```
● CSDN 博客 下载 学习 社区 GitCode 云服务 猿如意
                版权
                分类专栏: 华为OD机试2023 (JAVA) 文章标签: java 算法 华为
             F500-kletto 华为OD机试2023 (... 专栏收录该内容
                                                                       已订阅
                                                      750 订阅 216 篇文章
                                                                                 超级会员免费看
             M企作 (7) 该专栏为热销专栏榜 第1名
             华为OD机试真题,2023年度机试题库全覆盖,刷题指南点<u>这里</u>
             找数字
             知识点 哈希表<sup>Q</sup>
              时间限制: 1s 空间限制: 256MB 限定语言: 不限
             题目描述:
             给一个二维数组nums,对于每一个元素num[i],找出距离最近的且值相等的元素,输出横纵坐标差值
             的绝对值之和,如果没有等值元素,则输出-1。
             例如:
             输入数组nums为
             03542
             25783
             25424
             对于 num[0][0] = 0, 不存在相等的值。
             对于 num[0][1] = 3, 存在一个相等的值, 最近的坐标为num[1][4], 最小距离为4。
             对于 num[0][2] = 5, 存在两个相等的值, 最近的坐标为num[1][1], 故最小距离为2。
             对于 num[1][1] = 5, 存在两个相等的值, 最近的坐标为num[2][1], 故最小距离为1。
             故输出为
             -14 2 3 3
              11-1-14
              11 2 3 2
             输入描述:
             输入第一行为二维数组的行
             输入第二行为二维数组的列
             输入的数字以空格隔开。
             输出描述:
             数组形式返回所有坐标值。
             补充说明:
             1. 针对数组num[i][j],满足0<i<=100; 0<j<=100。
             2. 对于每个数字,最多存在100个与其相等的数字。
             示例1
             输入:
             3
             03542
             25783
             25424
             输出:
             [[-1, 4, 2, 3, 3], [1, 1, -1, -1, 4], [1, 1, 2, 3, 2]]
             解题思路:
               1. 将二维数组中的值和坐标转化为map对象。
               2. 遍历二维数组中的值,找出其值的所有坐标,求出其中距离最短的值
             代码:
                  public class Main{
                2
                     public static void main(String[] args) {
                3
                4
                5
                         Scanner sc = new Scanner(System.in);
                         int n = sc.nextInt();
                6
                         int m = sc.nextInt();
                8
                         int[][] nums = new int[n][m];
                9
               10
               11
                         /**
                         * 将数字和坐标转化为map
               12
               13
                         * key: nums数组中的各值
                          * value: nums数组中的坐标的集合
               14
               15
                         */
                         Map<Integer, List<int[]>> map = new HashMap<>();
               16
                         for(int i=0; i<n; i++){
               17
                            for(int j=0; j<m; j++){
               18
                               int input = sc.nextInt();
               19
                               nums[i][j] = input;
               20
               21
                               int[] ints1 = new int[]{i,j}; //将坐标转化为数组
                               List<int[]> tempList;
               22
                               if(map.containsKey(input)){
               23
                                   tempList = map.get(input);
               24
                               }else {
               25
                                   tempList = new ArrayList<>();
               26
               27
               28
                               tempList.add(ints1);
               29
                               map.put( input, tempList);
               30
               31
                         List<List<Integer>> resList = new ArrayList<>();
               33
                         for(int i=0; i<n; i++){
               34
                            List<Integer> list = new ArrayList<>();
               35
               36
                            for(int j=0; j<m; j++){
                               int num = nums[i][j];
               37
                               List<int[]> listInts = map.get(num);
               38
                               if(listInts.size() == 1){ //只有一个坐标,则不存在相等的值
               39
                                  list.add(-1);
               40
               41
                                   continue;
                               }
               42
               43
                               int min = Integer.MAX_VALUE;
                                                         //最小距离
               44
                               for(int k=0; k<listInts.size(); k++){</pre>
               45
                                   int[] intnum = listInts.get(k);
               46
                                   int distance = Math.abs(intnum[0]-i) + Math.abs(intnum[1]-j); //求出
               47
                                   if(distance == 0){ //距离为0则跳过
               48
                                      continue;
               49
               50
                                   min = Math.min(min, distance);
               51
               52
               53
                               list.add(min);
               54
               55
                            resList.add(list);
               56
               57
               58
               59
                         System.out.println(resList);
               60
               61
               62
               63 }
             满分答案:
                1 import java.util.Arrays;
                  import java.util.Scanner;
                  public class Main {
                4
                     public static void main(String[] args) {
                5
