**原数组与稀疏数组之间的转换，存取磁盘之中**

**package** ex;

**import** java.io.BufferedReader;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

**import** java.io.FileOutputStream;

**import** java.io.FileReader;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.io.Reader;

**public** **class** ChessTransformation {

**public** **static** **void** main(String[] args) {

// 创建一个原始二维数组 11\*11

// 0 表示无子， 1 表示黑子 ， 2 表示白子

**int**[][] chessArr1 = **new** **int**[11][11];

chessArr1[1][2] = 1;

chessArr1[2][3] = 2;

chessArr1[5][6] = 2;

// 输出原始数组

System.***out***.println("输出的原始二维数组:");

**for** (**int**[] row : chessArr1) {

**for** (**int** data : row) {

System.***out***.printf("%d\t", data);

}

System.***out***.println();

}

// 将 原始二维数组 转换为 稀疏数组

**int** sum = 0;

**for** (**int** i = 0; i < 11; i++) {

**for** (**int** j = 0; j < 11; j++) {

**if** (chessArr1[i][j] != 0) {

sum++;

}

}

}

System.***out***.println("sum:" + sum);

// 创建对应的稀疏数组

**int**[][] sparseArr = **new** **int**[sum + 1][3];

// 给稀疏数组赋值

sparseArr[0][0] = 11;

sparseArr[0][1] = 11;

sparseArr[0][2] = sum;

// 遍历二维数组，将非零数组存放在稀疏数组中

**int** count = 0; // 用于记录是第几个非零数据

**for** (**int** i = 0; i < 11; i++) {

**for** (**int** j = 0; j < 11; j++) {

**if** (chessArr1[i][j] != 0) {

count++;

sparseArr[count][0] = i;

sparseArr[count][1] = j;

sparseArr[count][2] = chessArr1[i][j];

}

}

}

System.***out***.println();

// 输出稀疏数组

System.***out***.println("输出稀疏数组");

**for** (**int**[] row : sparseArr) {

**for** (**int** data : row) {

System.***out***.printf("%d\t", data);

}

System.***out***.println();

}

// 将稀疏数组存放在磁盘中

// 先指定需要输出的文件名称，同时也是具体地址

// 此为基础流

String pathname = "C:\\Users\\zhenghong\\" + "Desktop\\yuanyou\\experice/sparse2.txt";

File file = **new** File(pathname);

**try** {

// 创建以服务器为主体的文件输出流，此为功能流

FileWriter out = **new** FileWriter(file);

// 将稀疏数组中 每一个元素都输出到指定文件夹中

**for** (**int** i = 0; i < sparseArr.length; i++) {

**for** (**int** j = 0; j < 3; j++) {

out.write(sparseArr[i][j] + " ");

}

out.write("\n");

}

// 清除管道中的流

out.flush();

// 关闭管道

out.close();

} **catch** (IOException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

}

// 将稀疏数组从管道中读取出来

File file2 = **new** File(pathname);

**int**[][] sparseArr2 = **new** **int**[sum + 1][3];

**try** {

// FileReader reader = new FileReader(file2) ;

FileReader reader = **new** FileReader(file2);

BufferedReader in = **new** BufferedReader(reader);

String line; // 一行数据

**int** row = 0;

**while** ((line = in.readLine()) != **null**) {

String[] temp = line.split(" ");

System.***out***.println(temp.length);

**for** (**int** i = 0; i < temp.length; i++) {

sparseArr2[row][i] = Integer.*parseInt*(temp[i]);

}

row++;

}

reader.close();

in.close();

} **catch** (FileNotFoundException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

} **catch** (NumberFormatException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

} **catch** (IOException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

}

System.***out***.println("sparseArr2");

**for** (**int**[] row : sparseArr2) {

**for** (**int** data : row) {

System.***out***.print(data + " ");

}

System.***out***.println();

}

// now

// sparseArr2 为输出的稀疏数组

// 将该稀疏数组转换为原始数组

// 先读取稀疏数组的第一行，得到原始数组的具体情况

**int**[][] chessArr2 = **new** **int**[sparseArr2[0][1]][sparseArr2[0][1]];

// 输出恢复的原始数组，但未恢复至最以前

**for** (**int**[] row : chessArr2) {

**for** (**int** data : row) {

System.***out***.print(data + " ");

}

System.***out***.println();

}

**for** (**int** i = 1; i < sparseArr2.length; i++) {

chessArr2[sparseArr2[i][0]][sparseArr2[i][1]] = sparseArr2[i][2];

}

System.***out***.println();

**for** (**int**[] row : chessArr2) {

**for** (**int** data : row) {

System.***out***.print(data + " ");

}

System.***out***.println();

}

}

}