

# 第三次上机实验报告

## 一、问题介绍

编程实现以下功能，并随机产生多组数据分析算法复杂度：

- (1) 采用动态规划算法实现矩阵连乘，输出最优括号化方案。
- (2) 采用哈夫曼编程构造一组字符的最优编码。

## 二、算法思想介绍

### ● 矩阵连乘

- (1) 算法原理：采用动态规划算法。假设矩阵  $A_i$  连乘到  $A_j$  的最优括号化方案的分割点在  $A_k$  与  $A_{k+1}$  之间，则矩阵  $A_i$  连乘到  $A_k$  和  $A_{k+1}$  连乘到  $A_j$  为其最优子结构。以  $m[i][j]$  记录标量计算所需的最少次数，矩阵  $A_i$  的大小为  $p_{i-1} \times p_i$ ，可得递归式：若  $i=j$ ，则  $m[i][j]=0$ ；若  $i \neq j$ ，则  $m[i][j]=\min\{m[i][k]+m[k+1][j]+p_{i-1} \times p_k \times p_j\}$ 。采用自底向上的算法，按  $j-i$  增大的顺序依次求出  $m[i][j]$  的值，得到待求的  $m[1][n]$ 。
- (2) 数据结构：二维数组  $m$  存放矩阵  $A_i$  连乘到  $A_j$  所需的最少标量计算次数，二维数组  $s$  存放矩阵  $A_i$  连乘到  $A_j$  的最优括号化方案的分割点。

### ● 哈夫曼编码

- (1) 算法原理：采用贪心算法。将每一个字符构建为一棵树，按其频率排成最小优先队列。每次选择  $\text{freq}$  值最小的两棵树合并为一棵树，其根结点的  $\text{freq}$  为两颗子树  $\text{freq}$  值之和，并插入最小优先队列中，直到队列中只有一棵树，即为哈夫曼树。
- (2) 数据结构：用结构体表示结点，其中包含频率  $\text{freq}$ 、左孩子  $\text{left}$  和右孩子  $\text{right}$ 。最小优先队列用最小堆实现，存放在数组中。哈夫曼树结点间的关系用链表表示。

## 三、实验中问题、理解与解决方法

- 矩阵连乘算法中数组  $m$  和  $s$  是二维数组，所占内存大，注意  $n$  值不能取太大，否则会溢出。
- 矩阵连乘算法中运行时间与矩阵个数有关，与矩阵大小无关，故矩阵的行、列数只需在 1 到 100 的整数中取值即可。
- 哈夫曼编程中，为了构建哈夫曼树，最小优先队列中应存放树（根结点）而不是频率  $\text{freq}$ 。
- 哈夫曼树中结点间用链表连接，用到结构体中的  $\text{left}$  和  $\text{right}$ ，而最小优先队列中结点在数组中依次排列，调用函数  $\text{PARENT}$ 、 $\text{LEFT}$ 、 $\text{RIGHT}$ ，注意不能将两种结构混淆
- 由于不便于将树的结构输出，因此采用中序遍历和先序遍历哈夫曼树，可确定哈夫曼树的结构，进而得到各个字符的编码。

## 四、测试结果

使用课本上的示例进行测试:

### 矩阵连乘:

若测试，输入0；若分析性能，输入1

需要标量乘法次数: 15125

最优方案:

<<A1<A2A3>><<A4A5>A6>>

### 哈夫曼编码:

若测试，输入0；若分析性能，输入1

5

生成二叉树中序遍历结果为: 45 100 12 25 13 55 5 14 9 30 16

生成二叉树先序遍历结果为: 100 45 55 25 12 13 30 14 5 9 16

与课本上的结果相同。

## 五、性能分析

性能分析（随机生成矩阵大小和字符频率，记录时间）示例：

矩阵连乘:

若测试，输入0；若分析性能，输入1

1

输入连乘矩阵的个数，随机生成矩阵大小

100

需要标量乘法次数: 501348

最优方案:

[illegible]

总时间: 0.029s

### 哈夫曼编码:

若测试，输入0；若分析性能，输入1

1

输入字符个数，随机生成字符频率

50

生成二叉树中序遍历结果为: 18459 37088 18629 74881 18886 37793 18907 155686 9473  
19170 9697 39008 19838 80805 20875 41797 10206 20922 10716 361450 10934 22548 11  
614 46345 23797 95709 23895 49364 25469 205764 13383 26866 1304 2823 1519 6546 3  
723 13483 6937 54472 27606 110055 27786 55583 13718 27797 14079 852309 28225 569  
71 28746 115073 28828 58102 29274 234012 14770 29585 14815 59184 29599 118939 29  
613 59755 30142 490859 30395 61325 30930 123820 31053 62495 15552 31442 15890 25  
6847 31458 63614 16018 32156 16138 133027 3848 7831 640 1872 1232 3983 2111 1638  
4 8553 33751 17367 69413 8556 17422 8866 35662 18240

