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\A3A3A6C728C3F4C4571D22892B97855D.jpg |
| 本周练习过的代码（例） |

public class Solution012 {  
 public void sortColors(int[] nums){  
 int[] cnt=new int[3];  
 for(int num:nums){  
 cnt[num]++;  
 }  
  
 for(int i=0;i<cnt[0];i++){  
 nums[i]=0;  
 }  
 for(int i=cnt[0];i<cnt[0]+cnt[1];i++){  
 nums[i]=1;  
 }  
 for(int i=cnt[0]+cnt[1];i<cnt[0]+cnt[1]+cnt[2];i++){  
 nums[i]=2;  
 }  
 }  
}

public class reSolution {  
 public void sortColors(int[] nums){  
 //处理元素范围是[0,R)的计数排序  
 int R=3;  
 int[] cnt=new int[R];  
 for(int num:nums){  
 cnt[num]++;  
 }  
  
 int[] index=new int[R+1];  
 for(int i=0;i<R;i++)  
 index[i+1]=index[i]+cnt[i];  
  
 for(int i=0;i+1<index.length;i++)  
 for(int j=index[i];j<index[i+1];j++)  
 nums[j]=i;  
 }  
}

public class Student {  
 private String name;  
 private int score;  
  
 public Student (String name,int score){  
 this.name=name;  
 this.score=score;  
 }  
  
 public String getName(){  
 return name;  
 }  
 public int getScore(){  
 return score;  
 }  
 @Override  
 public String toString(){  
 return String.*format*("Student(name:%s,score:%d)",name,score);  
 }  
  
  
}

import java.util.Random;  
  
public class Main {  
 public static void main(String[] args) {  
 int n=26\*26\*26\*26;  
  
 Student[] students=new Student[n];  
 int k=0;  
 Random rnd=new Random();  
 for(char c1='a';c1<='z';c1++)  
 for(char c2='a';c2<='z';c2++)  
 for(char c3='a';c3<='z';c3++)  
 for(char c4='a';c4<='z';c4++){  
 students[k]=new Student(""+c1+c2+c3+c4,rnd.nextInt(101));  
 k++;  
 }  
  
 int R=101;  
//o(n)  
 int[] cnt=new int[R];  
 for(Student student:students)  
 cnt[student.getScore()]++;  
//o(R)  
 int[] index=new int[R+1];  
 for(int i=0;i<R;i++)  
 index[i+1]=index[i]+cnt[i];  
//o(n)  
 Student[] temp=new Student[n];  
 for(Student student:students){  
 temp[index[student.getScore()]]=student;  
 index[student.getScore()]++;  
 }  
//o(n)  
 for(int i=0;i<n;i++)  
 students[i]=temp[i];  
  
 //整体复杂度O(n+R)  
  
 for(int i=1;i<n;i++){  
 if(students[i-1].getScore()>students[i].getScore())  
 throw new IllegalArgumentException("Error");  
  
 if(students[i-1].getScore()==students[i].getScore()){  
  
 if(students[i-1].getName().compareTo(students[i].getName())>0)  
 throw new RuntimeException("Error");  
 }  
  
 }  
 }  
}