算法设计与分析

作业（五）

姓 名	熊恪峥
学 号	22920202204622
日 期	2022年3月23日
学 院	信息学院
课程名称	算法设计与分析

作业（五）

目录

1 题6.4

对于维度序列为$\{5,10,3,12,5,50,6\}$的矩阵链，可以按递推方程(1)填表 1和表 1

$$m_{i,j}=\begin{cases}0 & i=j\\ \min_{i\le k<j}\{m_{i,k}+m_{k,j}+p_{i-1}p_jp_k\} & i\neq j\end{cases} \tag{1}$$

表格 1: $m_{i,j}$内容

0	150	330	405	1655	2010
inf	0	360	330	2430	1950
inf	inf	0	180	930	1770
inf	inf	inf	0	3000	1860
inf	inf	inf	inf	0	1500
inf	inf	inf	inf	inf	0

表格 2: $s_{i,j}$内容

N	1	2	2	4	2
N	N	2	2	2	2
N	N	N	3	4	4
N	N	N	N	4	4
N	N	N	N	N	5
N	N	N	N	N	N

并且根据$s_{i,j}$中的内容可以得到括号加法为

$$((A_1A_2)((A_3A_4)(A_5A_6)))$$

代码实现见附录：代码实现中的代码1

2 题6.6

$$P(n)=\begin{cases}\Theta(1) & n=1\\ \sum_{k=1}^{n-1}P(k)P(n-k) & n\ge 2\end{cases} \tag{2}$$

要证$P(n)=\Omega(2^n)$只需证$P(n)\ge c\cdot 2^n$

证： $P(n)\ge c\cdot 2^n$

当 $n=1$， $P(1)=1>c\cdot 2$，当$c\le 1$

若 $\forall k<n, P(k)\ge c\cdot 2^k$成立，则

$$\begin{aligned}P(n)&=\sum_{k=1}^{n-1}P(k)P(n-k)\\ &\ge c\cdot 2^1\times c\cdot 2^{n-1}+c\cdot 2^2\times c\cdot 2^{n-2}+\cdots+c\cdot 2^{n-1}\times c\cdot 2^1\\ &=c^2\cdot(n-1)\cdot(2^n)\\ &\ge c\cdot 2^n\end{aligned}$$

当

$$c \geq \frac{1}{n-1}$$

3 题6.8

由状态转移方程(3)

$$\begin{aligned} f &= \min\{f_1[n] + x_1, f_2[n] + x_2\} \\ f_1[n] &= \begin{cases} e_1 + a_{1,1} & j = 1 \\ \min\{f_1[j-1], f_2[j-1] + t_{2,j-1}\} + a_{1,j} & j \geq 2 \end{cases} \\ f_2[n] &= \begin{cases} e_2 + a_{2,1} & j = 1 \\ \min\{f_2[j-1], f_1[j-1] + t_{1,j-1}\} + a_{2,j} & j \geq 2 \end{cases} \end{aligned} \tag{3}$$

可知$f_1[n]$的计算引用了$f_1[n-1]$和$f_2[n-1]$，$f_1[n-1]$的计算引用了$f_1[n-2]$和$f_2[n-2]$，$\ldots$以此类推，设f_1, f_2中元素引用次数$r_1[n] = 1, r_2[n] = 1$，有

$$r_1[n-1] = r_1[n] + r_2[n], r_2[n-1] = r_1[n] + r_2[n]$$

易知

$$\begin{aligned} r_1[n] &= O(2^n) \\ r_2[n] &= O(2^n) \end{aligned}$$

则子问题都被多次计算，因此有重叠子问题的性质。

4 题6.10

根据状态转移方程可以实现算法 1

算法 1 LCS问题

Input: 两个字符及其长度，以及备忘录的缓冲区

procedure LCS(X,Y,m,n,$memo$)
  if $m == 0\, or\, n == 0$ **then**
    return $memo[i][j] = 0$
  if $memo[i][j] \neq -1$ **then**
    return $memo[i][j]$
  if $X[m] == Y[n]$ **then**
    return $memo[i][j] =$ LCS(X,Y,$m-1$,$n-1$,$memo$)+1
  else
    return $memo[i][j] = \max\{$ LCS(X,Y,$m-1$,n,$memo$),LCS(X,Y,m,$n-1$,$memo$)$\}$

5 题6.11

X_m和Y_n是A开始的字符串，不妨设存在长度≥ 2的最长公共子序列，则

$$X_m = A \dots s \dots$$
$$Y_n = A \dots s \dots$$

其中s可以是任意字符，表示起始的A之后任何一个重合的字符。

若存在一个最长公共子序列lcs_k，lcs_k不以A开头，则不妨设lcs_k以s开头。

则可以构造lcs'_k，令

$$lcs'_k = A + lcs_k = As \dots$$

由于X_m和Y_n是A开始的字符串，按照定义显然lcs'_k是X_m和Y_n的一个最长公共子序列。但

$$len(lcs'_k) = len(lcs_k) + 1 > len(lcs_k)$$

与最长公共子序列的最优性矛盾，因此假设不成立，一切最长公共子序列都以A开头

A 附录：代码实现

代码 1: 矩阵连乘

```
p =  list([5, 10, 3, 12, 5, 50, 6])
N = 6

m =  list([ list([0x7fffffff for i in  range(0, N + 1)]) for j in  range(0, N + 1)])
s =  list([ list([0x7fffffff for i in  range(0, N + 1)]) for j in  range(0, N + 1)])


def pretty_print(i, j):
    if i == j:
        print(’A{}’. format(i), end=’ ’)
    else:
        print("(", end=’ ’)
        pretty_print(i, s[i][j])
        pretty_print(s[i][j] + 1, j)
        print(")", end=’ ’)


for i in  range(1, N + 1):
    m[i][i] = 0

for i in  range(N, 0, -1):
    for j in  range(i, N + 1):
        if i == j:
            m[i][j] = 0
        else:
            for k in  range(i, j):
                val = m[i][k] + m[k + 1][j] + p[i - 1] * p[j] * p[k]
                if m[i][j] > val:
                    m[i][j] = val
                    s[i][j] = k

for i in  range(1, N + 1):
    for j in  range(1, N + 1):
        print("inf" if m[i][j] == 0x7fffffff else m[i][j], end=’ ’ if j != N else ’\n’)

for i in  range(1, N + 1):
    for j in  range(1, N + 1):
        print("None" if s[i][j] == 0x7fffffff else s[i][j], end=’ ’ if j != N else ’\n’)

pretty_print(1, N)
```