动态规划篇: 钢条切割问题

童咏昕

北京航空航天大学 计算机学院

中国大学MOOC北航《算法设计与分析》



- 钢条切割
 - 现有一段长度为10的钢条,可以零成本将其切割为多段长度更小钢条

钢条长度	0	1	2	3	4	5	6	7	8	9	10
价格p	0	1	5	8	9	10	17	17	20	24	24



• 钢条切割

• 现有一段长度为10的钢条,可以零成本将其切割为多段长度更小钢条

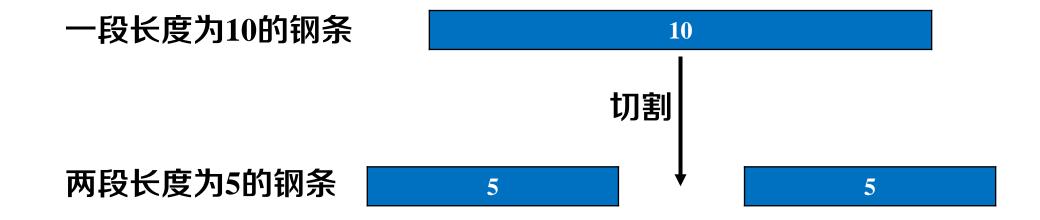
钢条长度	0	1	2	3	4	5	6	7	8	9	10
价格p	0	1	5	8	9	10	17	17	20	24	24

一段长度为10的钢条



- 钢条切割
 - 现有一段长度为10的钢条,可以零成本将其切割为多段长度更小钢条

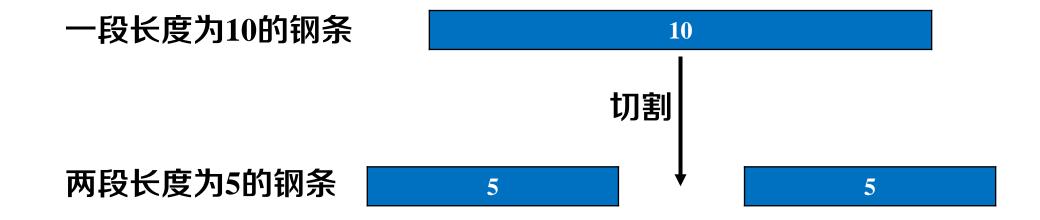
钢条长度	0	1	2	3	4	5	6	7	8	9	10
价格p	0	1	5	8	9	10	17	17	20	24	24





- 钢条切割
 - 现有一段长度为10的钢条,可以零成本将其切割为多段长度更小钢条

钢条长度	0	1	2	3	4	5	6	7	8	9	10
价格p	0	1	5	8	9	10	17	17	20	24	24





• 钢条切割

• 现有一段长度为10的钢条,可以零成本将其切割为多段长度更小钢条

钢条长度	0	1	2	3	4	5	6	7	8	9	10
价格p	0	1	5	8	9	10	17	17	20	24	24

切割	方案	总收益
方案1	{10}	24

10



• 钢条切割

• 现有一段长度为10的钢条,可以零成本将其切割为多段长度更小钢条

钢条长度	0	1	2	3	4	5	6	7	8	9	10
价格p	0	1	5	8	9	10	17	17	20	24	24

切割	方案	总收益
方案1	{10}	24
方案2	{5,5}	10+10=20

10	
-0	

5

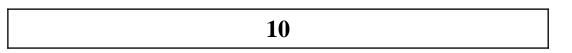


• 钢条切割

• 现有一段长度为10的钢条,可以零成本将其切割为多段长度更小钢条

钢条长度	0	1	2	3	4	5	6	7	8	9	10
价格p	0	1	5	8	9	10	17	17	20	24	24

切割	方案	总收益
方案1	{10}	24
方案2	{5,5}	10+10=20
方案3	{2,2,6}	5+5+17=27





|--|



• 钢条切割

• 现有一段长度为10的钢条,可以零成本将其切割为多段长度更小钢条

钢条长度	0	1	2	3	4	5	6	7	8	9	10
价格p	0	1	5	8	9	10	17	17	20	24	24

切割方案		总收益	
方案1	{10}	24	10
方案2	{5,5}	10+10=20	5 5
方案3	{2,2,6}	5+5+17=27	2 2 6

问题:怎样合理切割,使总收益最大?



