**深 圳 大 学 实 验 报 告**

**课程名称： 算法设计**

**实验名称： 动态规划—鸡蛋掉落问题**

**学院：计算机与软件学院 专业： 计算机科学与技术（创新班）**

**报告人： 何泽锋 学号： 2022150221 班级： 高性能特色班**

**指导教师： 杨烜**

**实验时间： 2024年5月10日-2024年5月17日**

**实验报告提交时间： 2024年5月18日**

**教务处制**

**一．实验目的**

1.掌握动态规划算法设计思想。

2.掌握鸡蛋坠落问题的动态规划解法

**二．实验要求**

1、给出鸡蛋掉落问题的动态规划方程；

2、随机产生f，e的值，对小数据模型利用蛮力法测试算法的正确性；

3、随机产生f ，e的值，对不同数据规模测试算法效率，并与理论效率进行比对，请提供能处理的数据最大规模，注意要在有限时间内处理完；

4、该算法是否有效率提高的空间？包括空间效率和时间效率。

**三．算法原理**

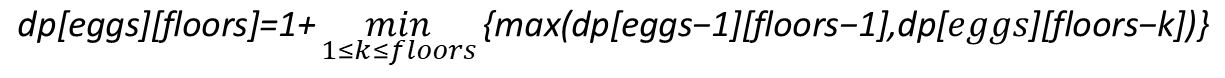
**1.动态规划方程**  
 **I.基本原理1：**

（1）在某层楼丢鸡蛋存在两种状态：

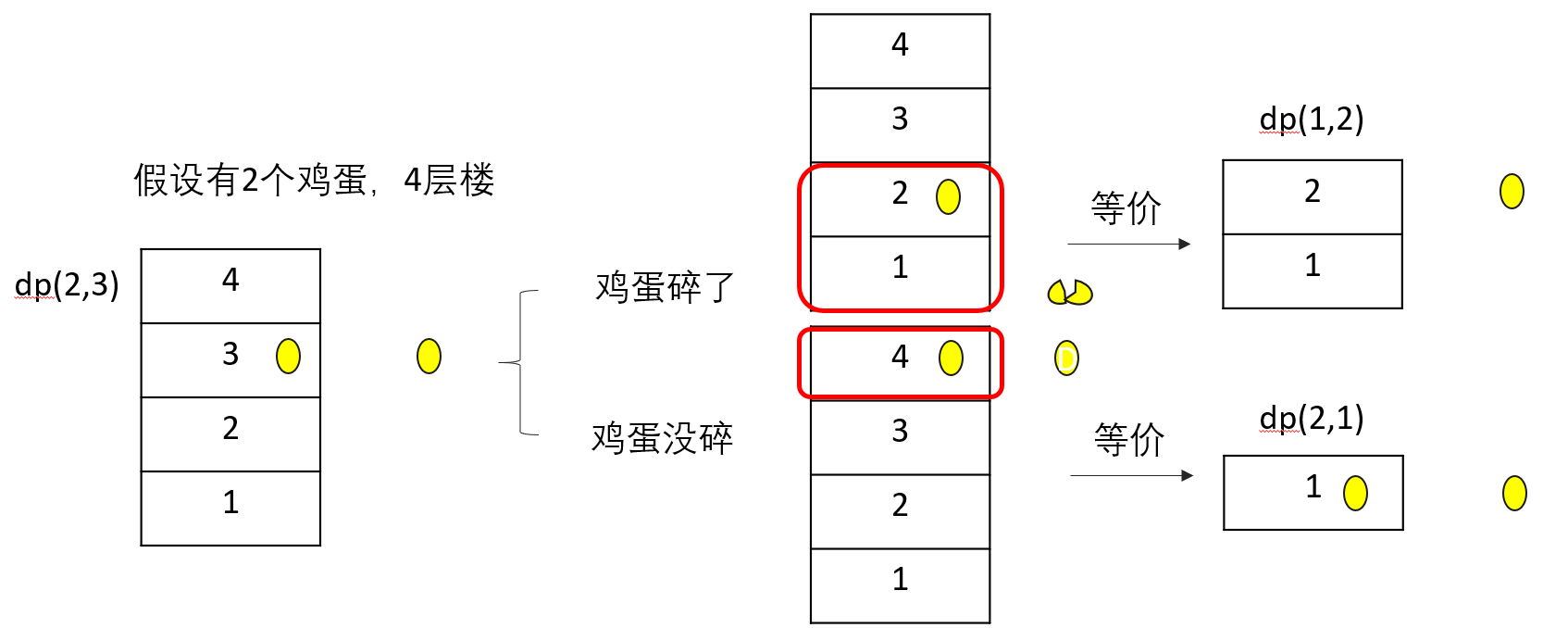
①鸡蛋碎了，此时手上鸡蛋数量减一，需要继续验证该层楼之下的楼层

②鸡蛋没碎，此时手上鸡蛋数量不变，需要验证该层楼之上的楼层

（2）根据以上两种状态可以写出状态转移方程：



（3）图形表示状态：



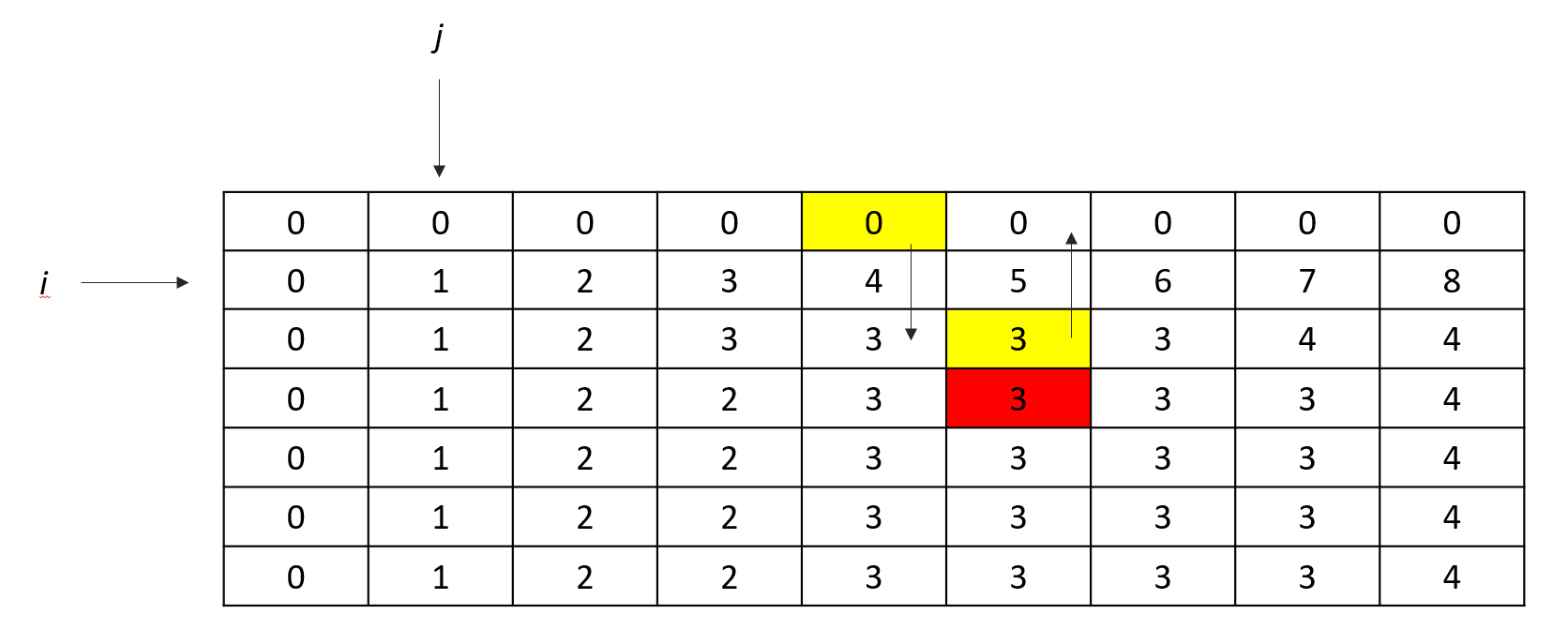
（4）状态转移表，其中行是鸡蛋数量，列是楼层数

边界条件：①当鸡蛋数为0，无论几层楼都只能操作0次

②当楼层数为0，无论几个鸡蛋都只能操作0次

③当鸡蛋数为1，有几层楼就要操作几次

如图所示，红色点状态由前一列和本列的状态推出



**II.基本原理2：**

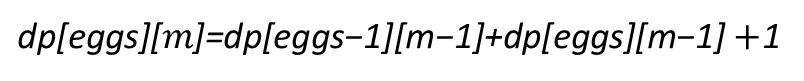
