稀疏数组

*问题来源：一个二维数组很多默认值是0，因此记录了很多没有意义的数据

1.基本介绍

当一个数组大部分元素是0，或者为同一个值得数组，可以用稀疏数组来保存该数组

稀疏数组的处理方法:

1. 记录数组有几行几列，有多少个不同的值
2. 把不同值的元素的行列值记录在一个小规模数组，来实现压缩

2.稀疏数组组成:

	row	col	val
0	原始数组的行数	原始数组的列数	有效数字个数
1	第一个有效数字的行数	第一个有效数字的列数	第一个有效数字的值
2	第二个有效数字的行数	第二个有效数字的列数	第二个有效数字的值
....	....	...	....

3.二维数组与稀疏数组转化思路

1.二维数组->稀疏数组SparseArray

1. 获取二维数组的行列数
2. 遍历二维数组，获取有效值个数sum
3. 根据sum创建稀疏数组sparseArr int[sum+1][3]
4. 将有效值存入稀疏数组

2.稀疏数组SparseArray->二维数组

1. 根据稀疏数组第一列创建二维数组
2. 读取后几行数据，写入二维数组

3.实现代码

```
package 线性结构.SparseArray;
import java.util.ArrayList;
/**
 * 稀疏数组SparseArray和二维数组
 * @author jndeng
 * @create 2019-11-05 19:31
 */
public class SparseArray {
    public static void main(String[] args) {
        int[][] a = new int[11][11];
        a[1][2]=1;
        a[2][3]=2;
        int[][] array = arrayToSparse(a, 0);
        for (int i = 0; i < array.length; i++) {
            for (int j = 0; j < array[i].length; j++) {
                System.out.print(array[i][j] + " ");
            }
            System.out.println();
        }
        System.out.println("-----------------------");
        array=sparseToArray(array,0);

        for (int i = 0; i < array.length; i++) {
            for (int j = 0; j < array[i].length; j++) {
                System.out.print(array[i][j] + " ");
            }
            System.out.println();
        }
    }

    /**
     * 二维数组转化为稀疏数组
```

```
     *
     * @param array 二维数组
     * @param num   无效值
     * @return 稀疏数组
     */
    public static int[][] arrayToSparse(int[][] array, int num) {
        int row = array.length;
        int col = array[0].length;
        int count = 0;
        ArrayList<int[]> arrayList = new ArrayList<>();
        for (int i = 0; i < row; i++) {
            for (int j = 0; j < col; j++) {
                if (array[i][j] != num) {
                    count++;
                    arrayList.add(new int[]{i, j, array[i][j]});
                }
            }
        }
        int[][] sparseArray = new int[count + 1][3];
        int sparseRow = sparseArray.length;
        sparseArray[0][0] = row;
        sparseArray[0][1] = col;
        sparseArray[0][2] = count;
        for (int i = 1; i < sparseRow; i++) {
            for (int j = 0; j < 3; j++) {
                sparseArray[i][j] = arrayList.get(i - 1)[j];
            }
        }
        return sparseArray;
    }

    /**
     * 稀疏数组转化为二维数组
     * @param sparseArray 稀疏数组
     * @param num 无效值
     * @return 二维数组
     */
    public static int[][] sparseToArray(int[][] sparseArray, int num) {
        if (sparseArray[0].length != 3)
            return null;

        int row = sparseArray.length;
        int[][] array = new int[sparseArray[0][0]][sparseArray[0][1]];

        for (int i = 0; i < array.length; i++) {
            for (int j = 0; j < array[i].length; j++) {
                array[i][j] = num;
```

```
            }
        }
        for (int i = 1; i < row; i++) {
            int nrow = sparseArray[i][0];
            int ncol =sparseArray[i][1];
            int nval=sparseArray[i][2];
            array[nrow][ncol]=nval;
        }
        return array;


    }
}
```