# 实验四 动态规划实验

1. **实验要求**

**实 验 l: 实 现 求 解 矩 阵 链 相 乘 问 题 的 算 法 。 定 义 的 规 模 为， 用 动 态 规 划 法 对 矩 阵 规 模 序 列 求 矩 阵 链 最 优 括 号 化 方 案 。 其 中 ， p 为 （ 1~50 ） 内 随 机 数 ， n 的 规 模 取 值 分 别 为 ： 4 ， 8 ， 12 ， 16 ， 20 ， 24 ， 28 。 给 出 算 法 运 行 所 需 的 时 间 ， 画 出 时 间 曲 线 。**

**实 验 2 ： 实 现 求 最 长 公 共 子 序 列 的算法 法 。 序 列 x 的 长 为 m ， 序 列 Y 的 长 为 n ， 序 列 x 和 Y 的 元 、 从 26 个 大 写 字 母 中 随 机 生 成 ， m 和 n 的 取 值 ：**

**第 1 组 （ 20 ， 10），（20 ， 20 ），（ 20 ， 30），（20 ， 40 ），（ 20 ， 50 ），（ 20 ， 60 ），（ 20 ， 70 ）**

**第 2 组 （ 15 ， 25），（30 ， 25 ） ， （ 45 ， 25 ） ， （ 60 ， 25 ） ， （ 75 ， 25 ） ， （ 90 ， 25 ） ， （ 105 ， 25 ）**

**给 出 算 法 运 行 所 需 的 时 间 ， 画 出 时 间 曲 线 。**

1. **实验环境**

**编译环境：Microsoft Visual Studio 2015**

**机器内存：8G**

**时钟主频：2.5GHz**

1. **实验过程**

**a) 为两个实验建立分别建立project1,project2文件夹，每个文件夹分别包含3个文件夹：**

**Input文件夹：存放输入数据**

**Source文件夹：源程序**

**Output文件夹：输出数据**

**b)input：**

**实验一:实现输入文件中每行一个随机数据保留到input文件夹下的input1.txt供读取,可预先生成1-50之间的30个随机数，按照实验要求分别读取大小为4+1，8+1，12+1，16+1，20+1，24+1，28+1的矩阵规模序列，进行对应的矩阵链相乘的试验。**