（1）已知有鸡蛋个数和操作次数，求最多能测多少层楼的临界点：

①鸡蛋碎了，此时手上鸡蛋数量减一，还可进行丢鸡蛋的操作次数减一，向下确定的范围

②鸡蛋没碎，此时手上鸡蛋数量不变，还可进行丢鸡蛋的操作次数减一，也就是说向上确定的范围

③加上当前层的总共，可以通过鸡蛋数量和丢鸡蛋的次数确定查找的层数范围

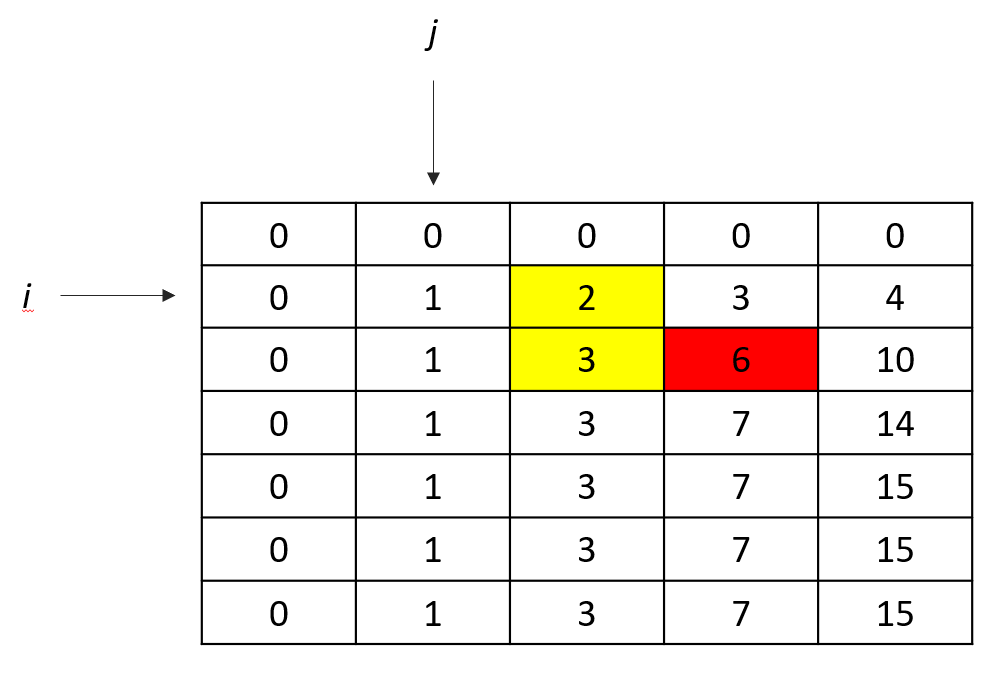
（2）根据以上两种状态可以写出状态转移方程



（3）状态转移表，其中行是鸡蛋数，列是操作次数

边界条件： ①当鸡蛋数为0，无论操作多少次，都只能确定0层

②当操作次数为0，无论有多少个鸡蛋都只能确定0层



**2.蛮力法验证动态规划正确性**

此处蛮力法使用的是暴力递归的方式，不会记录层间信息，每次都会重新搜查。在小规模数据（0，20）范围内使用同样的鸡蛋和楼层数测试递归法和dp结果是否相等，截取部分数据如下表所示



