第四章作业

1. 给定一个整数序列$a_1, \ldots, a_n$。相邻两个整数可以合并，合并两个整数的代价是这两个整数之和。通过不断合并最终可以将整个序列合并成一个整数，整个过程的总代价是每次合并操作代价之和。试设计一个动态规划算法给出$a_1, \ldots, a_n$的一个合并方案使得该方案的总代价最大。设计动态规划算法求解此问题并分析算法的时间复杂性。

假设：$m[i,j]$为计算$a_i, a_{i+1}, \ldots a_j$的最大代价

$$\therefore m[i,j] = \begin{cases} \max_{i \le k < j} \{ m[i,k] + m[k+1,j] + \sum_{q=i}^{j} a_q \} & i<j \\ 0 & i=j \end{cases}$$

类似矩阵链乘法

```
COMBINE-NUMBER-CHAIN(p)
n = length(p)
FOR i=1 TO n DO
    m[i,i]=0;
FOR l=2 TO n DO
    FOR i=1 TO n-l+1 DO
        j=i+l-1;
        m[i,j]=0;
        FOR k←i TO j-1 DO
            q=m[i,k]+m[k+1,j]+sum(i,j);
            IF q>m[i,j] THEN m[i,j]=q; s[i,j]=k
Return m and s;
```

```
sum(i,j)            x←0
FOR k←i TO j DO
    x=x+a_i;
Return x;
```

时间复杂性为$O(n^3)$，空间复杂性为$O(n^2)$。递归调用输出s即为最优解

下证此问题具有优化子结构

若合并序列在k处断开，即$m_{1 \cup n} = m_{1-k} + m_{k+1 \cup n}$为优化解

则在对应子问题m_{1-k}和$m_{k+1 \cup n}$的解也必为当前优化解，

否则会出现一个更好的优化解，与假设矛盾。

第五章作业

1. 现有一台计算机，在某个时刻同时到达了n个任务。该计算机在同一时间只能处理一个任务，每个任务都必须被不间断地得到处理。该计算机处理这n个任务需要的时间分别为$a_1, a_2, \ldots, a_n$。将第i个任务在调度策略中的结束时间记为e_i。请设计一个贪心算法输出这n个任务的一个调度使得用户的平均等待时间$\frac{1}{n}\sum e_i$达到最小。

每次选择a_i最小的活动

(设$a_1, a_2, \ldots, a_n$未排序)

时间复杂度为$O(n\log n)$

↑主要取决于排序算法的性能.

```
Greedy-Activity-Choosing
n←length(A);
Sort(A); S←∅;
For i←1 To n Do
    S←S∪{A[i]};  // 记A[i]为排序后的任务下标
Return S;
```

① 证明此问题具有优化子结构.

设S是任务A的调度问题的一个优化解且包括任务时间最低的a_k.

则$S' = S - \{a_k\}$是$A' = \{a_i \in A \mid a_i \geq a_k\}$的优化解.

反之，若S'不为最优调度，先执行a_k再逐一执行S'任务将会得到一个比S更好的解. 故S'为S的优化子结构.

② 证明此问题具有贪心选择性.

对$|S|$作归纳法

当$|S|=1$时，显然成立.

设$|S|<k$时，命题成立

当$|S|=k$时，$S=\{a_k\}\cup S'$，S'为A'的优化解

由归纳假设，$S' = \bigcup_{i=2}^{k}\{l_i\}$（$l_i$为顺序排序的$a_i$），于是$A = \bigcup_{i=1}^{k}\{l_i\}$.