• 形式化定义

钢条切割问题

Rod Cutting Problem

输入

钢条长度n



• 形式化定义

钢条切割问题

Rod Cutting Problem

输入

- 钢条长度n
- 价格表 $p_l(1 \le l \le n)$:表示长度为l的钢条价格



• 形式化定义

钢条切割问题

Rod Cutting Problem

输入

- 钢条长度n
- 价格表 $p_l(1 \le l \le n)$: 表示长度为l的钢条价格输出
- 求解一组切割方案 $T = \langle c_1, c_2, ..., c_m \rangle$, 令

$$\max \sum_{l=1}^{m} p_{c_l}$$

$$s. t. \sum_{l=1}^{m} c_l = n$$



• 形式化定义

钢条切割问题

Rod Cutting Problem

输入

- 钢条长度n
- 价格表 $p_l(1 \le l \le n)$:表示长度为l的钢条价格

输出

• 求解一组切割方案 $T = \langle c_1, c_2, ..., c_m \rangle$, 令

$$\max \sum_{l=1}^{m} p_{c_l}$$
 优化目标

$$s. t. \sum_{l=1}^{m} c_l = n$$



• 形式化定义

钢条切割问题

Rod Cutting Problem

输入

- 钢条长度n
- 价格表 $p_l(1 \le l \le n)$: 表示长度为l的钢条价格输出
- 求解一组切割方案 $T = \langle c_1, c_2, ..., c_m \rangle$, 令

$$\max \sum_{l=1}^{m} p_{c_l}$$
 优化目标

$$s.t.$$
 $\sum_{l=1}^{m} c_l = n$ 约束条件



- 假设至多切割1次
 - 枚举所有可能的切割位置



- 假设至多切割1次
 - 枚举所有可能的切割位置
 - o 不切: p[10]

10



- 假设至多切割1次
 - 枚举所有可能的切割位置

o 不切: p[10]

o 切割: p[i] + p[10 - i]

10

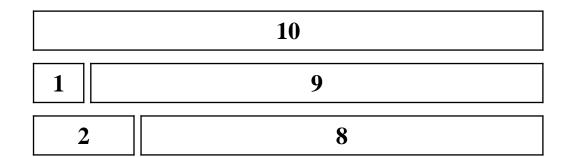
1 9



- 假设至多切割1次
 - 枚举所有可能的切割位置

o 不切: p[10]

o 切割: p[i] + p[10 - i]



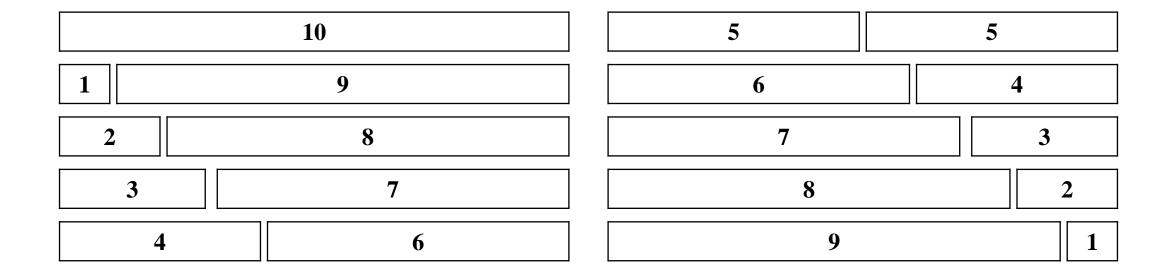


• 假设至多切割1次

• 枚举所有可能的切割位置

o 不切: p[10]

o 切割: p[i] + p[10 - i]





• 假设至多切割1次

• 枚举所有可能的切割位置

o 不切: p[10]

o 切割: p[i] + p[10 - i]

