# 算法基础 第二次程序作业

## 问题提出

1. 矩阵链乘问题 对于一列矩阵`A1, A2, ... , An`, 它们的乘积`A1*A2*...*An`因为结合律，能有各种不同的计算方法，可以在此问题上坐最优化，达到最优计算复杂度。
2. 最长子序列问题 对于两列串`S = {s[1],s[2],...,s[n]}, L = {l[1],l[2],...,l[m]}`, 考虑其子串要相同（即存在`i[],j[]`,使得`s[i[k]] = l[j[k]]`, 其中`i[],j[]`是递增序列）。

## 算法解决

### 算法分析

对于问题1，可以归化成动态规划问题：不变子问题为 `Ai*...*Aj`的最优子结构，即`Ai*...*Aj`的最佳时间。其中有关系：若在`Ak`处断括号，则有 `Ai*...Aj = [Ai*...Ak] * [Ak*...*Aj]`，则计算量上为 `time(Ai*...Aj) = time(Ai*...Ak) + time(Ak*...*Aj) + time([Ai*...Ak] * [Ak*...*Aj])` 对任意的`k`如此，则可直接优化`k`使最优子结构最小。

在时间上，每次计算损失其实是常数, 而共计会遍历`i:1->n, j:i->n`, 而`i->j`的复杂度为`j-i`,所以为 $$\sum_{i=1}^n \sum_{j=i}^n j-i = \sum_{i=1}^n \frac{(n-i+1)(n-i)}{2} = O(n^3)$$ 所以时间复杂度为`O(n^3)`（同时也是$\Theta(n^3)$）

对于问题二，可以首先给出两序列之间所有元素之间的对等矩阵`m`, 考虑联想最优子结构`s[i,...,j]`和`l[k,...,l]`之间和子结构的关系,若`s[j]=l[l]`，无疑考虑子结构`s[i,...,j-1]`和`l[k,...,l-1]`, 若不然，则考虑子结构`s[i,...,j-1]`和`l[k,...,l]`或者子结构`s[i,...,j]`和`l[k,...,l-1]`, 由于上面总能找到一个顺序以满足每次到已求解的子问题上，所以能给出最优解。

同时应为长度的递增性，只需考虑`s[1,...,i]`和`l[1,...,j]`之间的公共子序列即可，每次遍厉在其对否相等的矩阵上， 所以在时间复杂度上，为`O(mn)`

## 实验环境

本次实验在`OSX 12`上进行, 编译器为clang++, 其中为C++11标准 调试命令`make`：会将两个问题一起编译并运行，`make all1`或`make all2`会分别编译运行, 并且将输出重定向到`output`文件夹下。

在实验过程中碰到了`1000000 * 1000000 * 100000 < 0`的情况：`C++`会吧`long int`数转化成`int`来计算，后者会溢出，要乘一个`1L`来避免这个情况。

## 结果分析

### 结果检验

本次实验结果如下 对于第一个问题

```
g++ main1.cpp -o main1 -std=c++11
./main1
       0      4653171388704   12785314527312  20636468892444  24391086983634  26862403677774  25609290706218  32624986215675  36690830959149  42524697503391
       0      0       5420360467176   13521323090772  17932867437726  20728338526014  20306401589670  25634842841343  30194832173733  35757101045571
       0      0       0       8600283908748   14324904389754  17768301169806  19007733992178  20963660781939  26511355295517  31530749805783
       0      0       0       0       342574866412884 468133318003848 254756069339178 371803367223632 426595146433708 512164553124588
       0      0       0       0       0       187078776037272 131448021886854 237712972811700 294784641741752 379100948115088
       0      0       0       0       0       0       49370292662076  127280669322339 190347814248692 271368815235559
       0      0       0       0       0       0       0       63919065213332  129944624226742 209339586199776
       0      0       0       0       0       0       0       0       79541022783614  151507453545772
       0      0       0       0       0       0       0       0       0       258856308500146
       0      0       0       0       0       0       0       0       0       0

       0      0       1       1       1       1       1       1       1       1
       0      1       1       1       1       1       1       1       1       1
       0      0       2       2       3       4       5       6       7       8
       0      0       0       3       3       3       3       6       6       6
       0      0       0       0       4       4       4       6       6       6
       0      0       0       0       0       5       5       6       6       6
       0      0       0       0       0       0       6       6       6       6
       0      0       0       0       0       0       0       7       7       8
       0      0       0       0       0       0       0       0       8       8
       0      0       0       0       0       0       0       0       0       9
```

