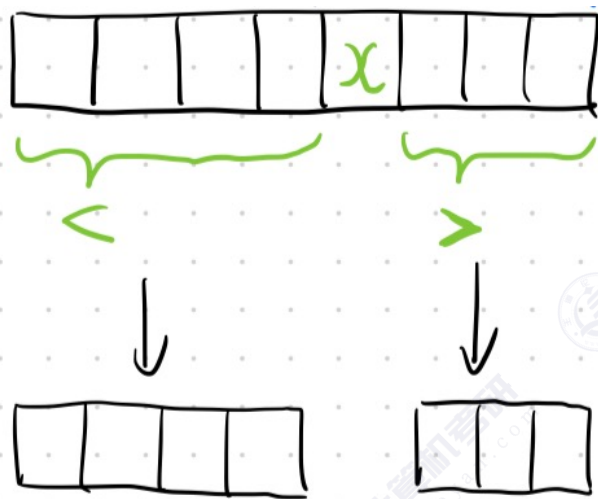


王道计算机考研强化课

# 算法题备考

快速排序算法  
(运用“**划分**”思想)

## 快速排序的“划分”思想

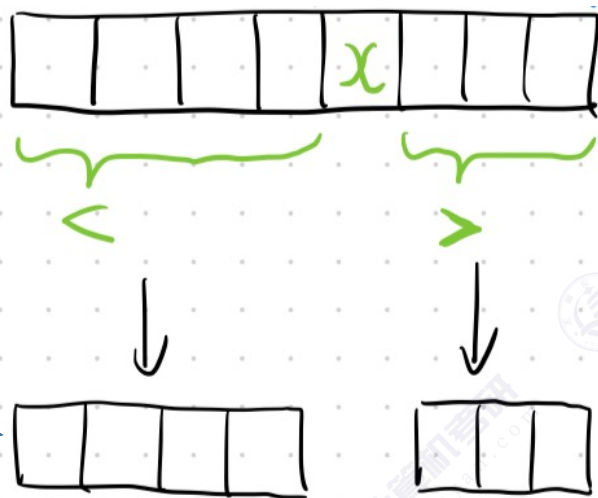


```
void Qsort (int A[], int L, int R) {  
    if (L >= R) return; // 递归终止  
    int M = huafen (A, L, R);  
    Qsort (A, L, M-1); // 左半部分快排  
    Qsort (A, M+1, R); // 右半部分  
}
```

```
int huafen (int A[], int L, int R) {  
    int mid = A[L];  
    while (L < R) {  
        while (A[R] >= mid) R--;  
        A[L] = A[R];  
        while (A[L] <= mid) L++;  
        A[R] = A[L];  
    }  
    A[L] = mid;  
    return L;  
}
```

“划分”函数返回值 =  $M$ ,  
说明此次选取的枢轴元素  
是数组中第  $M+1$  小的元素

## 快速排序的“划分”思想



更小的 $M$   
个元素

可以利用“划分”函数：

1. 找到数组中第  $k$  小（或第  $k$  大）的元素。
2. 找到数组中更小的（或更大的） $k$  个元素；
3. 把数组用划分为左右两个部分（更小的 $k$ 个元素在左、更大 $n-k$ 个元素在右边）



我有一个很大胆的想法

```
int huafen(int A[], int L, int R){  
    int mid = A[L];  
    while (L < R){  
        while (A[R] >= mid) R--;  
        A[L] = A[R];  
        while (A[L] <= mid) L++;  
        A[R] = A[L];  
    }  
    A[L] = mid;  
    return L;  
}
```

## 例：使用“划分”函数找到数组中第k小的元素

```
int huafen(int A[], int L, int R){  
    int mid = A[L];  
    while (L < R){  
        while (A[R] >= mid) R--;  
        A[L] = A[R];  
        while (A[L] <= mid) L++;  
        A[R] = A[L];  
    }  
    A[L] = mid;  
    return L;  
}
```

Handwritten annotations in the code:

- A yellow highlight on the condition `&& L < R` at the top right of the function body.
- A red arrow pointing down from the `&& L < R` highlight to the `while (L < R)` loop condition.
- A yellow highlight on the condition `A[R] >= mid` inside the first inner loop.
- A yellow highlight on the condition `A[L] <= mid` inside the second inner loop.
- A red arrow pointing up from the `A[L] <= mid` highlight to the `&& L < R` highlight.

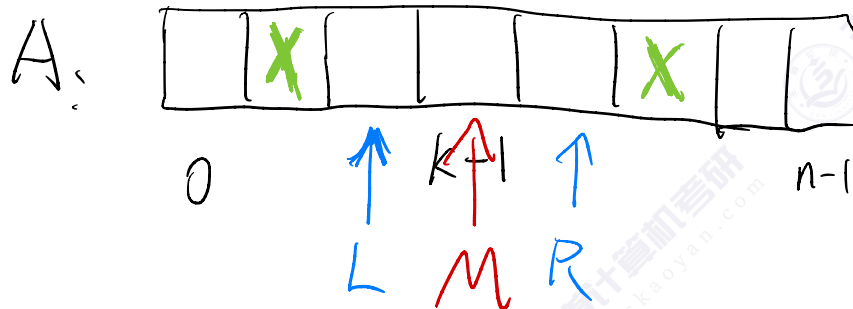


例：使用“划分”函数找到数组中第k小的元素

$k-1$

$A[n]$ , 下标  $0 \sim n-1$

```
int huafen(int A[], int L, int R){
    int mid = A[L];
    while (L < R){
        while (A[R] >= mid) R--;
        A[L] = A[R];
        while (A[L] <= mid) L++;
        A[R] = A[L];
    }
    A[L] = mid;
    return L;
}
```



```
int func(int A[], int n, int k){
    int L=0, R=n-1, M=0;
    while(1){
        M = huafen(A, L, R);
        if (M == k-1) break;
        else if (M > k-1) R = M-1;
        else if (M < k-1) L = M+1;
    }
    return A[k-1];
}
```



# 快速排序的“划分”思想算法题实战运用

Todo:

王道书 8.3.3\_大题\_5

王道书 8.3.3\_大题\_7

注：本题为王道书 8.3.3 大题\_5

## 王道书课后习题

算法题

试编写一个算法，使之能够在数组  $L[1 \dots n]$  中找出第  $k$  小的元素（即从小到大排序后处于第  $k$  个位置的元素）。



43. 已知由  $n$  ( $n \geq 2$ ) 个正整数构成的集合  $A = \{a_k | 0 \leq k < n\}$ , 将其划分为两个不相交的子集  $A_1$  和  $A_2$ , 元素个数分别是  $n_1$  和  $n_2$ ,  $A_1$  和  $A_2$  中元素之和分别为  $S_1$  和  $S_2$ 。设计一个尽可能高效的划分算法, 满足  $|n_1 - n_2|$  最小且  $|S_1 - S_2|$  最大。要求:

- (1) 给出算法的基本设计思想。
- (2) 根据设计思想, 采用 C 或 C++ 语言描述算法, 关键之处给出注释。
- (3) 说明你所设计算法的平均时间复杂度和空间复杂度。