可以看到此处二者操作次数相同，根据数学归纳法可以验证动态规划方程的正确性

**3.复杂度分析**

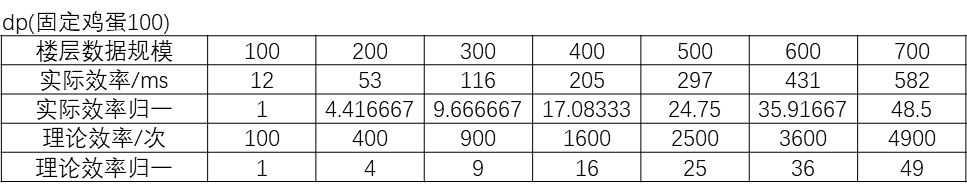
（1）基本dp的复杂度分析：

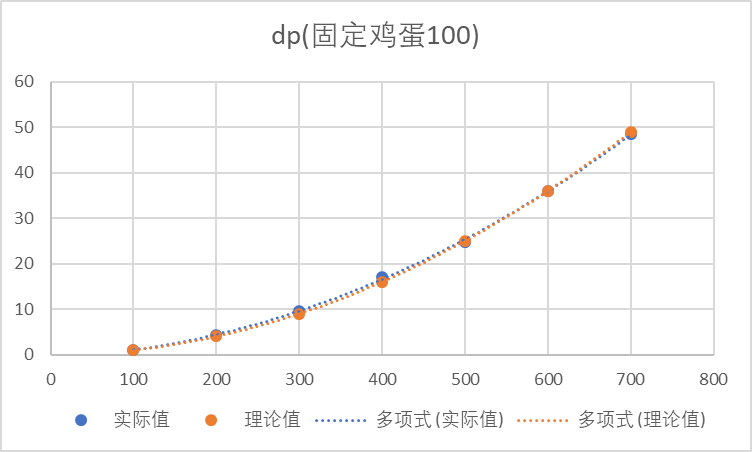
时间复杂度：使用了三重循环，分别遍历鸡蛋数、楼层数和最优解，可以计算得到其时间复杂度为

空间复杂度：使用了二维数组，第一维是鸡蛋数量，第二维是楼层数，因此空间复杂度为

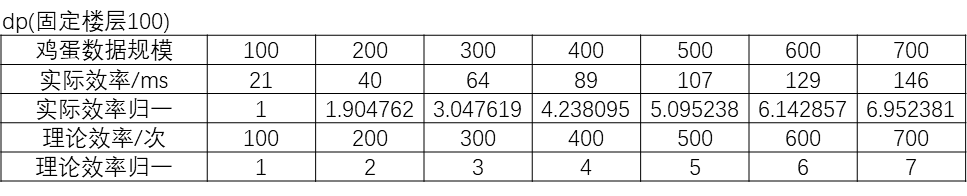
（2）数据验证基本dp时间复杂度

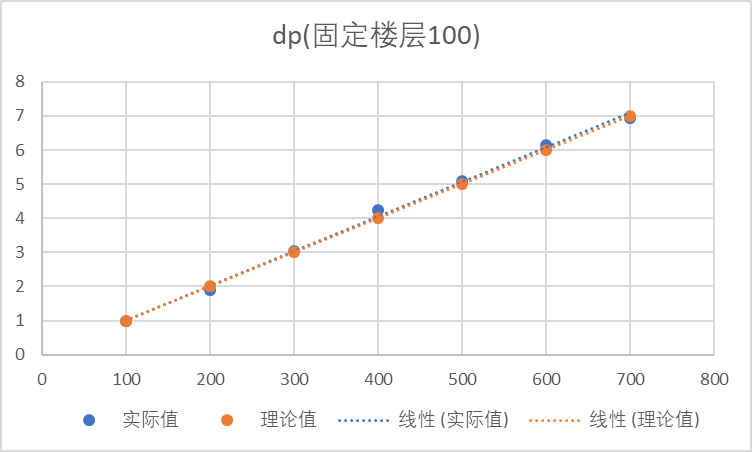
①固定鸡蛋数量为100，逐渐增加楼层数，根据时间复杂度，可以估计是以增长，具体数据如下表所示：





