# 稀疏数组转换

## 二维数组转稀疏数组的思路

1.遍历 原始的二维数组，得到有效数据的个数 sum

2.根据sum 就可以创建 稀疏数组 sparseArr int[sum + 1][3]

3.将二维数组的有效数据存入 稀疏数组

## 稀疏数组转二维数组的思路

1.先读取稀疏数组的第一行，根据第一行的数据，创建元素的二维数组

2.再读取稀疏数组的后几行数据，并赋值给原始的二维数组即可

/\*\*

\* @description 创建rwo行col列的数组

\* @param rwo:行、col：列、def：默认值

\* @return void

\* @status public

\*/

function createTwoArray(rwo = 1, col = 1, def = undefined) {

    // Array.from 函数如果有传第二个参数的时候， 新数组中的每个元素会执行该回调函数。

    // 使用 Array.prototype.fill 方法填充的值指向同一个引用地址。

    let twoArray = Array.from(new Array(rwo), () => new Array(col).fill(def));

    return twoArray;

}

/\*\*

\* @description 二维数组 转 稀疏数组的思路

\* @param twoArr:二维数组

\* @return void

\* @status public

\*/

function twoArrToSparseArr(twoArr) {

    // 将二维数组平铺成一维数组

    let flatArr = twoArr.flat();

    // 1.遍历 原始的二维数组，得到有效数据的个数 sum

    let sum = 0;

    for (let item of flatArr) {

        if (item != 0) {

            sum++;

        }

    }

    // 2.根据sum 就可以创建 稀疏数组 sparseArr int[sum + 1][3]

    let sparseArr = createTwoArray(sum + 1, 3);

    // 3.保存稀疏数组的行和列，以及有效数据个数

    sparseArr[0][0] = twoArr.length;

    sparseArr[0][1] = twoArr[0].length;

    sparseArr[0][2] = sum;

    // 4.遍历二维数组，将二维数组的有效数据存入 稀疏数组

    let count = 0;// 用于记录是第几个有效数据

    for (let i = 0; i < sparseArr[0][0]; i++) {

        for (let j = 0; j < sparseArr[0][1]; j++) {

            if (twoArr[i][j] != 0) {

                count++;

                sparseArr[count][0] = i;

                sparseArr[count][1] = j;

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

            }

        }

    }

    return sparseArr;

}

/\*\*

\* @description 稀疏数组 转 二维数组的思路

\* @param sparseArr:二维数组

\* @return void

\* @status public

\*/

function sparseArrToTwoArr(sparseArr) {

    // 1.先读取稀疏数组的第一行，根据第一行的数据，创建元素的二维数组

    let chessArr2 = createTwoArray(sparseArr[0][0], sparseArr[0][1], 0);

    // 2.再读取稀疏数组的后几行数据，并赋值给原始的二维数组即可

    for(let i = 1;i < sparseArr.length;i++){

        let row = sparseArr[i][0];

        let col = sparseArr[i][1];

        let value = sparseArr[i][2];

        chessArr2[row][col] = value;

    }

    return chessArr2;

}

// 创建一个原始的二维数组 11 \* 11

// 0: 表示没有棋子，1表示黑子，2表示蓝子

let chessArr1 = createTwoArray(11, 11, 0);

chessArr1[1][2] = 1;

chessArr1[2][4] = 1;

chessArr1[4][5] = 1;

let sparseArr = twoArrToSparseArr(chessArr1);

console.log(sparseArr);

let chessArr2 = sparseArrToTwoArr(sparseArr);

console.log(chessArr2);