对于第二个问题

```
g++ main2.cpp -o main2 -std=c++11
./main2
CDAABADBDD
DCABCCABCA
(5,9)   (4,7)   (3,6)   (2,2)   (0,1)
a.k.a
A B A A C
A B A A C

BCDADCCBCDDCCDC
ACCBAADBACCACDB
(13,13) (12,12) (11,10) (8,9)   (7,7)   (4,6)   (3,5)   (0,3)
a.k.a
D C C C B D A B
D C C C B D A B

BABBCBBACADADCABBADA
CBACDAAABDCBDBDBABAA
(19,19) (17,18) (16,17) (14,16) (13,10) (12,9)  (11,7)  (9,6)   (7,5)   (4,3)   (1,2)   (0,1)
a.k.a
A A B A C D A A A C A B
A A B A C D A A A C A B

BDCCBABDDDBBBBDCDDCDDBADD
CDCDBABBCCCBDDBCADCCACBDD
(24,24) (23,23) (21,22) (18,21) (15,19) (14,17) (10,14) (9,13)  (8,12)  (6,11)  (5,5)   (4,4)   (2,2)   (1,1)
a.k.a
D D B C C D B D D B A B C D
D D B C C D B D D B A B C D

ABDDDBBDDBDCDDBBCDBDDCCBCCDCCC
ADDACABBAAACCCCAADABBCCADDDCBD
(23,28) (22,27) (20,26) (19,25) (17,24) (16,22) (15,20) (14,19) (12,17) (11,14) (6,7)   (5,6)   (3,2)   (2,1)
a.k.a
B C D D D C B B D C B B D D
B C D D D C B B D C B B D D
```

## 时间分析

使用记录时间，结果如下

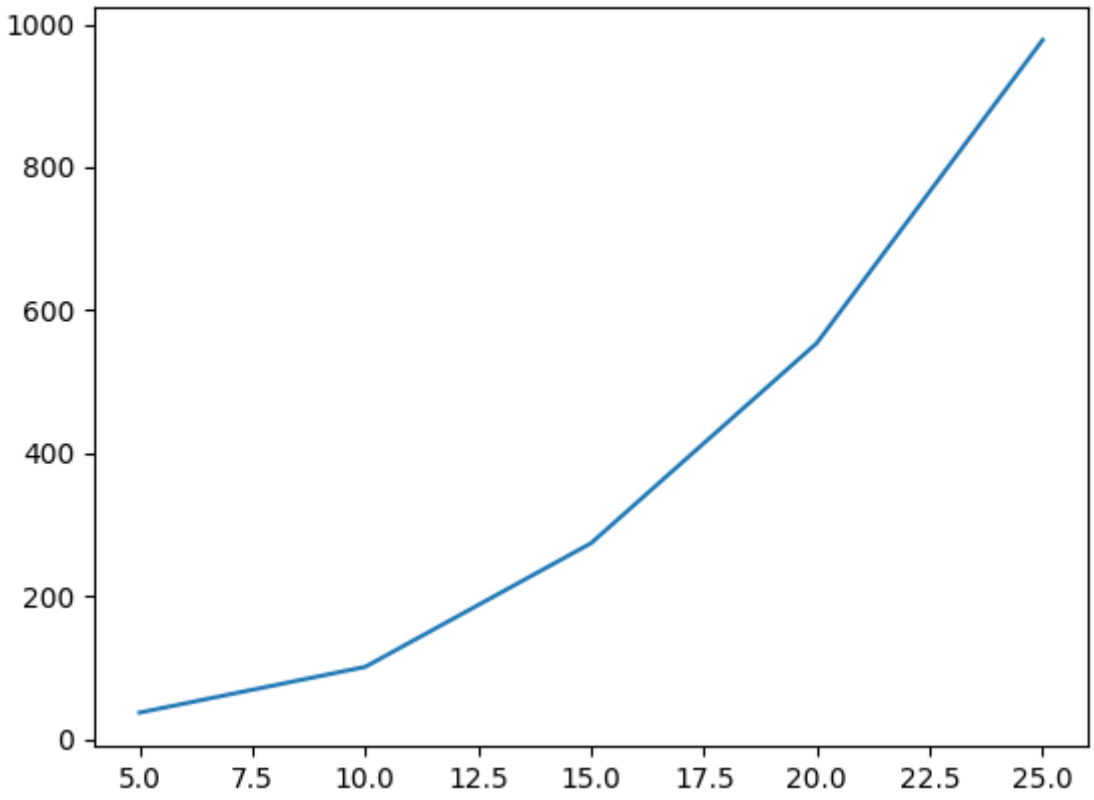
**问题一**

```
g++ main1.cpp -o main1 -std=c++11
./main1
Using time 37
Using time 101
Using time 274
Using time 554
Using time 978
```