• 最大收益 $\max_{1 \le i \le 9} \{p[i] + p[10 - i], p[10]\}$

	10		5	5			
1 9			6		4		
2	8		7	3			
3	3		8	2			
4 6			9				1

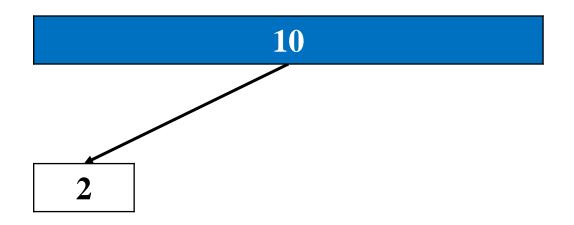


• 假设至多切割2次

10

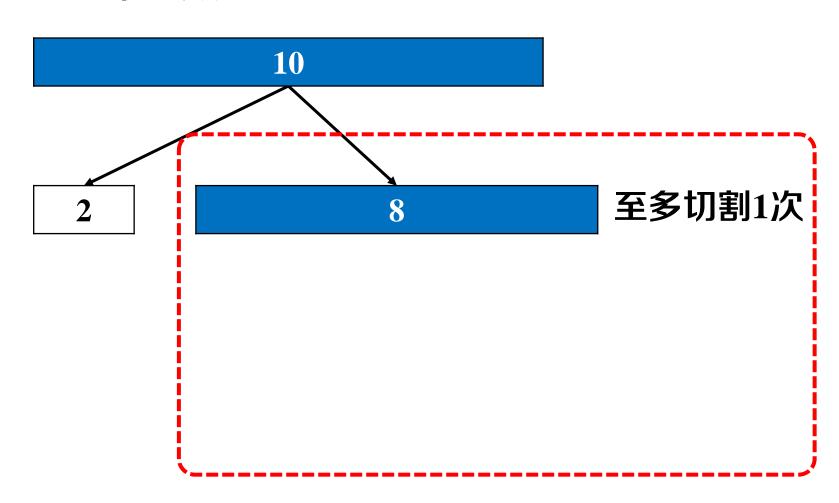


- 假设至多切割2次
 - 先将钢条切割出一段



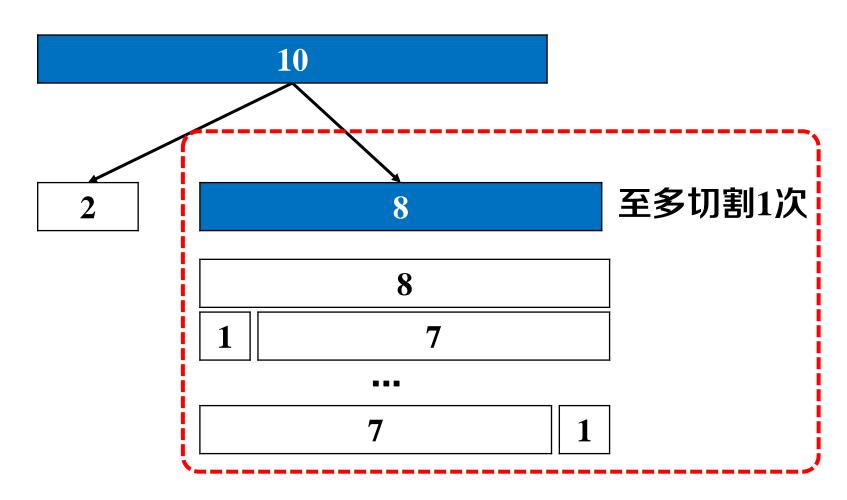


- 假设至多切割2次
 - 先将钢条切割出一段
 - 在剩余钢条中继续切割





- 假设至多切割2次
 - 先将钢条切割出一段
 - 在剩余钢条中继续切割



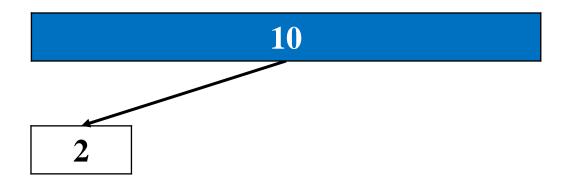


• 原始问题不限制切割次数

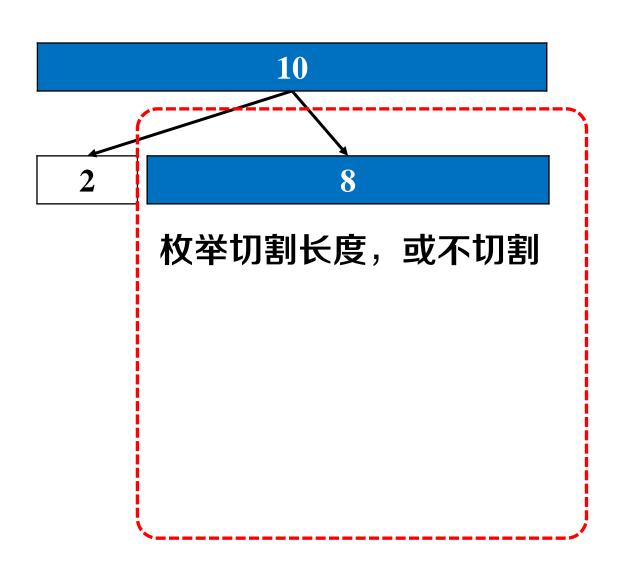
10