②固定楼层数为100，逐渐增加鸡蛋数，根据时间复杂度，可以估计是以线性增长，具体数据如下表所示：



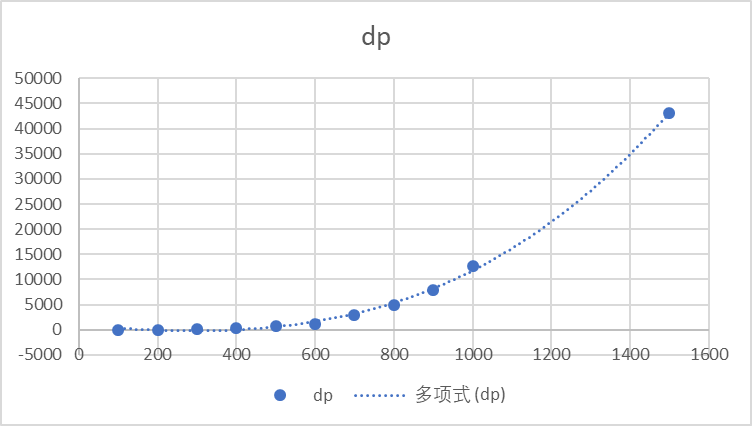


（3）测试基本dp能处理的最大规模数据

①设置k和n为同样大的数量级，随机生成该范围内数据，当5分钟内无法测试出数据即视为最大处理规模，此处为部分测试数据，测试结果最大为3000



②可以看到，当k和n取值非常接近时，时间复杂度接近



③在leetcode上运行测试后发现TLE



**4.效率优化**

**I.基本dp的空间复杂度优化**

优化原理：原始dp的时间复杂度和空间复杂度分别为和𝑂(𝑘𝑛)，通过使用两个数组来替代二维数组储存dp值，两个数组分别为dp和dpPrev，分别存储当前楼层操作次数和前一楼层操作次数，因此空间复杂度为𝑂(𝑛)

