

赖凯庭-华为实验班作业（20200923）

一、代码实践

百练oj-滑雪问题

```
import java.util.*;

public class Main{

    private static int[][] arr; //存放输入的矩阵
    private static int[][] value; //对应arr没点的“滑雪长度”
    private static int c; //arr的列数
    private static int r; //arr的行数

    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);
        while(sc.hasNext()){
            r = sc.nextInt();
            c = sc.nextInt();
            int max = 0;
            arr = new int[r][c];
            value = new int[r][c];
            for (int i = 0; i < r; i++) {
                for (int j = 0; j < c; j++) {
                    arr[i][j] = sc.nextInt();
                }
            }
            for (int i = 0; i < r; i++) {
                for (int j = 0; j < c; j++) {
                    int temp = ski(i, j, Integer.MAX_VALUE);
                    if (temp > max) {
                        max = temp;
                    }
                }
            }
            System.out.println(max);
        }
        sc.close();
    }

    private static int ski(int row, int col, int maxValue) {
        //出了矩阵的区域或者现在arr这个点的值大于上一个arr点的值，即跑到更高的点去了，无效，返回0
        if (col >= c || col < 0 || row >= r || row < 0 || maxValue <= arr[row][col]) {
            return 0;
        }
        //若已经计算过了此arr点的“滑雪长度”，直接返回
        if (value[row][col] > 0) {
            return value[row][col];
        }
    }
}
```

```

        //不然就计算此点的“滑雪长度” = 上下左右的点的“滑雪长度”的最大值 + 1
        value[row][col] = max(ski(row-1, col, arr[row][col]), ski(row+1, col,
arr[row][col]), ski(row, col-1, arr[row][col]), ski(row, col+1, arr[row][col])) +
1;

        return value[row][col];
    }

    private static int max(int ski1, int ski2, int ski3, int ski4) {
        return Math.max(Math.max(ski1, ski2), Math.max(ski3, ski4));
    }
}

```

二、计算机基础知识

1、L1范数与L2范数（《统计学习分析》）

L0范数是指向量中非0的元素的个数。（L0范数很难优化求解）

L1范数是指向量中各个元素绝对值之和

L2范数是指向量各元素的平方和然后求平方根

L1范数可以进行特征选择，即让特征的系数变为0。

L2范数可以防止过拟合，提升模型的泛化能力，有助于处理 **condition number**不好下的矩阵（数据变化很小矩阵求解后结果变化很大）

2、数据库的三级模式和两级映像？

三级模式：内模式、模式、外模式 两级映射：外模式-模式映射、模式-内模式映射

为什么要有三种模式和两级映射：有这三种模式能有效地组织、管理数据，提高数据库的逻辑独立性和物理独立性，用户级对应外模式，概念级对应模式，物理级对应内模式。

逻辑独立性：当模式发生改变，只要改变外模式-模式之间的映射关系，就可以保持外模式不变，对应的应用程序也可以不变。

物理独立性：通过模式-内模式的映射，定义建立数据的逻辑结构和物理结构的映射的对应关系，当存储结构发生变化时，只需改变模式-内模式映射，就能保持模式不变，因此应用程序也可以保持不变。

内模式：

内模式也称为存储模式，是数据物理结构和存储方式的描述，是数据在数据库内部的表示方式。需要定义所以的内部记录类型、索引和文件的组织方式，以及数据控制方面的细节。

例如，记录的存储方式是顺序存储、B树结构存储还是Hash方法存储；索引按照什么方式组织；数据是否压缩存储，是否加密；数据的存储记录结构有何规定

模式：又称为概念模式或逻辑模式，它是由数据库设计者综合所有的用户数据，按照统一的观点构造全局逻辑结构，是对全部数据的逻辑结构和特征的总体描述。它由数据模定义语言DDL来描述、定义、体现和反映数据库系统的整体观。概念模式不仅要描述概念记录类型，还要描述记录间的联系、操作、数据的完整性和安全性等要求。但是，概念模式不涉及存储结构、访问技术等细节。只有这样，概念模式才算做到了“物理数据独立性”。

外模式：外模式也称用户模式或子模式，是用户与数据库系统的接口，是用户用到的那部分数据的描述。它由若干个外部记录类型组成。用户使用数据操纵语言对数据库进行操作，实际上是对外模式的外部记录进行操作。

总之，数据按外模式的描述提供给用户；按内模式的描述存储在磁盘上；而模式提供了连接这两级模式的相对稳定的中间层，并使得两级中任意一级的改变都不受另一级的牵制。

三、小组项目进展

- (1) 向leetcode刷题的子项目中提交了一次PR，与小组成员分享题解分享
- (2) 优化了项目的issue，添加了CONDUCT及issue模板文件