动态规划算法的迭代实现

迭代计算的关键

- 每个子问题只计算一次
- 迭代过程
  - 从最小的子问题算起
  - 考虑计算顺序，以保证后面用到的值前面已经计算好
  - 存储结构保存计算结果——备忘录
- 解的追踪
  - 设计标记函数标记每步的决策
  - 考虑根据标记函数追踪解的算法

矩阵链乘法不同子问题

长度1：只含1个矩阵，有*n*个子问题(不需要计算)

长度2：含2个矩阵，*n*-1个子问题

长度3：含3个矩阵，*n*-2个子问题

…

长度*n*-1：含*n*-1个矩阵，2个子问题

长度 *n*：原始问题，只有1个

矩阵链乘法迭代顺序

长度为1：初值，$m[i, i] = 0$

长度为2：1..2，2..3，3..4，... , n-1..n

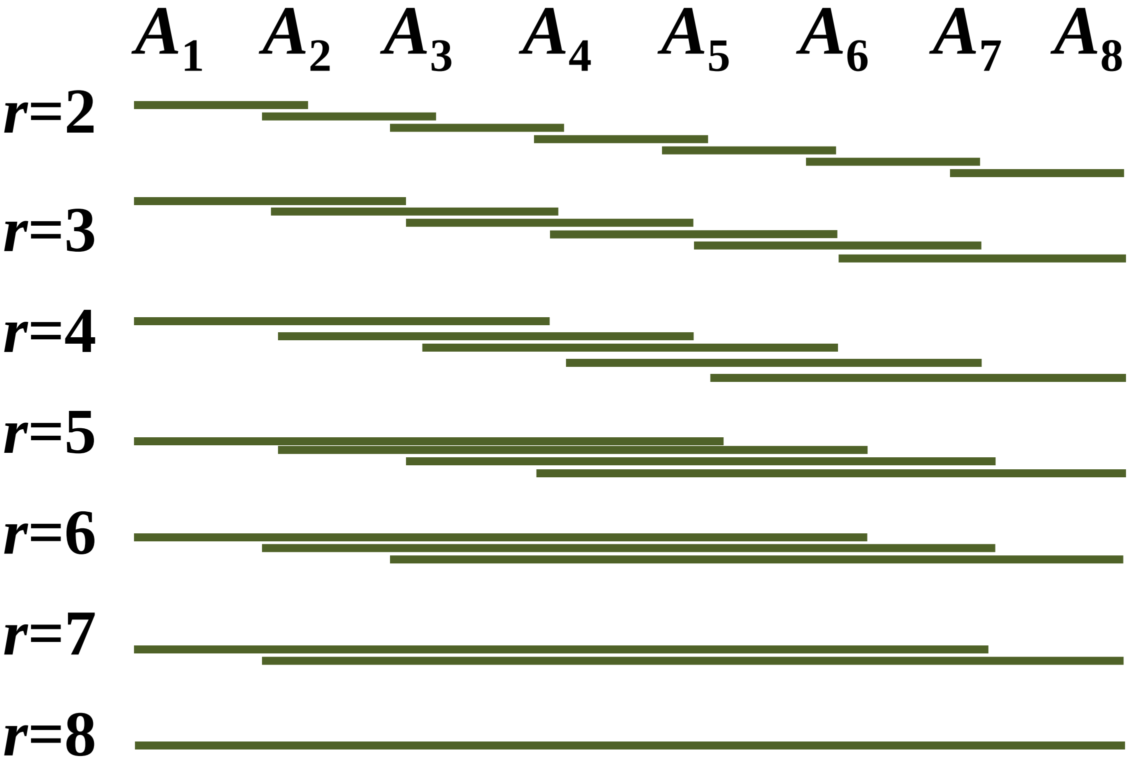
长度为3：1..3，2..4，3..5，... , n-2..n

...