**II.使用二分法优化基本dp的时间复杂度**

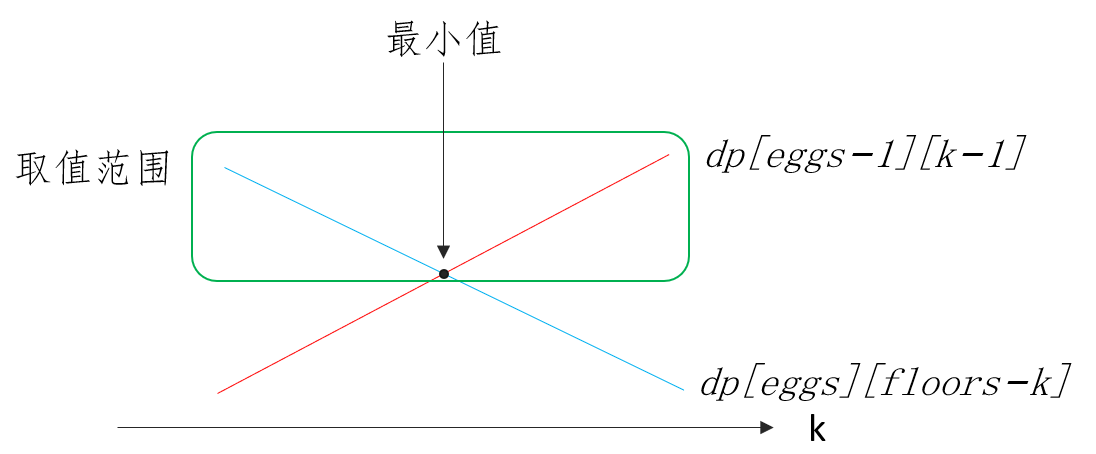
（1）优化原理：根据状态转移方程

可以分析得到两个相关函数：

① :当鸡蛋数量确定时，随着k增加楼层数增加，可以得知需要操作的次数逐渐增加，因此单调递增

② :当鸡蛋数量确定时，随着k增加楼层数减少，可以得知需要操作的次数逐渐减少增加，因此单调递增

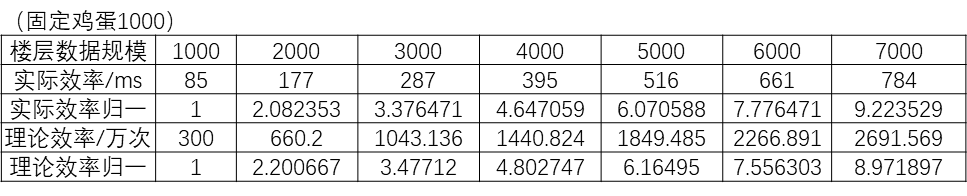
做成图像如下所示：

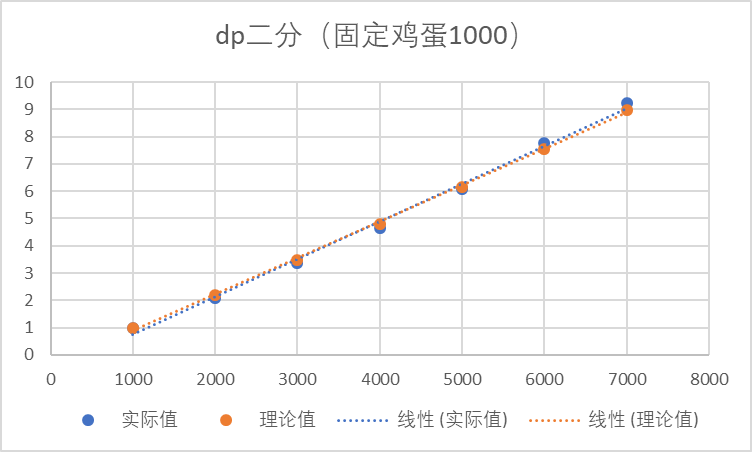


因此，最优解可以通过使用二分的方式寻找最小值，使得寻找的时间从优化到因此完整时间复杂度为

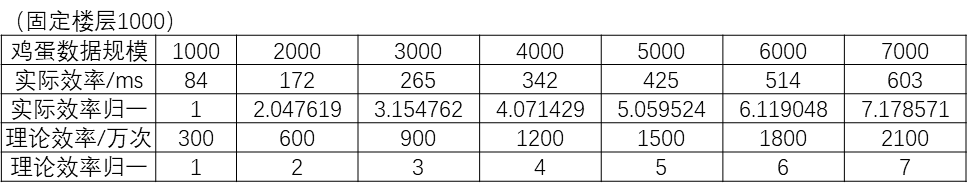
（2）数据验证二分dp时间复杂度：

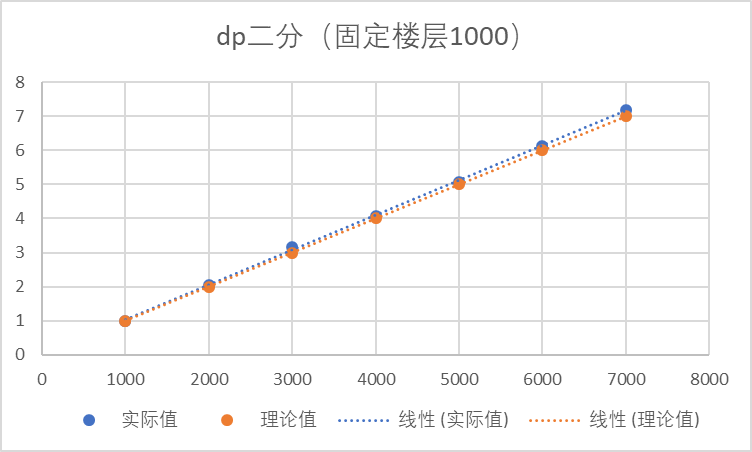
①固定鸡蛋数量为1000，逐渐增加楼层数，根据时间复杂度，可以估计是以增长，具体数据如下表所示：





②固定楼层数为1000，逐渐增加鸡蛋数，根据时间复杂度，可以估计是以线性增长，具体数据如下表所示：

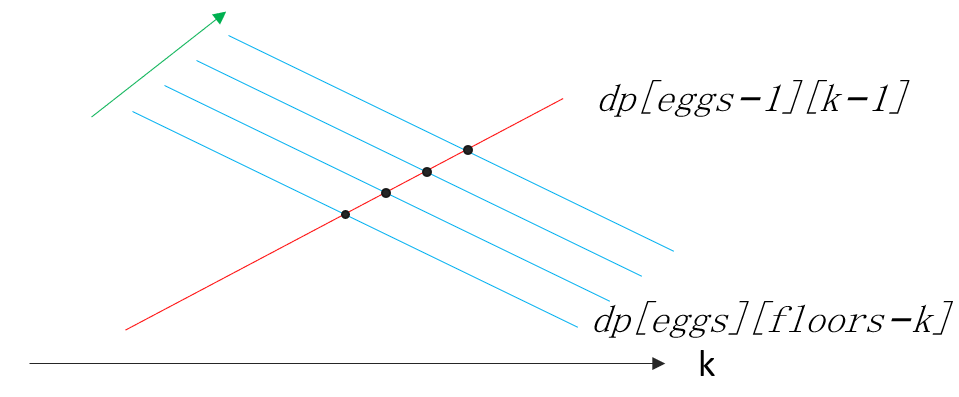




（3）最大数据处理规模：120000

**III. 决策单调性优化dp**

（1）优化原理：该方法是基于二分法继续优化得到的，首先令Xa是能够找最小移动次数的最小的X，根据上一个方法中我们分析单调性的方法，我们可以再次分析，并可以最终得出是随着floors的增加而增加的，图表如下所示：

