



14、找出第k小的元素

24计算机考研成员一战成硕！



题目描述：

在数组 $L[1...n]$ 中找出第K小的元素(从小到大处于第k个位置的元素)

相关内容



算法题解

思路

1.先排序

2.直接输出值（题解没给最优解）

也有基于快排或者堆排的更低复杂度的算法，但是不建议。

思路一：不知道你们看到找到前k个元素的时候，你会想到什么，不知道会不会想到王道课后习题里面考察的一类题：经过第一轮排序，某个元素已经再正确的位置的排序有什么...例如冒泡排序，可以每一次都把最小的那个元素冒泡到第一个，题目要求找到第k小的元素，我们是不是只需要冒k个泡就行，所以时间复杂度可以在 $O(kn)$ ，选择排序同理

思路二：还有一种方法是建堆，建立大根堆或者小根堆，大根堆是最大的元素在最上面，小根堆是最小的元素在最上面，我们可以把整个数组建一个小根堆，然后每次弹出最上面那个元素，然后根据堆的特性，最小的元素会上浮，一次上浮的时间复杂度是 $\log n$ ，建堆的时间复杂度是 on ，所以弹出前 k 个元素的时间复杂度是不是 $O(n+k*\log n)$

快排模板建议直接背



知识点

```
1 void Quicksort(int q[] , int l , int r) {
2     //如果只有一个元素直接返回
3     if( l >= r )return;
4     //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
5     int x = q[l], i = l - 1, j = r + 1;
6     //第二部分：
7     while( i < j ) {
8         //从左边寻找一个不满足小于x的元素
9         do i++; while( q[i] < x );
10        //从右边寻找一个不满足大于x的元素
11        do j--; while( q[j] > x );
12        //并且在 i < j 情况在再交换
13        //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
14        if( i < j )swap(q[i] , q[j]);
15    }
16    //第三部分：
17    //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
18    // 例如：原数组：5, 2, 7,9,24,-1,7,354
19    //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
20    Quicksort(q , l , j);
21    Quicksort(q , j + 1 , r);
22 }
23 void main() {
24     int k;
25     cin >> k;
26     int arr[] = { 5,7,5,3,1 };
27     Quicksort(arr , 0 , 4);
28     cout << arr[k - 1];
29 }
```



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。