

算清基础 HW7

15.2-1

< 5, 10, 3, 12, 5, 50, 6 >

| m | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|-----|-----|-----|------|------|
| 1 | 0 | 150 | 330 | 405 | 1655 | 2010 |
| 2 | | 0 | 360 | 330 | 2430 | 1950 |
| 3 | | | 0 | 180 | 930 | 1770 |
| 4 | | | | 0 | 3000 | 1860 |
| 5 | | | | | 0 | 1500 |
| 6 | | | | | | 0 |

| s | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|
| 1 | 1 | 2 | 2 | 4 | 2 |
| 2 | | 2 | 2 | 2 | 2 |
| 3 | | | 3 | 4 | 4 |
| 4 | | | | 4 | 4 |
| 5 | | | | | 5 |

$$((5 \times 10)(10 \times 3))((3 \times 12)(12 \times 5))((5 \times 50)(50 \times 6))$$

15.2-5

$$\begin{aligned}
 \sum_{l=2}^n \sum_{j=1}^{n+1-l} \sum_{k=i}^{j+l-2} 2 &= \sum_{l=2}^n \sum_{i=1}^{n+1-l} 2(l-1) \\
 &= \sum_{l=2}^n 2(l-1)(n+1-l) \\
 &= \sum_{l=1}^{n-1} 2l(n-l) \\
 &= 2n \sum_{l=1}^{n-1} l - 2 \sum_{l=1}^{n-1} l^2
 \end{aligned}$$

$$= 2n \cdot \frac{n(n-1)}{2} - 2 \cdot \frac{(n-1)n(2n-1)}{6}$$

$$= n^2(n-1) - \frac{1}{3}(n-1)n(2n-1)$$

$$= n(n-1) \left(n - \frac{2n}{3} + \frac{1}{3} \right)$$

$$= \frac{n(n-1)(n+1)}{3}$$

$$= \frac{n^3 - n}{3}$$

15.3-2

MERGE-SORT($a[0] \dots a[15]$)

MERGE-SORT($a[0 \dots 7]$)

MERGE-SORT($a[8 \dots 15]$)

M-S($a[0 \dots 3]$)

M-S($a[4 \dots 7]$)

M-S($a[8 \dots 11]$)

M-S($a[12 \dots 15]$)

M-S($a[0 \dots 1]$)

M-S($a[2 \dots 3]$)

M-S

M-S

M-S

M-S

M-S

M-S

$a[0]$

$a[1]$

$a[2]$

$a[3]$

$a[4 \dots 5]$

$a[6 \dots 7]$

$a[8 \dots 9]$

$a[10 \dots 11]$

$a[12 \dots 13]$

$a[14 \dots 15]$

$a[4]$

$a[5]$

$a[6]$

$a[7]$

$a[8]$

$a[9]$

$a[10]$

$a[11]$

$a[12]$

$a[13]$

$a[14]$

$a[15]$

该算法对排序的数组的任何一对索引最多执行一次调用,即这些子问题并不重叠,因此备忘技术对 MERGE-SORT 无效.

15.3-4.

假设给定矩阵 A_1, A_2, A_3, A_4 , $p = \langle 1000, 100, 20, 10, 1000 \rangle$.

当 $k=3$ 时 $p_0 p_k p_4$ 最小. 下面解决子问题 $A_1 A_2 A_3$, $k=2$ 时 p_0, p_k, p_3 最小. 得到结果: $((A_1 A_2) A_3) A_4$

需要 $1000 \times 100 \times 20 + 1000 \times 20 \times 10 + 1000 \times 10 \times 1000 = 1220000$ 次乘法

若用动态规划, 可得 $((A_1 (A_2 A_3)) A_4)$

需要 $100 \times 20 \times 10 + 1000 \times 100 \times 10 + 1000 \times 10 \times 1000 = 11020000$

可见贪心算法得到了次优解.

15.4-1

$\langle 1, 0, 0, 1, 1, 0 \rangle$ 或 $\langle 1, 0, 1, 0, 1, 0 \rangle$

15.4-4

$2 \times \min(m, n)$: 由于程序中反用到 $c[i-1, j-1]$ $c[i-1, j]$ $c[i, j-1]$, 所以使用两个数组, 第一个存储上一行的 $c[i, j]$, 第二个存储当前行的 $c[i, j]$. 最开始第一个数组初始化为 0. 最左侧一列全为 0 当作已知条件即可.

$\min(m, n) + O(1)$:

用一个长为 $\min(m, n) + 1$ 的数组 a 记录, 其中当需要计算 $c[i, j]$ 时 a 中:

$$\begin{cases} a[k] = c[i, k], & 1 \leq k < j-1 \\ a[k] = c[i-1, k], & k \geq j-1 \\ a[0] = c[i, j-1] \end{cases}$$

开始时应将 a 初始化为 0 且从左向右计算.

$c[i, j]$ 需要的 3 个值: $c[i, j-1] = a[0]$, $c[i-1, j-1] = a[j-1]$

$c[i-1, j] = a[j]$

当 $c[i, j]$ 计算好后将 $a[j]$ 放入 $a[j-1]$, 将 $c[i, j]$ 放入 $a[0]$.
即可.