算法分析与设计 **计算机科学与工程学院**
2022 年春季学期 **网络工程系**

第 3 次实验：动态规划与贪心算法 (Draft)

主讲人: 宋昕 <songxin@xaut.edu.cn> *2022 年 5 月 5 日*

3.1 实验任务

3.1.1 动态规划算法

最优矩阵连乘问题

说明: 最优矩阵连乘问题采用动态规划的方法进行求解，动态规划算法的设计一般是自下而上（用循环来设计的）。但也可以用自顶向下（用递归函数）的方法进行设计，设计时需要准备一张表格，如果要用到表格中某行某列的值，先看值是否已经存在，如存在，则直接读取，如果不存在，则通过计算的方式得到，并填入表格，这种方法叫做备忘录法，它避免了直接递归产生的重复计算。

任务:

1. 用下文中给出的 MatrixChain 算法和 PrintSolution 算法，编程实现矩阵连乘最优值（最优的乘法次数）和最优解（最优的加括号方式）的计算。
2. 参考下文的 MatrixChanRec 算法（递归、未采用查表方式），将其改进为递归且查表的备忘录算法。
3. 实现上一步的备忘录算法，输出最优值，并用第一步实现的 PrintSolution 算法输出最优解。

MatrixChain(A, n, p)

```
1: 创建m[n,n]表和s[n,n]表                    ▷ m 表保存最优值，s 表保存最优值的分点位置
2: for i = 1 to n do
3:     m[i,j] = 0
4: for l = 2 to n do                          ▷ l 表示连乘矩阵的个数
5:     for i = 1 to n − l + 1 do              ▷ i 表示 m 表的行号
6:         j = i + l − 1                      ▷ j 表示 m 表的列号，已知 l 和 i 的条件下，j 由 l
                                                和 i 计算得到
7:         m[i,j] = ∞
8:         for k = i to j − 1 do
9:             q = m[i,k] + m[k+1,j] + p_{i−1} p_k p_j
10:            if q < m[i,j] then
11:                m[i,j] = q
12:                s[i,j] = k
13: return m and s
```

PrintSolution(s, i, n)

```
1: if i == j then
2:     print 'A_i'                            ▷ 输出字母 A 和下标变量 i 的值
3: else
4:     print '('                              ▷ 输出左括号符号
5:     PrintSolution(s, i, s[i,j])
6:     PrintSolution(s, s[i,j] + 1, j)
7:     print ')'                              ▷ 输出右括号符号
```

MatrixChainRec(m, i, j, p)

```
1: if i == j then
2:     m[i,j] = 0                             ▷ Basce case
3:     return m[i,j]
4: else
5:     for k = i to j − 1 do
6:         cost = MatrixChainRec(m,i,k,p) + MatrixChainRec(m,k+1,j,p) + p_{i−1} p_k p_j
7:         if m[i,j] > cost then
8:             m[i,j] = cost
9:     return m[i,j]
```

最长公共子序列问题 LCS

说明：LCS 问题和矩阵连乘问题十分相似，求解过程基本相同。任务：与矩阵连乘问题类似

1. 编程实现循环版的 LCS 最优值求解，并编程实现递归版的最优解输出。
2. 设计递归版的 LCS 最优值求解算法，即备忘录算法。
3. 编程实现上一步的备忘录算法。

选做题目

算法实现题 3-1：独立任务最优调度问题算法实现题 3-3：石子合并问题

3.1.2 贪心算法

贪心算法的重点是要找到贪心属性，以下题目先在草稿纸上验算并设计算法，找出一定的规律后再考虑编程实现）

算法实现题 4-1

会场安排问题

算法实现题 4-3

磁带最优存储问题

算法实现题 4-9

虚拟汽车加油问题

选做题目

算法实现题 4-4：磁盘最优存储问题

算法实现题 4-6：最优服务次序问题

算法实现题 4-7：多处最优服务次序问题

算法实现题 4-8：d 森林问题