**实验二:按照题目要求分为两组读文件,第一组生成的7对随机字符串存在input文件夹下的inputA.txt中供程序读取，第二组生成的7对随机字符串存放在 inputB.txt中.每行一个随机字符串**

**计算机生成了可选文字:**

**(c)output ：**

**． 实 验 一 ： 为 结 果 建 立 一 个 输 出 文 件 夹 output。其 输 出 结 果 输 出 到 该 文 件 夹 中 ：**

**． output.txt ：保 存 矩 阵 链 相 乘 括 号 化 方 案 ， 从 上 往 下 分 别 保 存 不 同 n 值 的 规 模 矩 阵 序 列 和 对应 的 括 号 化 方 案 保 存 到 output.txt 中**

**． time.txt ： 将 求 解 不 同 n 值 对 应 规 模 的 矩 阵 链 的 测 试 时 间 保 存 到 time.txt**

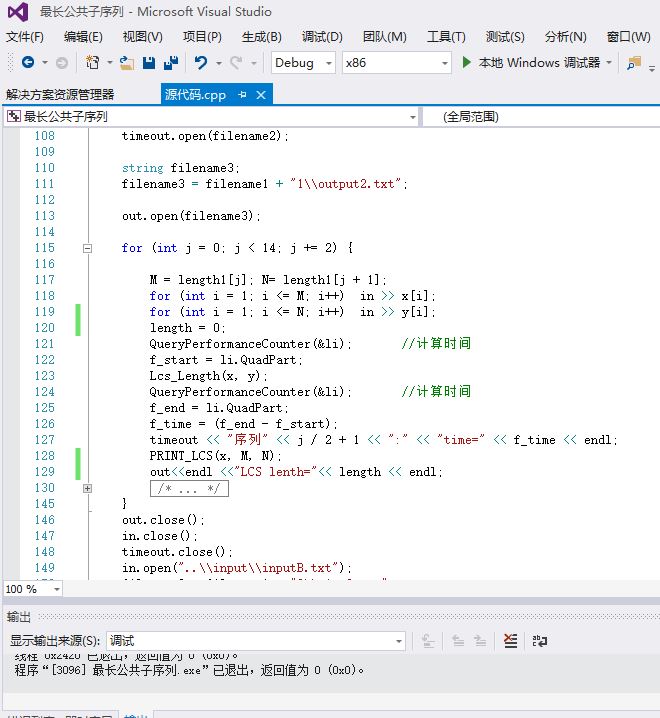
**． 实 验 二 ： 为 两 组 实 验 分 别 建 立 一 个 子 文 件 夹 e × 1 ， e × 2 ， 两 个 实 验 分 别 的 七 个 输 出 结 果 输 出 保 存**

**到 对 应 子 文 件 夹 中 ：**

**． output2.txt ： 要 求 按 序 号 输 出 每 个 公 共 序 列 的 字 符 串 和 长 度 保 存 到 output. 中**

**· time2.txt ： 将 求 解 这 组 实 验 每 个 解 所 消 耗 时 间 保 存 到 对 应 的 time.txt**

**d）实验过程截图：**

****

1. **实验关键代码**

**使用windows系统的计时api来计算算法的时间性能**

****

**a ） 实 验 一 需 要 实 现 矩 阵 链 相 乘 问 题 基 本 算 法 主 要 包 括 如 下 。**

**1 ． 计 算 最 优 代 价**

****

**2 ． 构 造 最 优 解 并 输 出**

****

**b ） 实 验 二 需 要 实 现 最 长 公 共 子 序 列 问 题 基 本 算 法 主 要 包 括 如 下 ：**

**0.生成随机字符序列**

****

**1 ． 求 解 最 长 子 序 列 问 题 的 动 态 规 划 算 法**

****

**2 ． 输 出 求 解 的 最 长 子 序 列 的 递 归 算 法**

****

1. **实验结果与分析**

****

1. **实验1结果分析**

**矩阵链乘法时间复杂度为O(n^3)，但增加了O(n^2)的辅助空间。由图1可以看出，算法所用时间T基本与数据规模n的立方成线性关系。但并不明显，尤其是在n较小的时候。思考原因可能是数据规模过小，时间消耗主要为函数、辅助空间的申请与调用。**

**图1**

1. **实验2**

**最长公共子序列的时间复杂度为O(m\*n)，但也增加了O(m\*n)的辅助空间。**

**在第一组实验中，观察表中序列1的数据，发现数据明显不正常。用后六个数据做成图二，可以看出算法所用时间T与m\*n基本成线性关系，符合渐进复杂度。思考原因可能是算法在首次调用Lcs\_length（x,y）时是第一次申请额外空间，可能会有额外的时间开销。**

**图二**

**验证：在进行计算Lcs\_length（x,y）的时间性能之前，先对两个额外的字符串调用一次Lcs\_length(x,y)，得到的实验结果为：**

**序列1:time=21**

**序列2:time=37**

**序列3:time=49**

**序列4:time=68**

**序列5:time=75**

**序列6:time=82**

**序列7:time=95**

**即得到正常渐进时间的序列1用时。**

**在第二组实验中，用后六个数据做成图三，可以看出算法所用时间T与m\*n基本成线性关系，符合渐进复杂度。比较两组实验，第二组实验用时明显比第一组实验用时长。考虑原因是到c/c++中是按行存储数组中的元素，b[i][j],c[i][j]取值时，频繁的改变i比频繁的改变j更耗用时间。第二组实验序列1的异常也是因为此。**

**图三**

1. **实验心得**

**动态规划法能将原指数级的运算改进到多项式级，但需要额外的辅助空间。而当数据较小的，实验环境和存储空间对算法的所用时间有很重要的影响。**