枚举切割长度,或不切割

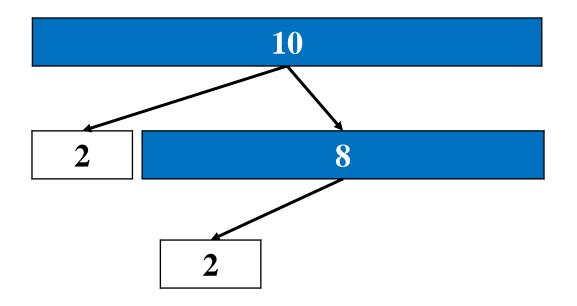




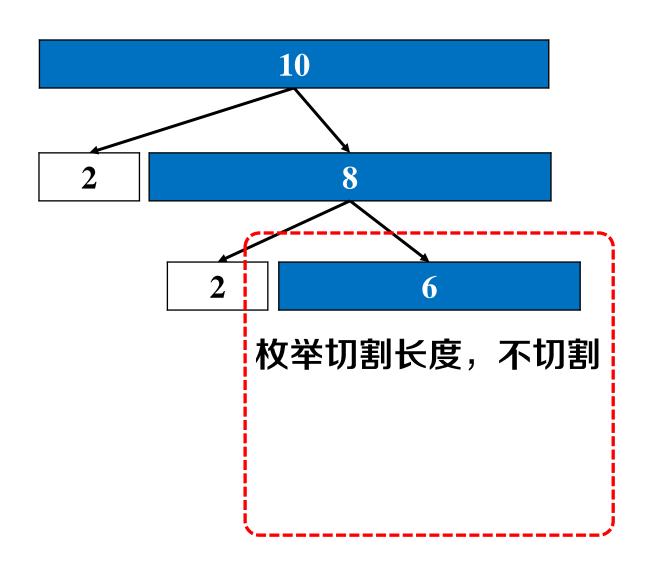




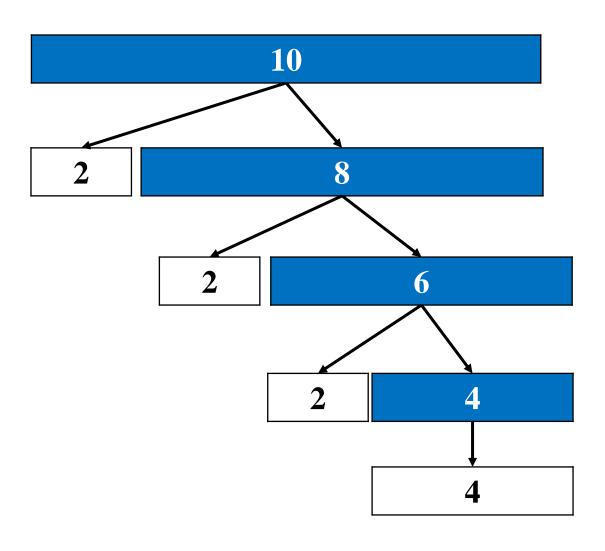






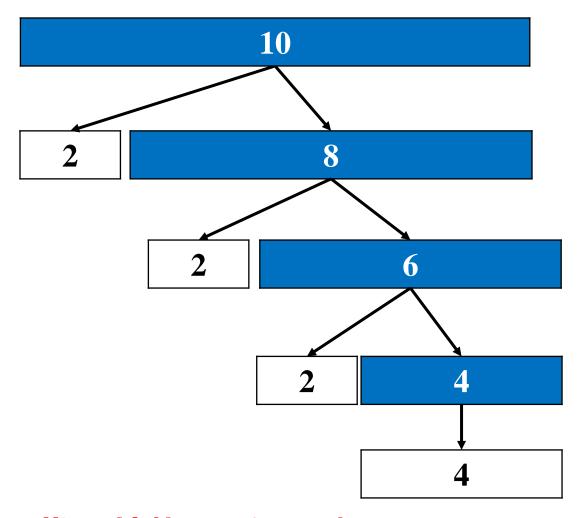








• 原始问题不限制切割次数



• 可能存在最优子结构和重叠子问题

问题结构分析



- 给出问题表示
 - C[j]: 切割长度为j的钢条可得最大总收益

C[j]

j

问题结构分析



递推关系建立



自底向上计算



最优方案追踪

问题结构分析



- 给出问题表示
 - C[j]: 切割长度为j的钢条可得最大总收益

C[j]

j

- 明确原始问题
 - C[n]: 切割长度为n的钢条可得最大总收益

问题结构分析



递推关系建立



自底向上计算



最优方案追踪

递推关系建立:分析最优(子)结构



C[10]