长度为 n-1：1..n-1，2..n

长度为n：1..n

n=8的子问题计算顺序



算法 MatrixChain (P, n)

1. 令所有的 $m[i,i]$初值为0
2. for $r \leftarrow 2$ to n do // r为链长
3. for $i \leftarrow 1$ to $n-r+1$ do // 左边界i
4. $j \leftarrow i+r-1$ // 右边界j
5. $m[i,j] \leftarrow m[i+1,j]+p_{i-1}p_i p_j$ //$k=i$
6. $s[i,j] \leftarrow i$ //记录k
7. for $k \leftarrow i+1$ to $j-1$ do // 遍历k
8. $t \leftarrow \underline{m[i,k]+m[k+1,j]+p_{i-1}\,p_k\,p_j}$
9. if $t<m[i,j]$
10. then $m[i,j] \leftarrow t$ //更新解
11. $s[i,j] \leftarrow k$

遍历长r子问题

遍历所有划分

二维数组m与s为备忘录

时间复杂度

- 根据伪码：行 2, 3, 7 都是$O(n)$，循环执行$O(n^3)$次，内部为$O(1)$

$$W(n) = O(n^3)$$

- 根据备忘录：估计每项工作量, 求和.

  子问题有$O(n^2)$个，确定每个子问题的最少乘法次数需要对不同划分位置比较，需要$O(n)$时间.

$$W(n) = O(n^3)$$

- 追踪解工作量 $O(n)$，总工作量$O(n^3)$.

实例

输入： $P = <30, 35, 15, 5, 10, 20>$,

$n = 5$

矩阵链：$A_1 A_2 A_3 A_4 A_5$，其中

A_1: 30×35，A_2: 35×15，A_3: 15×5,

A_4: 5×10，A_5: 10×20

备忘录：存储所有子问题的最小乘法次数及得到这个值的划分位置.

备忘录 $m[i,j]$

$P = <30, 35, 15, 5, 10, 20>$

r=1	m[1,1]=0	m[2,2]=0	m[3,3]=0	m[4,4]=0	m[5,5]=0
r=2	m[1,2]=15750	m[2,3]=2625	m[3,4]=750	m[4,5]=1000	
r=3	m[1,3]=7875	m[2,4]=4375	m[3,5]=2500		
r=4	m[1,4]=9375	m[2,5]=7125			
r=5	m[1,5]=11875				

$$m[2,5] = \min\{0+2500+35\times15\times20,\ 2625+1000+35\times5\times20,\ 4375+0+35\times10\times20\} = 7125$$

标记函数 $s[i,j]$

r=2	$s[1,2]=1$	$s[2,3]=2$	$s[3,4]=3$	$s[4,5]=4$	
r=3	$s[1,3]=1$	$s[2,4]=3$	$s[3,5]=3$		
r=4	$s[1,4]=3$	$s[2,5]=3$			
r=5	$\boxed{s[1,5]=3}$				

解的追踪：$s[1,5]=3 \Rightarrow (A_1A_2A_3)(A_4A_5)$

$s[1,3]=1 \Rightarrow A_1(A_2A_3)$

输出

计算顺序：$(A_1(A_2A_3))(A_4A_5)$

最少的乘法次数：$m[1,5]=11875$

两种实现的比较

递归实现：时间复杂性高，空间较小

迭代实现：时间复杂性低，空间消耗多

原因：递归实现子问题多次重复计算，子问题计算次数呈指数增长. 迭代实现每个子问题只计算一次.

动态规划时间复杂度：

备忘录各项计算量之和 + 追踪解工作量

通常追踪工作量不超过计算工作量，是问题规模的多项式函数

11

动态规划算法的要素

- 划分子问题，确定子问题边界，将问题求解转 变成多步判断的过程.
- 定义优化函数,以该函数极大(或极小)值作为依据,确定是否满足优化原则.
- 列优化函数的递推方程和边界条件
- 自底向上计算，设计备忘录 (表格)
- 考虑是否需要设立标记函数
- 用递推方程或备忘录估计时间复杂度