                         Scanner in = new Scanner(System.in);
                6
                7
                         int m = in.nextInt();
                         int n = in.nextInt();
                8
                         int[][] map = new int[m][n];
                9
               10
                         for (int i = 0; i < m; i++) {
               11
                            for (int j = 0; j < n; j++) {
               12
                               map[i][j] = in.nextInt();
               13
               14
               15
                         }
               16
                         int[][] ans = new int[m][n];
               17
                         for (int i = 0; i < m; i++) {
               18
                            for (int j = 0; j < n; j++) {
               19
                               ans[i][j] = find(map, i, j);
               20
               21
                         }
               22
               23
                         System.out.println(Arrays.deepToString(ans));
               24
               25
               26
               27
               28
                     private static int find(int[][] map, int i, int j) {
               29
                         int target = map[i][j];
               30
                         int ans = Integer.MAX_VALUE;
               31
               32
                         for (int k = 0; k < map.length; k++) {
               33
                            for (int l = 0; l < map[0].length; l++) {</pre>
               34
                               if (k == i \&\& l == j) {
               35
                                   continue;
               36
               37
                               if (map[k][l] == target) {
               38
               39
                                   ans = Math.min(ans, Math.abs(i - k) + Math.abs(j - l));
               40
               41
               42
               43
               44
                         return ans == Integer.MAX_VALUE ? -1 : ans;
               45
               46
               47 }
            兆 文章知识点与官方知识档案匹配,可进一步学习相关知识
            算法技能树 > 首页 > 概览 40101 人正在系统学习中
               若博豆(已关注)
                                                                                         已订阅
                                                                               专栏目录
            【华为OD机试真题 python】找数字|找等值元素【2022 Q4 | 100分】
                                                                                <mark>华为OD机试</mark> Python OD笔试 <mark>找数字</mark>:给一个二维数组nums,对于每一个元素num[i],<mark>找</mark>出距离最近的且值相等的元素,输出横纵坐标...
           华为OD机试真题.pdf
                                                                                           04-15
           华为机试真题(非牛客网试练题)OD考试真题,不定期更新,文档含代码解答
           评论 2条>
                                                                                        写评论
                           这题也可以直接遍历二维数组 每一行的距离放在1个list 每一行的list再放在新的list中输出
                            1 for (int i = 0; i < hang; i++) {
                                        ArrayList<Integer> list1=new ArrayList<>();//每一行的距离
                                        for (int j = 0; j < lie; j++) {
                                           int tem=ints[i][j];
                                           int min_cha=Integer.MAX_VALUE;
                                           int count=0;
                                           //遍历找相等值
                                           for (int k = 0; k < hang; k++) {
                                              for (int l = 0; l < lie; l++) {
                                                  if (ints[k][l]==tem&&(k!=i||l!=j)){//注意不包括自身
                           10
                           11
                                                     count++;
                                                     int cha=Math.abs(k-i)+Math.abs(l-j);
                           12
               hyzhang123 热评 13
                                                     if (cha<min_cha){</pre>
                                                        min_cha=cha;
                           14
                           16
                           17
                                           }
                           18
                           19
                                           //輸出距离
                           20
                                           if (count>0){
                           21
                                              list1.add(min_cha);
                           22
                           23
                                           }else {
                           24
                                              list1.add(-1);
                           25
                                           }
                           26
            【华为OD机试真题 JAVA】 找车位【2022 Q4 | 100分】
                                                                               华为OD机试 Java OD笔试 找车位:停车场有一横排车位,0代表没有停车,1代表有车。
            【2023】华为OD机试真题Java-题目0197-找数字
                                                                              给一个二维数组nums,对于每一个元素num[i],<mark>找</mark>出距离最近的且值相等的元素,输出横纵坐标差值的绝对值之和,如果没有等值元素...
           人才招聘系统
           两江人力招聘系统网
            【华为OD机试真题 java、python】找等值元素(100%通过+复盘思路)
                                                                                   hjc的博客 0 682
           给一个二维数组nums,对于每一个元素num[i],<mark>找</mark>出距离最近的且值相等的元素,输出横纵坐标差值的绝对值之和,如果没有等值元素....
