**深 圳 大 学 实 验 报 告**

**课程名称：­ 算法设计与分析**

**实验项目名称： 动态规划—金罐游戏问题**

**学院： 计算机与软件学院**

**专业： 软件工程**

**指导教师： 马里佳**

**报告人： 张欣杰 学号： 2020151091**

**实验时间： 2022/05/06**

**实验报告提交时间： 2022/05/19**

**教务部制**

|  |
| --- |
| **一、实验目的：**  1.掌握动态规划算法设计思想  2.掌握金罐游戏问题的动态规划解法 |
| **二、实验原理：**   1. 通过实验，给出解决问题的动态规划方程； 2. 随机产生金罐的个数和金币值，对小数据模型利用蛮力法测试算法的正确性，并记录A选择的是哪个金罐；随机产生金罐的个数和金币值，对不同数据规模（n的值）测试算法效率，并与理论效率进行比对。 |
| **三、实验用品：**  操作系统：Windows 10  实验环境：Microsoft Visual Studio 2022 |
| **四、实验过程及内容：**   1. 暴力法 2. 算法分析     图 1算法详解  使用暴力法求解金罐游戏的本质是使用递归求解如何取金币最大问题，上图为递归算法流程图，由于A和B的策略都是最佳策略，打红色圈圈的情况是不可能出现的，也正因如此，在本问题中，我们从递归底层往上层看，可以发现从开始的选择就决定了游戏的胜负，因此通过递归即可得到最佳选择，因此A要获胜则需要选择递归顶层的选择。   1. 暴力法算法伪代码   MaxCoins(left, right, coins) //left为左端，right为右端，coins为罐子序列  if right-left==0  return coins[right]  Left=coins[left]-MaxCoins(left+1, right, coins)  Right=coins[right]-MaxCoins(left, righy-1, coins)  if Left>Right  return Left  return Right   1. 动态规划 2. 算法分析   动态规划的核心思想是定义一个二维数组dp[i][j]，其意义是在区间[I,j]的罐子中先手玩家与后手玩家最终得到的金币数量的差值。若A在dp[i][j]中取了左端罐子coins[i]，则B在dp[i+1][j]中成为先手玩家，则此时A与B的金币差值为dp[i+1][j]，那么玩家A在区间[i,j]中与B最终取得的金币的差值为dp[i][j]=coins[i]-dp[i+1][j]；若在开始时A取的时右端的罐子coins[j]，则最终A与B的金币差值为dp[i][j-1]，最终差值为dp[i][j]=coins[i]-dp[i][j-1]。   1. 算法伪代码   DP(dp, coins, n) //dp为动规数组，coins为罐子数组，n为罐子个数  for i=n-2 to 0  for j=i+1 to n-1  L=coins[i]-dp[i+1][j]  R=coins[j]-dp[i][j-1]  if L>R  dp[i][j]=L  else  dp[i][j]=R   1. 时间复杂度分析 2. 暴力法   在暴力法中，使用的是递归的方法，每次都考虑玩家A和B选取两端罐子的两种选择，因此，在n个罐子中，一共有2^n-1种选择，因此蛮力法的理论时间复杂度为   1. 动态规划   根据动态规划算法分析可以得到，在本问题中，不难推导出动态规划方程，通过推到得到动态规划方程为  通过推导式和算法分析可以计算，在动态规划求解本问题的时候，只需要计算数组中的个元素的值，因此可以推导出动态规划的理论时间复杂度为   1. 小数据规模测试 2. 暴力法小规模数据测试   使用10个数据验证算法正确性    图 2 验证10个数据的正确性   1. 动态规划法小规模数据测试   使用100个数据验证算法正确性    图 3验证100个数据的正确性 |
