

# 轻轻松松明白快速排序

作者：夏侯佐鑫      日期：2012/2/17

## 1、快速排序算法自然语言描述

对一个典型子数组  $A[p..r]$  排序的分治过程为三个步骤：

- (1) 分解： $A[p..r]$  被划分为两个（可能空）的子数组  $A[p..q-1]$  和  $A[q+1..r]$ ，使得
$$A[p..q-1] \leq A[q] \leq A[q+1..r]$$
- (2) 解决：通过递归调用快速排序，对子数组  $A[p..q-1]$  和  $A[q+1..r]$  排序。
- (3) 合并。

## 2、快速排序算法的伪代码描述

### QUICKSORT(A, p, r)

```
1  if p < r
2  then q ← PARTITION(A, p, r)    //关键
3  QUICKSORT(A, p, q - 1)
4  QUICKSORT(A, q + 1, r)
```

快速排序算法的关键是 PARTITION 过程，它对  $A[p..r]$  进行就地重排：

### PARTITION(A, p, r)

```
1  x ← A[r]
2  i ← p - 1
3  for j ← p to r - 1
4  do if A[j] ≤ x
5  then i ← i + 1
6  exchange A[i] <-> A[j]
7  exchange A[i + 1] <-> A[r]
8  return i + 1
```

## 3、排序过程举例

4 (主元)	1	7	8	3	5	6	4
p/i	j						r

因为  $1 < 4$ ，所以  $i=i+1$ ， $a[i]$  与  $a[j]$  交换

4 (主元)	1	7	8	3	5	6	4
p	i	j					r

因为  $7 > 4$ ，所以  $i$  不变， $j++$

4 (主元)	1	7	8	3	5	6	4
p	i		j				r

同理，只要  $a[j] \leq a[p]=4$ ，则  $i=i+1$  之后进行  $a[j]$  与  $a[i]$  交换，接着上面完成的第一趟 PARTITION 过程

4 (主元)	1	7	8	3	5	6	4
p	i		j				r

3<4, i=i+1 后交换, 交换后结果

4 (主元)	1	3(原 7)	8	7(原 3)	5	6	4
p		i		j			r

5>4, 不交换, j++

4 (主元)	1	3	8	7	5	6	4
p		i			j		r

6>4, 不交换, j++

4 (主元)	1	3	8	7	5	6	4
P		i				j	r

4=4, i=i+1 后交换, 交换后结果

4 (主元)	1	3	4(原 8)	7	5	6	8(原 4)
p			i				r/j

最后还别忘了最后一次交换过程  $a[p] \leftrightarrow a[i]$ , 把主元也就是分组的参考值放在中间。  
最后第一次分好组结果为

4 (原 4)	1	3	4(原 4)	7	5	6	8
p			i				r/j

至此, 以  $a[i]$  为边界,  $a[i]$  左边元素全都小于或等于  $a[i]$ , 而右边的元素都大于  $a[i]$ , 然后再对左右两边进行相同的递归划分就可以完成排序过程。

#### 4、快速排序算法两种 C 语言实现方式

/\*方法 1: 首元为主元的快速排序(按数组下标), 最初 i, j 状态如下所示\*/

4 (主元)	1	7	8	3	5	6	4
p/i	j						r

```
void QuickSortBegDom(int a[], int p, int r)
```

```
{
```

```
    int i = p;
```

```

int j = p + 1;

int temp = 0; //交换变量

if(p<r)
{
    for(j=p+1; j<=r; j++)
    {
        if(a[j]<a[p])
        {
            i = i + 1;

            temp = a[i];

            a[i] = a[j];

            a[j] = temp;

        }
    }

    temp = a[i];

    a[i] = a[p];

    a[p] = temp;

    /*递归调用*/

    QuickSortBegDom(a, p, i-1);

    QuickSortBegDom(a, i+1, r);

}
}

```

/\*方法二：尾元为主元的快速排序(按数组下标)，最初 i, j 状态如下所示\*/

(NULL)	4	1	7	8	3	5	6	4
i	p/j							r

```
void QuickSortEndDom(int a[], int p, int r)
```

```
{
```

```
    int i = p-1;
```

```
    int j = p;
```

```
    int temp = 0; //元素交换
```

```
    if(p<r)
```

```
    {
```

```
        /*Partition*/
```

```
        for(j=p; j<r; j++)
```

```
        {
```

```
            if(a[j]<=a[r])
```

```
            {
```

```
                i = i + 1;
```

```
                temp = a[j];
```

```
                a[j]=a[i];
```

```
                a[i]=temp;
```

```
            }
```

```
        }
```

```
        temp = a[i+1];
```

```
a[i+1] = a[r];
```

```
a[r] = temp;
```

```
/*递归调用*/
```

```
QuickSortEndDom(a, p, i);
```

```
QuickSortEndDom(a, i+2, r);
```

```
}
```

```
}
```