学习情况表

|  |  |  |  |
| --- | --- | --- | --- |
| **姓名** | 杜金瑞 | **学号** | 2020905073 |
| **学院** | 信息工程学院 | **专业** | 计算机科学与技术 |

|  |
| --- |
| 学习情况简述 |
| C:\Users\29043\Documents\Tencent Files\2904326062\Image\C2C\7C740D3503D14713F649E1DBE7FDE83F.jpg  C:\Users\29043\Documents\Tencent Files\2904326062\Image\C2C\BA748872926AED9AFED00EC325FBB0DE.jpg  C:\Users\29043\Documents\Tencent Files\2904326062\Image\C2C\3D4B78BEA637ED293753014D41C8BB83.jpg |
| 本周练习过的代码（例） |

import java.util.Arrays;  
import java.util.Random;  
  
public class QuickSort {  
  
 private QuickSort(){}  
  
  
 public static <E extends Comparable<E>> void sort(E[] arr){  
  
 Random rnd=new Random();  
  
 *sort*(arr,0,arr.length-1,rnd);  
 }  
  
 private static <E extends Comparable<E>> void sort(E[] arr,int l,int r,Random rnd){  
  
 if(l>=r) return;  
  
 int p=*partition*(arr,l,r,rnd);  
 *sort*(arr,l,p-1,rnd);  
 *sort*(arr,p+1,r,rnd);  
 }  
  
 private static <E extends Comparable<E>> int partition(E[] arr,int l,int r,Random rnd){  
  
 //生成[l,r]间的随机索引  
 int p=l+rnd.nextInt(r-l+1);  
 *swap*(arr,l,p);  
 //arr[l+1,j]<v arr[j+1,r]>v  
 int j=l;  
 for(int i=j+1;i<=r;i++){  
 if(arr[i].compareTo(arr[l])<0){  
 j++;  
 *swap*(arr,i,j);  
 }  
 }  
 *swap*(arr,j,l);  
 return j;  
 }  
  
 public static <E extends Comparable<E>> void sort2ways(E[] arr){  
  
 Random rnd=new Random();  
  
 *sort2ways*(arr,0,arr.length-1,rnd);  
 }  
  
 private static <E extends Comparable<E>> void sort2ways(E[] arr,int l,int r,Random rnd){  
  
 if(l>=r) return;  
  
 int p=*partition2*(arr,l,r,rnd);  
 *sort2ways*(arr,l,p-1,rnd);  
 *sort2ways*(arr,p+1,r,rnd);  
 }  
  
 private static <E extends Comparable<E>> int partition2(E[] arr,int l,int r,Random rnd){  
  
 //生成[l,r]间的随机索引  
 int p=l+rnd.nextInt(r-l+1);  
 *swap*(arr,l,p);  
  
 //arr[l+1,i-1]<=v arr[j+1,r]>=v  
 int i=l+1,j=r;  
 while(true){  
  
 while(i<=j && arr[i].compareTo(arr[l])<0)  
 i++;  
 while(j>=i && arr[j].compareTo(arr[l])>0)  
 j--;  
  
 if(i>=j) break;  
 *swap*(arr,i,j);  
 i++;  
 j--;  
 }  
 *swap*(arr,l,j);  
 return j;  
 }  
  
 public static <E extends Comparable<E>> void sort3ways(E[] arr){  
  
 Random rnd=new Random();  
  
 *sort3ways*(arr,0,arr.length-1,rnd);  
 }  
  
 private static <E extends Comparable<E>> void sort3ways(E[] arr,int l,int r,Random rnd){  
  
 if(l>=r) return;  
  
 //生成[l,r]间的随机索引  
 int p=l+rnd.nextInt(r-l+1);  
 *swap*(arr,l,p);  
  
 //arr[l+1,lt]<v arr[lt+1,i-1]==v arr[gt,r]>v  
 int lt=l,gt=r+1,i=l+1;  
  
 while(i<gt){  
 if(arr[i].compareTo(arr[l])<0) {  
 lt++;  
 *swap*(arr, i, lt);  
 i++;  
 }else if(arr[i].compareTo(arr[l])>0){  
 gt--;  
 *swap*(arr,i,gt+1);  
 }else{ //arr[i]==arr[l]  
 i++;  
 }  
 }  
 *swap*(arr,l,lt);  
 //arr[l,lt-1]<v arr[lt,gt-1]==v arr[gt,r]>v  
  
 *sort3ways*(arr, l, lt-1, rnd);  
 *sort3ways*(arr,gt,r,rnd);  
 }  
  
  
 public static <E extends Comparable<E>> void swap(E[] arr,int i,int j){  
  
 E temp=arr[i];  
 arr[i]=arr[j];  
 arr[j]=temp;  
 }  
  
  
 public static void main(String[] args){  
  
 int n=10000;  
  
 Integer[] arr=ArrayGenerator.*generateRandomArray*(n,n);  
 Integer[] arr2= Arrays.*copyOf*(arr,arr.length);  
 Integer[] arr3= Arrays.*copyOf*(arr,arr.length);  
  
 System.*out*.println("Random array");  
 sortingHelper.*sortest*("QuickSort",arr);  
 sortingHelper.*sortest*("QuickSort2ways",arr2);  
 sortingHelper.*sortest*("QuickSort3ways",arr3);  
 System.*out*.println();  
  
 arr=ArrayGenerator.*generateOrderedArray*(n);  
 arr2= Arrays.*copyOf*(arr,arr.length);  
 arr3= Arrays.*copyOf*(arr,arr.length);  
  
 System.*out*.println("Ordered array");  
 sortingHelper.*sortest*("QuickSort",arr);  
 sortingHelper.*sortest*("QuickSort2ways",arr2);  
 sortingHelper.*sortest*("QuickSort3ways",arr3);  
 System.*out*.println();  
  
 arr=ArrayGenerator.*generateRandomArray*(n,1);  
 arr2= Arrays.*copyOf*(arr,arr.length);  
 arr3= Arrays.*copyOf*(arr,arr.length);  
  
 System.*out*.println("Same value array");  
 sortingHelper.*sortest*("QuickSort",arr);  
 sortingHelper.*sortest*("QuickSort2ways",arr2);  
 sortingHelper.*sortest*("QuickSort3ways",arr3);  
 System.*out*.println();  
  
 }  
}

public class sort3ways {  
 public void sortColors(int[] nums) {  
  
 int l=-1,r=nums.length,i=0;  
  
 //arr[l+1,lt]<v arr[lt+1,i-1]==v arr[gt,r]>v  
  
 while(i<r){  
 if( nums[i]==0){  
 l++;  
 swap(nums,i,l);  
 i++;  
 }else if(nums[i]==2){  
 r--;  
 swap(nums,i,r);  
 }else{  
 i++;  
 }  
 }  
  
 }  
  
 public void swap(int[] arr,int i,int j){  
  
 int temp=arr[i];  
 arr[i]=arr[j];  
 arr[j]=temp;  
 }  
  
}