

```

1 //日期: 2018/ 时间:
2 /*随机数说明, 需要包含头文件<time.h>或者<stdlib.h>.
3 首先在main()函数开头加上srand((unsigned)time(NULL));
4 rand()函数产生一个范围是[0,RAND_MAX]的随机数。rand()/RAND_MAX就会产生一个[0,1]的 ↗
   随机数。
5 想要生成大范围的随机数, 就要用 (rand()/RAND_MAX)*(b-a) + a , 再进行四舍五入取整转 ↗
   换成int型
6 最终(int)(round((rand()/RAND_MAX)*(b-a) + a)) -----就产生了[a,b]范围内的随 ↗
   机数。闭区间
7 */
8 #include <stdio.h>
9 #include <stdlib.h>
10 #include <time.h>
11 #include <math.h>
12 #include <algorithm>
13 using namespace std;
14
15 //two pointer方法。
16 //整序列中的元素, 使当前序列最左端的元素调整后满足, 最左侧所有元素都不超过该元素, 右 ↗
   侧所有元素都大于该元素
17 int Partition(int A[],int left,int right){
18     int temp = A[left];    //将A[left]存放至临时变量temp
19     while(left < right){
20         while(left < right && A[right] > temp) right--;
21         A[left]=A[right];
22         while(left < right && A[left] <= temp ) left++;
23         A[right]=A[left];
24     }
25     A[left] = temp;
26     return left;
27 }
28
29 //快排的思想是: 1) 调整序列中的元素, 使当前序列最左端的元素调整后满足, 最左侧所有元素 ↗
   都不超过该元素, 右侧所有元素都大于该元素
30 // 2) 对该元素的左侧和右侧分别进行递归, 直到当前调整区间的长度不超过1
31 //left与right初值为序列首尾下标 (例如1和n)
32 void quickSort(int A[],int left,int right){
33     if(left < right){    //当前区间长度超过1
34         //将[left,right]按A[left]一分为二
35         int pos = Partition(A, left,right);
36         quickSort(A,left,pos-1);    //对左子区间递归进行快速排序
37         quickSort(A,pos+1,right);    //对右子区间递归进行快速排序
38     }
39 }
40
41 //快速排序产生复杂度为 $O(n^2)$ , 主要原因在于, 当序列中元素接近有序时, 主元没有把当前区 ↗
   间划分为两个长度接近的子区间
42 //规避方法是: 在A[left,right]中随机选取一个主元,
43 //因此不妨生成一个范围在[left,right]内的随机数p, 然后以A[p]作为主元进行划分。
44 //具体做法是: 将A[p]与A[left]交换, 然后按照原先的Partition函数的写法即可
45 int randPartition(int A[],int left,int right){
46     int p = (int)(round(1.0*rand()/RAND_MAX * (right-left) + left));
47     swap(A[p],A[left]);    //include <algorithm>
48
49     int temp = A[left];    //将A[left]存放至临时变量temp
50     while(left < right){

```

```
51     while(left < right && A[right] > temp) right--;  
52     A[left]=A[right];  
53     while(left < right && A[left] <= temp ) left++;  
54     A[right]=A[left];  
55 }  
56 A[left] = temp;  
57 return left;d  
58 }  
59  
60 int main(){  
61  
62  
63     return 0;  
64 }  
65  
66
```