****

（2）效率分析：

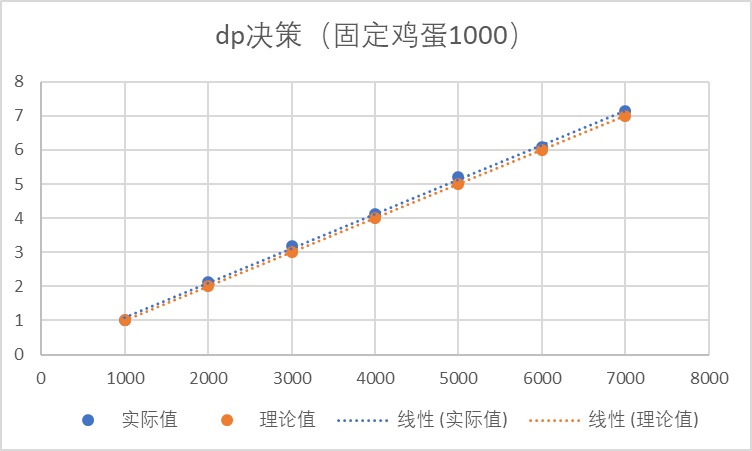
①时间复杂度：需要计算*O(kn)*个状态，同时对于每个k，最优解指针只会遍历0 到 n，复杂度也是*O(kn)*。因此总体时间复杂度为*O(kn)*

②空间复杂度：*O(n)*

（3）数据验证复杂度：

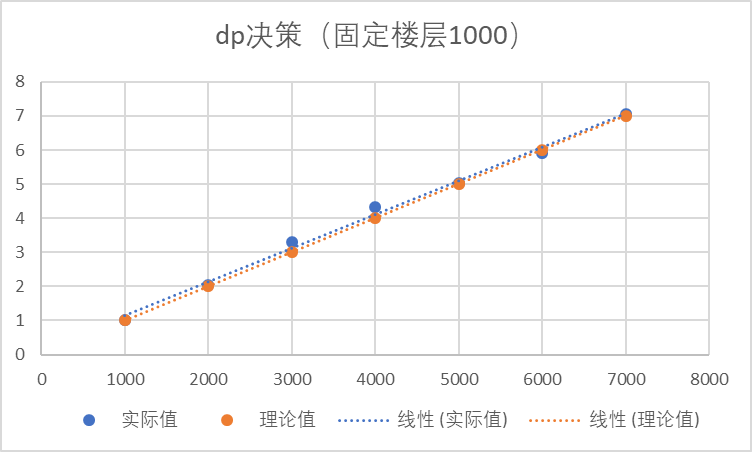
①固定鸡蛋数量为1000，逐渐增加楼层数，根据时间复杂度*O(kn)*，可以估计是线性增长，具体数据如下表所示：