也就是图表的形式

| (序列长度) | 5 | 10 | 15 | 20 | 25 |
|---|---|---|---|---|---|
| (micorseconds) | 37 | 101 | 274 | 554 | 978 |

以及其时间图




分析：可知其时间复杂度大致在`O(n^2)`到`O(n^3)`之间，这是应为操作的内容还不算太大，在实际中基本常数时间操作占据了一部分时间导致的，以及还有数据中的不确定性。

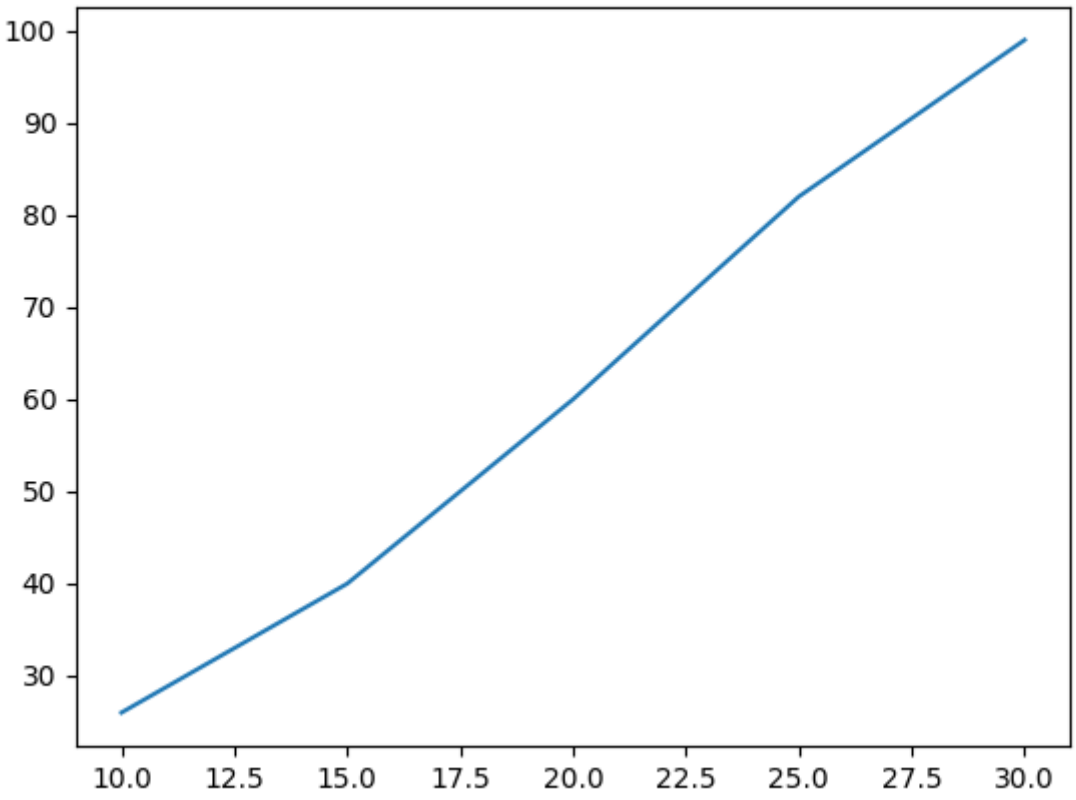
**问题二**

```
g++ main2.cpp -o main2 -std=c++11
./main2
Using time 26
Using time 40
Using time 60
Using time 82
Using time 99
```

或者图标的形式

| (序列长度) | 10 | 15 | 20 | 25 | 30 |
|---|---|---|---|---|---|
| (micorseconds) | 26 | 40 | 60 | 82 | 99 |

以及其时间统计图




分析：可知其时间复杂度大致在`O(n)`到`O(n^2)`之间（理论值为`O(n^2)`）

1. 操作的内容不是很大，时间开销本来就少。
2. 实际操作中固定的操作会占比大（比如分配数组），导致影响小。

## 总结

这次作业提高了编程能力