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\B4D6750DFD50B9CD8C8F49675532AB6E.jpg |
| 本周练习过的代码（例） |

import java.util.Arrays;  
  
public class MergeSort {  
  
 private MergeSort(){}  
  
  
 public static <E extends Comparable<E>> void sort(E[] arr){  
  
 E[] temp= Arrays.*copyOf*(arr,arr.length); //temp=arr[l,r+1);  
 *sort*(arr,0,arr.length-1,temp);  
 }  
  
 public static <E extends Comparable<E>> void sort(E[] arr,int l,int r,E[] temp){  
  
 if(l>=r)return;  
// if(r-l<=15){  
// insert.insertsort2(arr,l,r);  
// return;  
// }  
 int mid=l+(r-l)/2;  
 *sort*(arr,l,mid,temp);  
 *sort*(arr,mid+1,r,temp);  
 if(arr[mid].compareTo(arr[mid+1])>0)  
 *merge*(arr,l,mid,r,temp);  
 }  
  
 //自底向上归并  
 public static<E extends Comparable<E>> void sortBU(E[] arr){  
  
 E[] temp=Arrays.*copyOf*(arr,arr.length);  
 int n=arr.length;  
 //遍历合并的区间长度  
 for(int sz=1;sz<n;sz+=sz){  
 //合并[i,i+sz-1]与[i+sz,Math.main(i+sz+sz-1,n-1)]  
 for(int i=0;i+sz<n;i+=sz+sz)  
 if(arr[i+sz-1].compareTo(arr[i+sz])>0)  
 *merge*(arr,i,i+sz-1,Math.*min*(i+sz+sz-1,n-1),temp);  
 }  
 }  
 //合并两个有序区间，arr[l,mid]与arr[mid+1,r]  
 public static<E extends Comparable<E>> void merge(E[] arr,int l,int mid,int r,E[] temp){  
  
 // E[] temp= Arrays.copyOfRange(arr,l,r+1); //temp=arr[l,r+1);  
  
 System.*arraycopy*(arr,l,temp,l,r-l+1);  
 int i=l,j=mid+1;  
 for(int k=l;k<=r;k++){  
 if(i>mid){  
 arr[k]=temp[j]; j++;  
 } else if(j>r){  
 arr[k]=temp[i]; i++;  
 } else if(temp[i].compareTo(temp[j])<=0){  
 arr[k]=temp[i]; i++;  
 } else{  
 arr[k]=temp[j]; j++;  
 }  
 }  
 }  
  
 public static<E extends Comparable<E>> void insertsort2(E[] arr,int l,int r){  
 for(int i=l;i<r;i++) {  
 E t = arr[i];  
 int j;  
 for(j=i;j-1>=l&&t.compareTo(arr[j-1])<0;j--)  
 arr[j]=arr[j-1];  
 arr[j]=t;  
 }  
 }  
  
 public static void main(String[] args){  
  
 int n=10000;  
 Integer[] arr=ArrayGenerator.*generateRandomArray*(n,n);  
 Integer[] arr2=Arrays.*copyOf*(arr,arr.length);  
  
 sortingHelper.*sortest*("MergeSort",arr);  
  
 sortingHelper.*sortest*("MergeSort2",arr2);  
 }  
}

import java.util.Arrays;  
  
public class reversePairs {  
  
 private int res=0;  
 public int reversePairs(int[] nums){  
  
 int[] temp= new int[nums.length]; //temp=arr[l,r+1);  
  
 res=0;  
 sort(nums,0,nums.length-1,temp);  
 return res;  
 }  
  
 private void sort(int[] arr,int l,int r,int[] temp){  
  
 if(l>=r)return;  
  
 int mid=l+(r-l)/2;  
 sort(arr,l,mid,temp);  
 sort(arr,mid+1,r,temp);  
 if(arr[mid]>arr[mid+1])  
 merge(arr,l,mid,r,temp);  
 }  
  
 private void merge(int[] arr,int l,int mid,int r,int[] temp){  
  
 // E[] temp= Arrays.copyOfRange(arr,l,r+1); //temp=arr[l,r+1);  
  
 System.*arraycopy*(arr,l,temp,l,r-l+1);  
 int i=l,j=mid+1;  
 for(int k=l;k<=r;k++){  
 if(i>mid){  
 arr[k]=temp[j]; j++;  
 } else if(j>r){  
 arr[k]=temp[i]; i++;  
 } else if(temp[i]<=temp[j]){  
 arr[k]=temp[i]; i++;  
 } else{  
 res+=mid-i+1;  
 arr[k]=temp[j]; j++;  
 }  
 }  
 }  
  
}