算法作业 6

孟妍廷 2015202009

2017 年 10 月 30 日

15.2-1

解: 依题意可得:

p_0	p_1	p_2	p_3	p_4	p_5	p_6
5	10	3	12	5	50	6

根据 MATRIX-CHAIN-ORDER 算法可得

$$m[i,i] = 0, i = 1....6$$

$$m[1,2] = m[1,1] + m[2,2] + p_0p_1p_2 = 150, k = 1$$
$$m[2,3] = m[2,2] + m[3,3] + p_1p_2p_3 = 360, k = 2$$
$$m[3,4] = m[3,3] + m[4,4] + p_2p_3p_4 = 180, k = 3$$
$$m[4,5] = m[4,4] + m[5,5] + p_3p_4p_5 = 3000, k = 4$$
$$m[5,6] = m[5,5] + m[6,6] + p_4p_5p_6 = 1500, k = 5$$

$$m[1,3] = min\begin{cases} m[1,1] + m[2,3] + p_0p_1p_3 = 0 + 360 + 5 \times 10 \times 12 = 960 \\ m[1,2] + m[3,3] + p_0p_2p_3 = 150 + 0 + 5 \times 3 \times 12 = 330 \quad \star \end{cases}, k = 2$$

$$m[2,4] = min\begin{cases} m[2,2] + m[3,4] + p_1p_2p_4 = 0 + 180 + 10 \times 3 \times 5 = 330 \quad \star \\ m[2,3] + m[4,4] + p_1p_3p_4 = 360 + 0 + 10 \times 12 \times 5 = 960 \end{cases}, k = 2$$

$$m[3,5] = min\begin{cases} m[3,3] + m[4,5] + p_2p_3p_5 = 0 + 3000 + 3 \times 12 \times 50 = 4800 \\ m[3,4] + m[5,5] + p_2p_4p_5 = 180 + 0 + 3 \times 5 \times 50 = 930 \quad \star \end{cases}, k = 4$$

$$m[4,6] = min\begin{cases} m[4,4] + m[5,6] + p_3p_4p_6 = 0 + 1500 + 12 \times 5 \times 6 = 1860 \quad \star \\ m[4,5] + m[6,6] + p_3p_5p_6 = 3000 + 0 + 12 \times 50 \times 6 = 6600 \end{cases}, k = 4$$

$$m[1,4] = min\begin{cases} m[1,1] + m[2,4] + p_0p_1p_4 = 0 + 330 + 5 \times 10 \times 5 = 580 \\ m[1,2] + m[3,4] + p_0p_2p_4 = 150 + 180 + 5 \times 3 \times 5 = 405 \quad \star \\ m[1,3] + m[4,4] + p_0p_3p_4 = 330 + 0 + 5 \times 12 \times 5 = 630 \end{cases}, k = 2$$

$$m[2,5] = min\begin{cases} m[2,2] + m[3,5] + p_1p_2p_5 = 0 + 930 + 10 \times 3 \times 50 = 2430 \quad \star \\ m[2,3] + m[4,5] + p_1p_3p_5 = 360 + 3000 + 10 \times 12 \times 50 = 9360 \\ m[2,4] + m[5,5] + p_1p_4p_5 = 330 + 0 + 10 \times 5 \times 50 = 2830 \end{cases}, k = 2$$