10

问题结构分析



递推关系建立



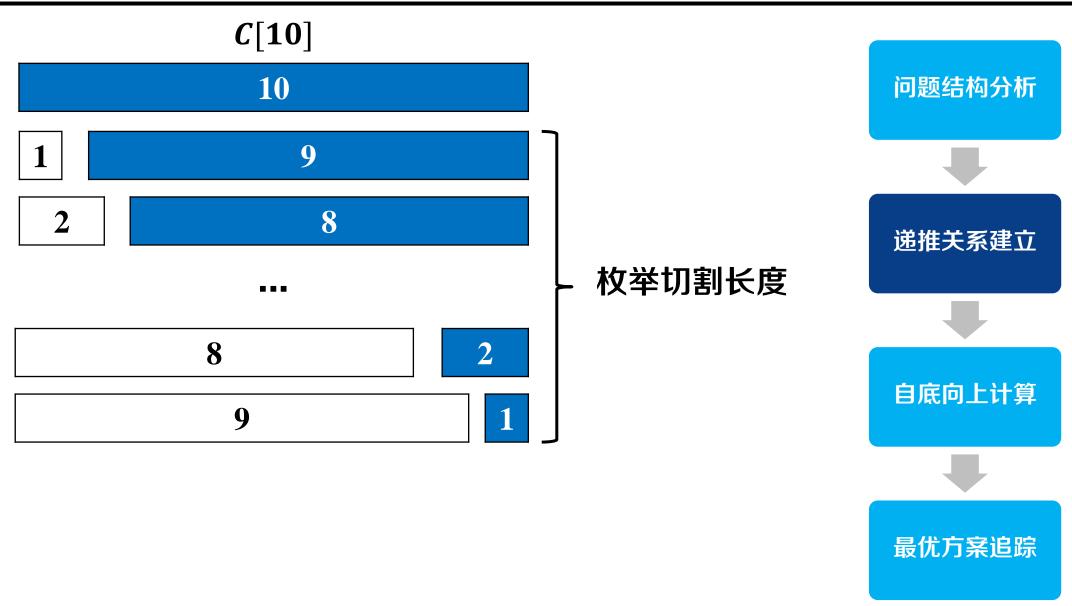
自底向上计算



最优方案追踪

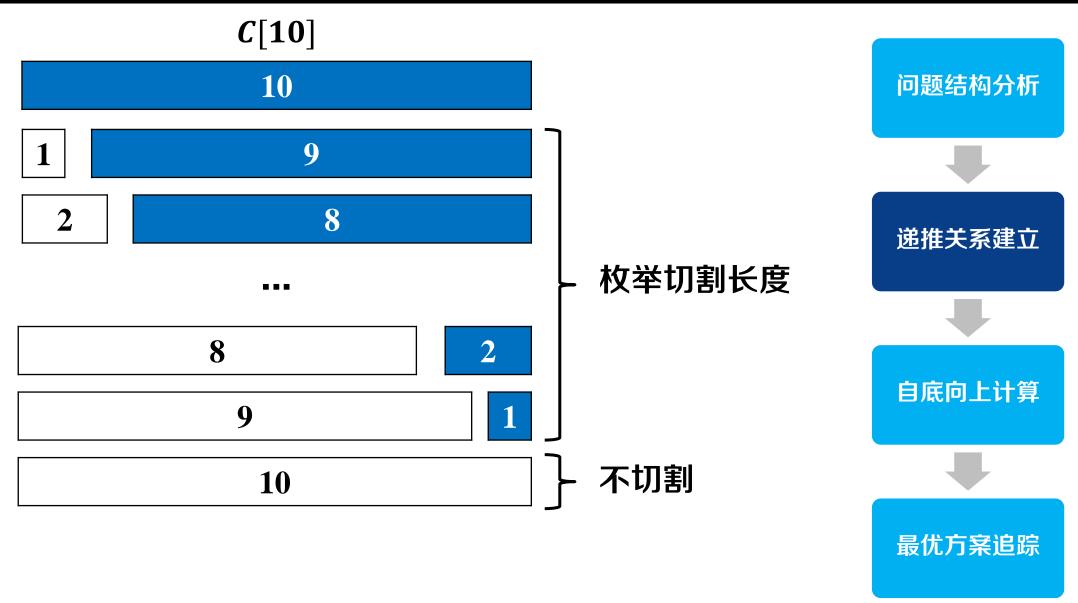