           C++算法之 二维数组的查找
                                                                                     Coder () 2968
           题目:在一个二维数组当中,每一行都按照从左到右递增的顺序排序,每一列都按照从上到下递增的顺序排序。请完成一个函数,输入...
            【满分】【华为OD机试真题2023 JAVA】查找重复代码
                                                                            查<mark>找</mark>重复代码时间限制:1s空间限制:32MB限定语言:不限题目描述:小明负责维护项目下的代码,需要查<mark>找</mark>出重复代码,用以支撑后...
            【华为机试真题详解】检查是否存在满足条件的数字组合
                                                                               不太灵光的程序员 ① 784
            《华为机试真题详解 Python实现》专栏含牛客网华为专栏、华为面经试题、华为OD机试真题。如果您在准备华为的面试,期间有想了...
            【新2023】华为OD机试 - 矩阵的最大值(Python)
                                                         梦想橡皮擦,专栏100例写作模式先行者,现象级专... ① 1634
           找数字(Python) 题目 给定一个仅包含0和1的n*n二维矩阵 请计算二维矩阵的最大值 计算规则如下 每行元素按下标顺序组成一个二进...
                                                                           weixin_43346653的博客 ② 299
           LeetCode 面试题04. 二维数组中的查找 (Python)
           题目注意: 本题与主站 240 题相同: https://leetcode-cn.com/problems/search-a-2d-matrix-ii/ 来源: 力扣(LeetCode) 链接: https://le...
                                                                           leetcode--python--1365
           1365. 有多少小于当前<mark>数字的数字</mark> 给你一个数组 nums,对于其中每个元素 nums[i],请你统计数组中比它小的所有<mark>数字</mark>的数目。 换而言....
           华为OD机试真题2023(JAVA)
                                                                             华为机试题库已由2022版换为2023版。本栏准备90+道算法真题。
           机试题目一
                                                                               1、<mark>找</mark>朋友 package Q2; import java.util.Arrays; import java.util.Scanner; /** * <mark>找</mark>朋友 * 在学校中,N个小朋友站成一排,第i个小朋友的身...
           华为OD机试(2022&2023)专栏介绍 + 真题目录 热门推荐
                                                                              华为OD机试真题(Java & JavaScript & Python)
           二维数组找数字
                                                                               题目描述: 在一个二维数组中(每个一维数组的长度相同),每一行都按照从左到右递增的顺序排序,每一列都按照从上到下递增的顺序...
           华为机试——找出数组中不同的两个数字
                                                                            weixin_44758107的博客 ① 39
            (进阶)已知数组中<mark>数字</mark>都两两相同,只有两个不同,<mark>找</mark>出这两个<mark>数字</mark> 这里也提供三种方法,前两种和之前相似,第三种需要熟练异或...
           LeetCode刷题-1365.有多少小于当前数字的数字(给你一个数组 nums,对于其中每个元素 nu... AnswerCoder的博客 💿 932
           LeetCode每日一题系列-1365.有多少小于当前数字的数字补Leetcode 10月26号的每日一题一、题目描述:给你一个数组 nums,对于...
                                                                                                    Beta
                                                                                                   二维数组的查找 (根据给定的数查找相等的元素)
                                                                            完成一个函数,输入这样的一个二维数组和一个整数,判断数组中是否含有该整数。
                                                                                                    Beta
            【项目实战】基于SpringCloud Gateway的webflux与Google的kaptcha实现图形验证码功能 最新发布 本本本添哥 ② 255
           基于SpringCloud 微服务架构的项目中,需要使用图形验证码,以下是使用的记录。
                                                                                                   "相关推荐"对你有帮助么?
                                                                                                   6)
                              非常没帮助
                                                         ⇒ 有帮助
                                                                   非常有帮助
                                                                                                   举报
                  关于我们 招贤纳士 商务合作 寻求报道 ☎ 400-660-0108 ☑ kefu@csdn.net ⑤ 在线客服 工作时间 8:30-22:00
                 公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心
                  家长监护 网络110报警服务 中国互联网举报中心 Chrome商店下载 账号管理规范 版权与免责声明 版权申诉 出版物许可证 营业执照
```

©1999-2023北京创新乐知网络技术有限公司

① 发布