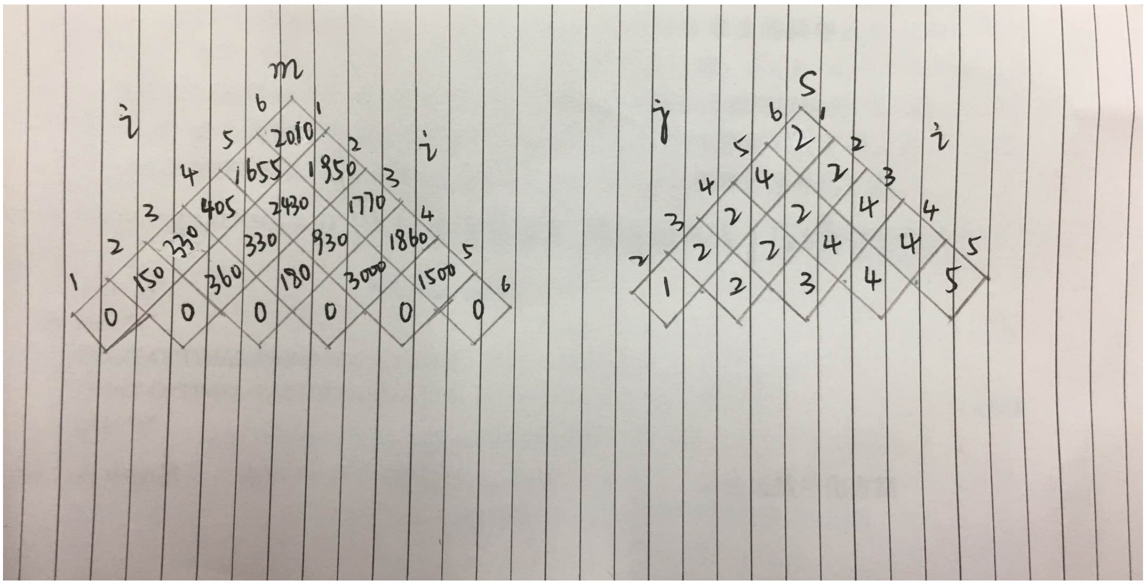
$$m[3,6] = min\begin{cases} m[3,3] + m[4,6] + p_2p_3p_6 = 0 + 1860 + 3 \times 12 \times 6 = 2076 \\ m[3,4] + m[5,6] + p_2p_4p_6 = 180 + 1500 + 3 \times 5 \times 6 = 1770 \quad \star \\ m[3,5] + m[6,6] + p_2p_5p_6 = 930 + 0 + 3 \times 50 \times 6 = 1830 \end{cases}, k = 4$$

$$m[1,5]=min\begin{cases} m[1,1]+m[2,5]+p_0p_1p_5=0+2430+5\times10\times50=4930 \\ m[1,2]+m[3,5]+p_0p_2p_5=150+930+5\times3\times50=1830 \\ m[1,3]+m[4,5]+p_0p_3p_5=330+3000+5\times12\times50=6330 \\ m[1,4]+m[5,5]+p_0p_4p_5=405+0+5\times5\times50=1655 \quad \star \end{cases},k=4$$

$$m[2,6]=min\begin{cases} m[2,2]+m[3,6]+p_1p_2p_6=0+1770+10\times3\times6=1950 \quad \star \\ m[2,3]+m[4,6]+p_1p_3p_6=360+1860+10\times12\times6=2940 \\ m[2,4]+m[5,6]+p_1p_4p_6=330+1500+10\times5\times6=2130 \\ m[2,5]+m[6,6]+p_1p_5p_6=2430+0+10\times50\times6=5430 \end{cases},k=2$$

$$m[1,6]=min\begin{cases} m[1,1]+m[2,6]+p_0p_1p_6=0+1950+5\times10\times6=2250 \\ m[1,2]+m[3,6]+p_0p_2p_6=150+1770+5\times3\times6=2010 \quad \star \\ m[1,3]+m[4,6]+p_0p_3p_6=330+1860+5\times12\times6=2550 \\ m[1,4]+m[5,6]+p_0p_4p_6=405+1500+5\times5\times6=2055 \\ m[1,5]+m[6,6]+p_0p_5p_6=1655+0+5\times50\times6=3155 \end{cases},k=2$$

可得 m 矩阵和 s 矩阵如图:



根据 PRINT-OPTIMAL-PARENS 算法可得最优括号方案为 $((A_1A_2)((A_3A_4)(A_5A_6)))$

5.2

解: 用动态规划求解:

由思考题讲解的分析可知, 对于任意字符串，如果头和尾相同，那么它的最长回文子序列一定是去头去尾之后的部分的最长回文子序列加上头和尾。如果头和尾不同，那么它的最长回文子序列是去头部分和去尾部分的最长回文子序列中较长的一个。则可知:

设字符串为s, $f(i,j)$表示$s(i...j)$的最长回文子序列

当$i>j$时, $f(i,j)=0$

当$i=j$, $f(i,j)=1$

当$i<j$时 :

$$f(i,j)=\begin{cases} 2+f(i+1,j-1), s[i]=s[j] \\ max\{f(i+1,j),f(i,j-1)\}, s[i]\neq s[j] \end{cases}$$

则算法如下:

$SOLUTION(s)$
  $let\ b[1....n,1...n], f[1...n,1...n] be\ new\ tables$
  $for\ i=1\ ton$
    $f[i,i]=1$
  $for\ i=n\ to\ 2$
    $for\ j=i-1\ to\ 1$
      $f[i,j]=0$
  $for\ i=n-1\ to\ 1$
    $for\ j=i+1\ to\ n$
      $if\ s[i]==s[j]$
        $f[i,j]=f[i+1,j-1]+2$
        $b[i,j]=\nwarrow$
      $elseif\ f[i+1,j]\geq f[i,j-1]$
        $f[i,j]=f[i+1,j]$
        $b[i,j]=\rightarrow$
      $else\ f[i,j]=f[i,j-1]$
        $b[i,j]=\uparrow$
  $return\ b\ and\ f$
时间复杂度为 $O(n^2)$.
$PRINT(b,s,i,j)$
  $if\ i<0\ or\ j<0\ or\ i>j$
    $return$ $if\ b[i,j]=\nwarrow$
    $PRINT(b,s,i+1,j-1)$
    $print\ s_i, s_j$
  $elseif\ b[i,j]=\rightarrow$
    $PRINT(b,s,i+1,j)$
  $else\ PRINT(b,s,i,j-1)$
时间复杂度为 $O(2n)$.

15-4(整齐打印)
解: 利用动态规划, 在考虑第 i 到第 j 个词时, 认为前 i-1 个词已经实现整齐打印, 初始状态为 0
算法如下:
$SOLUTION(s)$
  $fist[1...n]=0//$记录排好后, 每一行首单词下标
  $a[1...n,1...n]//$记录第i个数到第j个数若在一行的额外空格符数量
  $b[1...n,1...n]//$记录排好后第i个数到第j个数的额外空格符数量
  $rest[0...n]=\infty//$记录排到每一个状态时的额外空格符数量
  $for\ i=1\ to\ n$
    $a[i,i]=M-l_i$
    $for\ j=i+1\ to\ n$
      $a[i,j]=a[i,j-1]-l_j$
  $for\ i=1\ to\ n$
    $for\ j=i\ to\ n$
      $if\ a[i,j]<0//$说明不在一行
        $b[i,j]=\infty$
      $elseif\ j==n//$最后一行
        $b[i,j]=0$
      $else\ b[i,j]=a[i,j]^3$
  $rest[0]=0//$初始状态为0
  $for\ j=1\ to\ n$
    $for\ i=1\ to\ j$
      $if\ rest[i-1]\neq\infty\ and\ b[i,j]\neq\infty\ and\ rest[i-1]+b[i,j]>rest[i]$
        $rest[j]=rest[i-1]+b[i,j]//i$到$j$为一行
        $fist[j]=i//$首元素
时间复杂度为 $O(n^2)$, 空间复杂度为 O(2n).
$PRINT(s,first,j)$
  $i=first[j]//j$所在行的首元素下标
  $if\ i=1$

$return$
$PRINT(s, first, i-1)$//递归
$for\ k = i\ to\ j$
$print(s_k)$
时间复杂度为 O(n), 空间复杂度为 O(1).