②固定鸡蛋数量为1000，逐渐增加楼层数，根据时间复杂度*O(kn)*，可以估计是线性增长，具体数据如下表所示：





（4）最大数据处理规模：40000

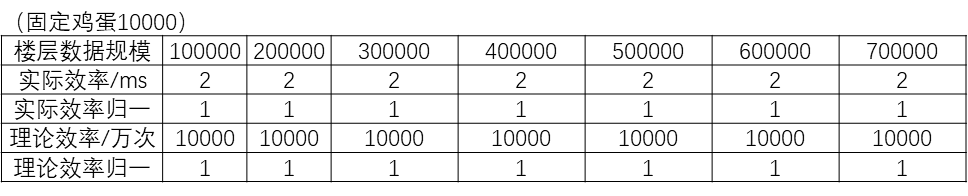
**V**

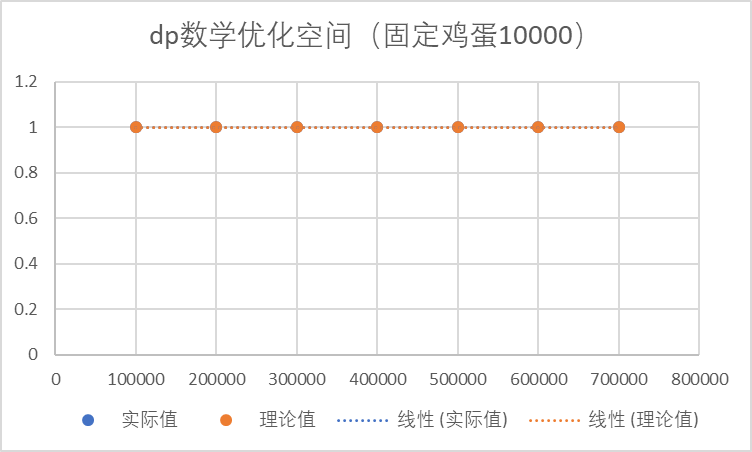
**V.优化空间的逆向dp**

（1）优化原理：与第一种dp优化相似，每轮的dp只与上一轮的dp有关，因此可以优化使用一个数组，因此其空间复杂度为

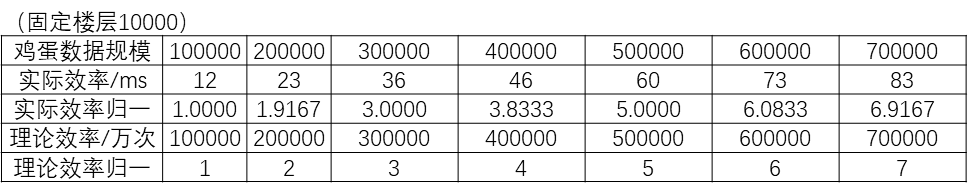
（2）数据验证：

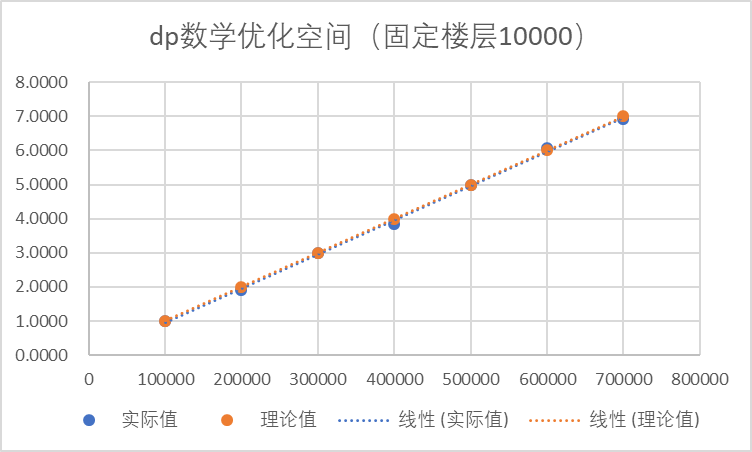
①固定鸡蛋数量为10000，逐渐增加楼层数，根据时间复杂度，因为k次根号接近1，因此估计接近常函数，具体数据如下表所示：





②固定楼层数为10000，逐渐增加鸡蛋数，根据时间复杂度，可以估计是以线性增长，具体数据如下表所示：





（3）最大数据处理规模：5000000

**四.实验结论**

通过本次实验了解了动态规划的使用，本质上动态规划也是搜索，但是dp通过记录之前的结果，并通过状态转移方程得到当前结果，可以减少搜索的深度，大大优化了时间。建立状态转移表的过程也可以使用不同的方法，通过不同方式优化搜索的时间。同时，可以根据状态转移方程使用的数据进行空间复杂度优化。经过数据检验可以知道，测试一个算法的数据处理范围不仅受到时间复杂度影响，空间复杂度同样也会影响。在使用二维数组时两个维度分别存放什么数据也会影响效率，这是因为在地址空间查找时，若第二个维度较大，探查时需要进行较大的跳转，增加寻找的时间。

|  |
| --- |
| 指导教师批阅意见：    成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。