| **五、实验现象及数据处理：**   1. 大规模数据运行结果 2. 暴力法   在暴力法中，求解该问题的时间复杂度是十分高的，理论时间复杂度达到了指数级别，因此在运行暴力法求解金罐游戏问题的时候，每增加一个数据，对整体运行时间的影响是十分巨大的，在测试暴力法的数据时，当数据量为10的时候，10次运行的平均时间为96.8us，当数据量达到20的时候，平均运行时间增长至80935.7us，当数据量来到26的时候，平均运行时间更是增长到了6273755us。  暴力法的平均运行时间    图 4暴力法数据测试   1. 动态规划   使用动态规划算法时，速度比暴力法快了许多，特别是在大规模数据的时候，动态规划算法在时间复杂度的优化上展现了巨大的优越性，在数据量达到10000的时候，动态规划算法运行的时间与暴力法数据个数为20个的时候相近，但是高效率的算法是由空间换来的，当运行动态规划算法的时候，内存占用量时十分巨大的。  动态规划算法平均运行时间    图 5动态规划数据测试   1. 动态规划算法优化   在使用普通动态规划算法的时候，由于动态规划数组时二维的，因此在向系统申请空间的时候，dp数组的内存开销时十分巨大的，当运行50000数据的时候，即使在计算完及时delete dp数组以释放空间，但多次运行的时候依旧会导致内存超限，如下所示，内存开销十分巨大    图 6未优化的空间复杂度  针对内存开销过大的情况，提出空间优化算法，通过分析，在同台规划方程中，dp[i][j]的值只与dp[i+1][j]和dp[i][j-1]有关，当得到dp[i][j]的值的时候，dp[i+1][j]的值则不会再使用，因此可以省略i，使用一维数组dp[j]来代替二维数组dp[j]，优化后的动态规划方程如下  经过优化后，再次运行代码，试运行50000数据量的代码，代码成功运行，通过诊断窗口查看，可以观察到优化后的算法的空间复杂度大大降低。    图 7优化后的算法空间复杂度   1. 优化后的算法大规模运行效率   将动态规划算法进行优化之后，经过小规模数据测试验证算法正确性之后，使用大规模数据运行，得到平均运行结果。    图 8大规模数据测试   1. 算法效率分析 2. 暴力法   通过数据分析，可以得到如下数据，理论时间以6个数据为基准计算。    图 9暴力法时间对比图  通过对比，在前部分运行时间中，理论值与实际值是十分接近的，但是当数据量较大的时候，理论时间和实际时间是有误差的，猜测误差是由电脑运行环境及性能引起的。   1. 未优化的动态规划算法   通过数据分析，可以得到如下数据，理论时间以6个数据为基准计算。    图 10未经优化的动态规划算法  理论实践是以数据量为1000的平均运行时间作为基准值进行计算的，通过对比，理论值和实际值的值是十分接近的，均符合O(n2)。   1. 优化后的动态规划算法   通过数据分析，可以得到如下数据，理论时间以6个数据为基准计算。    理论实践是以数据量为10000的平均运行时间作为基准值进行计算的，通过对比，理论值和实际值的值是十分接近的，并且运行时间与未优化的动态规划相近，算法时间复杂度均符合O(n2)，但空间复杂度大大降低。 |
| **六、实验结论：**   1. 通过本次实验，基本了解了动态规划的过程，每次决策依赖于当前状态，又随即引起状态的转移。一个决策序列就是在变化的状态中产生出来的，这种多阶段最优化决策解决问题的过程被称为动态规划。 2. 金罐游戏问题使用蛮力法求解和动态规划求解花费差别极大，蛮力求解通过递归求解，其求解过程为一棵满二叉树，故花费的时间为θ(2^n)，而动态规划通过保留子问题的解，时间花费约为θ(n^2)，即两种算法的时间复杂度分别为O(2^n)和O(n^2)。 3. 由于未经优化的动态规划算法由于使用的是二维数组储存，因此当测试大数据据的时候，很容易造成内存溢出，当测试50000数据的时候，内存占用高达30Gb，因此需要对动态规划数组进行优化，可以将动态规划数组降至一维，优化完成的动态规划算法，空间内存占用大大降低。 |
| **思考题：** |
| **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**。