

南京邮电大学

实验报告

(2025 / 2026 学年 第 二 学期)

课程名称	计算机问题求解：使用算法
实验名称	动态规划法
实验时间	2026 年 5 月 21 日
指导单位	计算机学院软件工程系
指导教师	

学生姓名	班级学号		
学院(系)	计算机学院	专 业	软件工程

实 验 报 告

实验名称	动态规划			指导教师	
实验类型	设计	实验学时	2	实验时间	2026 5 21
一、 实验目的和要求 掌握动态规划法思想在实际中的应用，理解最长公共子序列（LCS）的问题特征； 学会用动态规划和备忘录方法分析子问题结构，设计具体算法，并编程实现两输入序列的比较，输出最长公共子序列； 掌握矩阵链乘问题的最优子结构性质，用动态规划求解最优加括号方案，使总乘法次数最少。					
二、 实验环境(实验设备) 操作系统：（Windows 10等） 调试软件名称： 调试软件版本号： 上机地点：学6-					

三、实验原理及内容

任务一：最长公共子序列 (LCS)

【算法原理】

最长公共子序列问题具有最优子结构性质：设 $X = x_1x_2\dots x_m$, $Y = y_1y_2\dots y_n$, 定义 $dp[i][j]$ 为 X 前 i 个字符与 Y 前 j 个字符的 LCS 长度，递推关系为：

- 若 $X[i] == Y[j]$, 则 $dp[i][j] = dp[i-1][j-1] + 1$

- 否则 $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

自底向上填表后，通过回溯 b 矩阵还原具体的 LCS 字符串。备忘录方法则自顶向下递归，遇到已计算的子问题直接返回缓存结果，避免重复计算。时间复杂度均为 $O(mn)$ 。

任务二：矩阵链乘最优加括号

【算法原理】

矩阵链乘问题同样具有最优子结构：设矩阵 $A_1, A_2, \dots, A_n$, $m[i][j]$ 表示计算 $A_i \dots A_j$ 所需的最少乘法次数，递推关系为：

$$m[i][j] = \min\{m[i][k] + m[k+1][j] + p_{i-1} * p_k * p_j\}, i \leq k < j$$

自底向上，先计算链长为 2 的情况，逐步推到链长为 n ，同时记录最优分割点 $s[i][j]$ ，最后递归输出最优加括号方案。时间复杂度 $O(n^3)$ ，空间复杂度 $O(n^2)$ 。

代码实现

```
#include <iostream>
#include <vector>
#include <string>
#include <climits>
#include <functional>
using namespace std;

// =====
// 任务一：最长公共子序列 (LCS)
// 动态规划 + 备忘录方法
// =====

// 方法一：动态规划 (自底向上)
void lcs_dp(const string& X, const string& Y) {
    int m = X.size(), n = Y.size();
    // dp[i][j] = X前i个字符与Y前j个字符的LCS长度
    vector<vector<int>> dp(m + 1, vector<int>(n + 1, 0));
    // b[i][j] 记录路径方向: 1=左上(匹配), 2=上, 3=左
    vector<vector<int>> b(m + 1, vector<int>(n + 1, 0));

    for (int i = 1; i <= m; i++) {
        for (int j = 1; j <= n; j++) {
            if (X[i-1] == Y[j-1]) {
                dp[i][j] = dp[i-1][j-1] + 1;
                b[i][j] = 1;
            } else if (dp[i-1][j] >= dp[i][j-1]) {
                dp[i][j] = dp[i-1][j];
                b[i][j] = 2;
            } else {
                dp[i][j] = dp[i][j-1];
                b[i][j] = 3;
            }
        }
    }
}
```

实验报告

四、实验小结（包括问题和解决方法、心得体会、意见与建议等）

【遇到的问题及解决方法】

1. LCS 回溯路径时方向记录出错：最初对 b 矩阵的赋值逻辑混淆了上和左的方向，导致还原的字符串顺序错误。逐步跟踪 b 矩阵的填充过程后，厘清了三个分支的含义，问题解决。
2. 矩阵链乘的下标设计：维度数组 p 需要 $n+1$ 个元素而非 n 个，刚开始读入时少读了一个，运行时数组越界。调整输入格式后正常。

【心得体会】

动态规划的核心在于找到[最优子结构]——大问题的最优解包含小问题的最优解。LCS 和矩阵链乘虽然问题形式不同，但递推思路一脉相承。备忘录方法和自底向上的 DP 本质上等价，前者思路更自然，后者效率更可控。写这两个题时感受最深的是，状态定义清晰了，代码几乎自然地就出来了；反之若状态定义模糊，怎么调都容易出错

五.支撑毕业要求指标点

- (1) 3-2-M能够根据用户需求, 选取适当的研究方法和技术手段, 确定复杂工程问题的解决方案。
- (2) 3-3-H能综合利用专业知识对解决方案进行优化, 体现创新意识, 并考虑健康、安全以及环境等因素。
- (3) 7-2-M正确理解和评价计算机及应用领域复杂工程问题实施对环境保护及社会可持续发展等的影响, 评价产品周期中可能对人类和环境造成的损害和隐患。

六、指导教师评语

评 分	细 则	评分项	优秀	良好	中等	合格
	遵守实验室规章制度					
	学习态度					
	算法思想准备情况					
	程序设计能力					
	解决问题能力					
	课题功能实现情况					
	算法设计合理性					
	算法效能评价					
	回答问题准确度					
	报告书写认真程度					
	内容详实程度					
	文字表达熟练程度					
	其它评价意见					
	本次实验能力达成评价 (总成绩)					
成 绩		批阅人		日 期		