

# 《算法分析与设计》第3次作业\*

姓名: 唐珞鑫 学号: 71117106 成绩: \_\_\_\_\_

## 算法分析题

**题目1:** 有 $N$ 个数字, 不改变它们的相对位置, 在中间加入 $K$ 个乘号和 $N - K - 1$ 个加号 (即每两个相邻数字之间都有一个符号) 使最终结果尽量最大, 请给出归纳公式 (例如在数字1、2、3、4、5中加入2个乘号, 计算方法有 $1 * 2 * (3 + 4 + 5) = 24$ ,  $1 * (2 + 3) * (4 + 5) = 45$ ,  $(1 + 2 + 3) * (4 + 5) = 45$ 等等)

**答:**  $F[i][j]$ : 表示前 $i$ 个数中加入 $j$ 个乘号所得的最大运算结果, (其中 $j \leq K$ )。  $sum[i]$ 表示前 $i$ 个数的和。 对于 $F[i][j]$ 而言, 设最后一个乘号出现在第 $m$ 个元素之前, 问题就转化成求前 $m - 1$ 个数加 $j - 1$ 个乘号最大的算式值载乘上第 $m$ 个元素到第 $i$ 个元素的和。

$$F[i][j] = \max\{F[i][j], F[m-1][j-1] * (sum[i] - sum[m-1])\} \quad F[i][0] = sum[i]$$

**题目2:** 顺序放好的  $n$  根钢管的重量各为 $W[i], 1 \leq i \leq n$ 。我们须要把他们依照顺序焊成一根钢管, 但每次焊接可任意选两根相邻的钢管来焊接。每次焊接的代价与被焊两段钢管的总重量成正比。为简单起见, 把代价定为被焊两段钢管的总重量。例如,  $W[1] = 5, W[2] = 1, W[3] = 2$ , 如果先把 $W[1]$ 和 $W[2]$ 焊好, 代价为 $5 + 1 = 6$ 。焊好的这块有重量6, 再把 $W[3]$ 焊上, 又要代价 $6 + 2 = 8$ , 总代价是14。但如果先焊 $W[2]$ 和 $W[3]$ , 再焊 $W[1]$ , 则总代价为11。用动态规划的方法设计一个算法计算出最优的焊接顺序使总代价最小。

**答:** 与矩阵连乘问题类似,  $m[i][j]$ 表示将第 $i$ 到第 $j$ 截钢管焊接好的最小代价,  $sum[i][j]$ 表示第 $i$ 根到第 $j$ 根钢管的质量和,  $k$  为断开点, 即是使 $i$ 到 $j$ 焊接代价最小的那个位置。

$m[i][j]$  可以递归地定义为:

$$m[i][j] = \begin{cases} 0 & i=j \\ \min\{m[i][k] + m[k+1][j] + sum[i][j]\} & i < j \end{cases}$$

---

\*要求: 1、分析题请用书面化语言给出详细分析过程; 2、实现题请先写出算法思想, 其次用伪代码描述, C++源码采用在线提交, 尽量AC, Nike name统一使用学号-姓名的格式; 3、作业提交PDF文件。

## 算法实现题

题目3： 题目内容见 <https://vjudge.net/contest/290313>

答： 用 $c[i][j]$ 记录序列 $X$ 和 $Y$ 的最长公共子序列的长度。当 $i = 0$ 或是 $j = 0$ 时，空序列是 $X, Y$ 的最长公共子序列，此时 $c[i][j] = 0$ 。其他情况下，可得递归关系如下：

$$m[i][j] = \begin{cases} 0 & i=j=0 \\ c[i-1][j-1] + 1 & x_i=y_i \\ \max\{c[i][j-1], c[i-1][j]\} & x_i \neq y_i \end{cases}$$

```
伪代码： LCSLength(){  
    for(iTom) {  
         $c[i][0] = 0;$   
    }  
    for(iTon) {  
         $c[0][i] = 0;$   
    }  
    for(iTom)  
    for(jTon) {  
        if( $x[i] == y[j]$ ) {  $c[i][j] = c[i-1][j-1] + 1; b[i][j] = 1;$  }  
        elseif( $c[i-1][j] >= c[i][j-1]$ ) {  $c[i][j] = c[i-1][j]; b[i][j] = 2;$  }  
        else {  $c[i][j] = c[i][j-1]; b[i][j] = 3;$  }  
    }  
}  
LCS(){  
    if( $i == 0 || j == 0$ ) return;  
    if( $b[i][j] == 1$ ) LCS( $i-1, j-1, x, b$ )  
    elseif( $b[i][j] == 2$ ) LCS( $i-1, j, x, b$ );  
    else LCS( $i, j-1, x, b$ );  
}
```