6，4，1，0，7，9，10，8。此时， $i=2$ ， $j=7$ ，从第7位往前找，第一个比5小的数是 $a[6]=0$ ，因此：2，3，0，6，4，1，5，7，9，10，8。此时， $i=2$ ， $j=6$ ，从第2位往后找，第一个比5大的数是 $a[3]=6$ ，因此：2，3，0，5，4，1，6，7，9，10，8。此时， $i=3$ ， $j=6$ ，从第6位往前找，第一个比5小的数是 $a[5]=1$ ，因此：2，3，0，1，4，5，6，7，9，10，8。此时， $i=3$ ， $j=5$ ，从第3位往后找，直到第6位才有比5大的数(正常区间已无满足的数)，这时， $i=j=6$ ， ref 成为一条分界线，它之前的数都比它小，之后的数都比它大，对于前后两部分数，可以采用同样的方法来排序。

青少年软件编程等级考试Python标准公益课（3级）

参考脚本代码：

```
def quicksort(list, p, r):  
    if p < r:  
        q = partition(list, p, r)  
        quicksort(list, p, q)  
        quicksort(list, q + 1, r)  
  
def partition(list, p, r):  
    i = p - 1  
    for j in range(p, r):  
        if list[j] <= list[r]:  
            i += 1  
            list[i], list[j] = list[j], list[i]  
    list[i + 1], list[r] = list[r], list[i + 1]  
    return i  
  
list1 = [2, 8, 7, 1, 3, 5, 6, 4]  
quicksort(list1, 0, len(list1) - 1)  
print(list1)
```