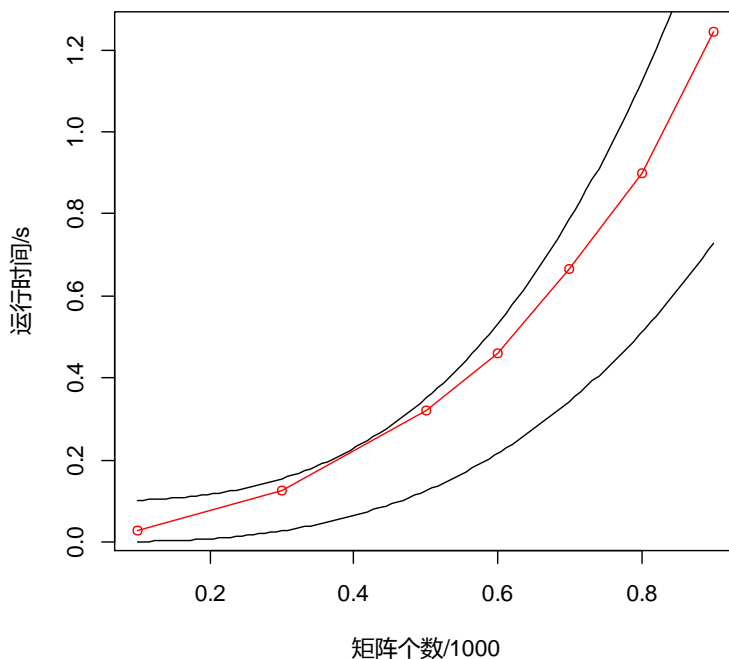
生成二叉树先序遍历结果为: 852309 361450 155686 74881 37088 18459 18629 37793 188  
86 18907 80805 39008 19170 9473 9697 19838 41797 20875 20922 10206 10716 205764  
95709 46345 22548 10934 11614 23797 49364 23895 25469 110055 54472 26866 13383 1  
3483 6546 2823 1304 1519 3723 6937 27606 55583 27786 27797 13718 14079 490859 23  
4012 115073 56971 28225 28746 58102 28828 29274 118939 59184 29585 14770 14815 2  
9599 59755 29613 30142 256847 123820 61325 30395 30930 62495 31053 31442 15552 1  
5890 133027 63614 31458 32156 16018 16138 69413 33751 16384 7831 3848 3983 1872  
640 1232 2111 8553 17367 35662 17422 8556 8866 18240

总时间: 0.001s

矩阵连乘，作时间关于矩阵个数的表格：

数组个数	100	300	500	600	700	800	900
时间一/s	0.028	0.129	0.315	0.459	0.654	0.892	1.233
时间二/s	0.03	0.12	0.314	0.451	0.687	0.913	1.23
时间三/s	0.029	0.125	0.328	0.463	0.652	0.892	1.265
平均时间/s	0.029	0.125	0.319	0.458	0.664	0.899	1.243

作出曲线图，运行时间-矩阵个数/1000 图（红线）



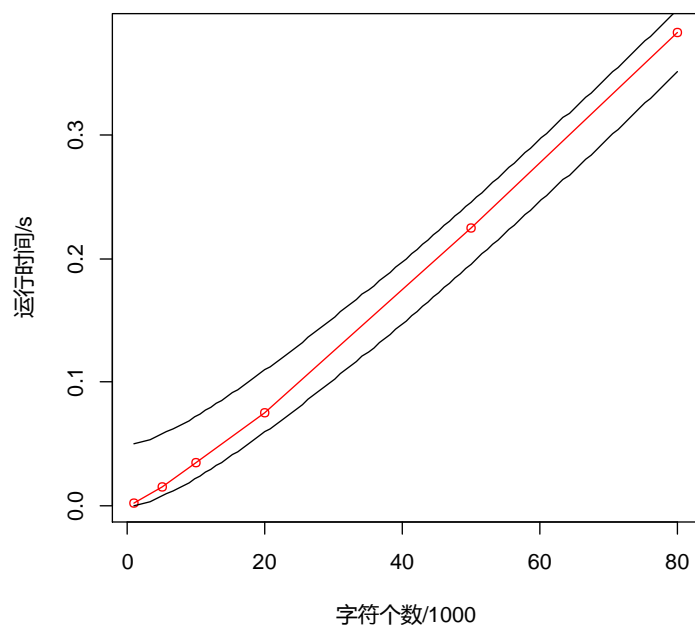
下面的黑色曲线为  $y=x^3$ ，上面的黑色曲线为  $y=2x^3+0.1$ ，则此曲线趋势为三次函数，且在  $y=x^3$  与  $y=2x^3+0.1$  之间。

哈夫曼编码，作时间关于字符个数的表格：

字符个数	1000	5000	10000	20000	50000	80000
时间一/s	0.004	0.015	0.037	0.075	0.231	0.402
时间二/s	0.003	0.017	0.034	0.079	0.225	0.375
时间三/s	0.003	0.016	0.034	0.074	0.218	0.368
平均时间/s	0.003	0.016	0.035	0.076	0.225	0.382

作出曲线图，运行时间-字符个数/1000 图（红线）

下面的黑色曲线为  $y=x\lg(x)/1000$ ，上面的黑色曲线为  $y=x\lg(x)/1000+0.05$ ，则此曲线趋势为  $x\lg(x)$  函数，且在  $y=x\lg(x)/1000$  与  $y=x\lg(x)/1000+0.05$  之间



## 六、 实验结论

用动态规划算法实现矩阵连乘最优解的时间复杂度为  $\Theta(n^3)$ ，用贪心算法实现哈夫曼编码的时间复杂度为  $O(n \lg(n))$