递推关系建立:分析最优(子)结构



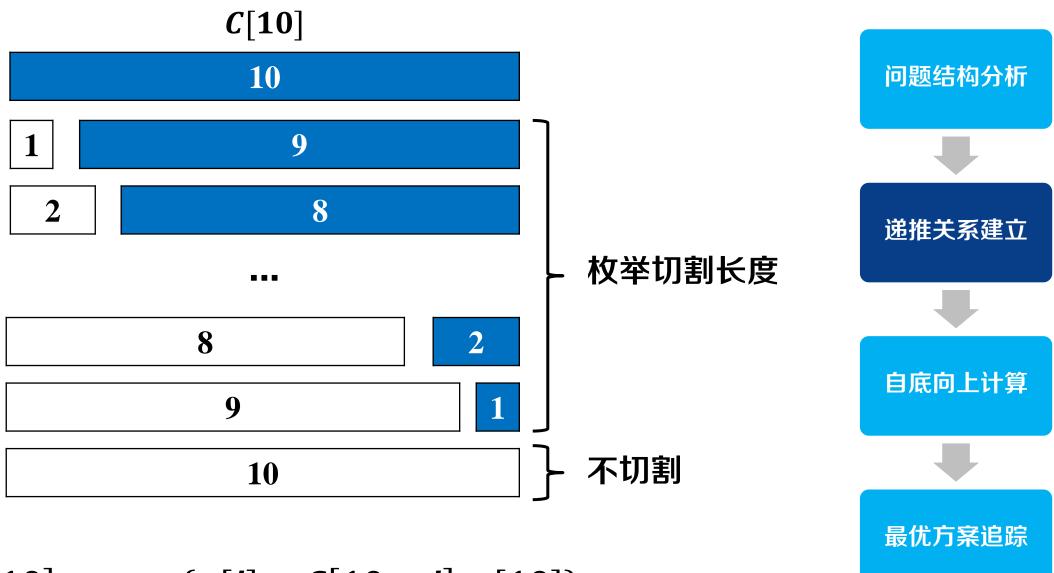


递推关系建立:分析最优(子)结构



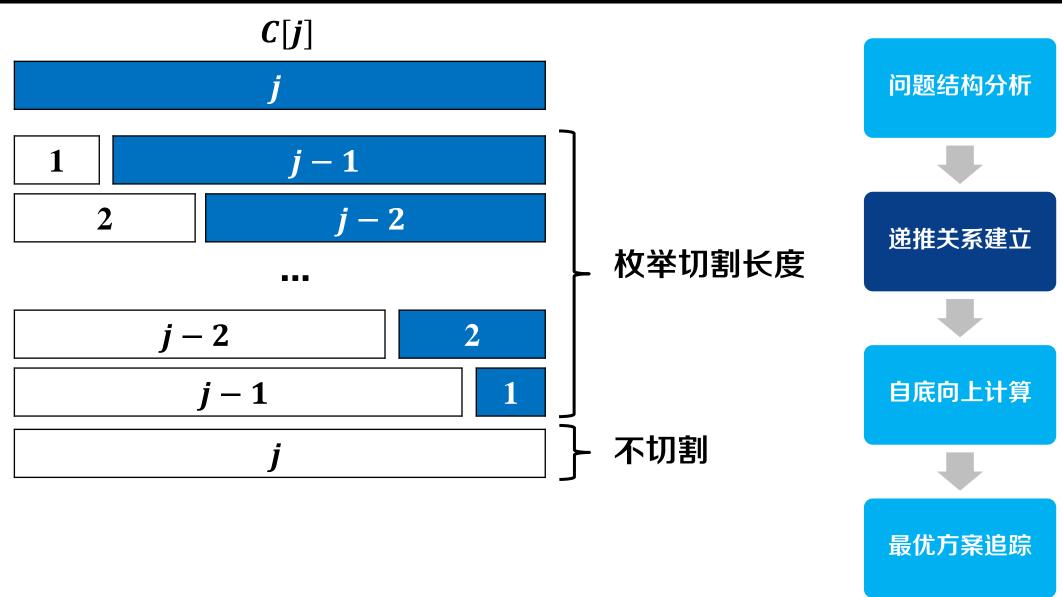




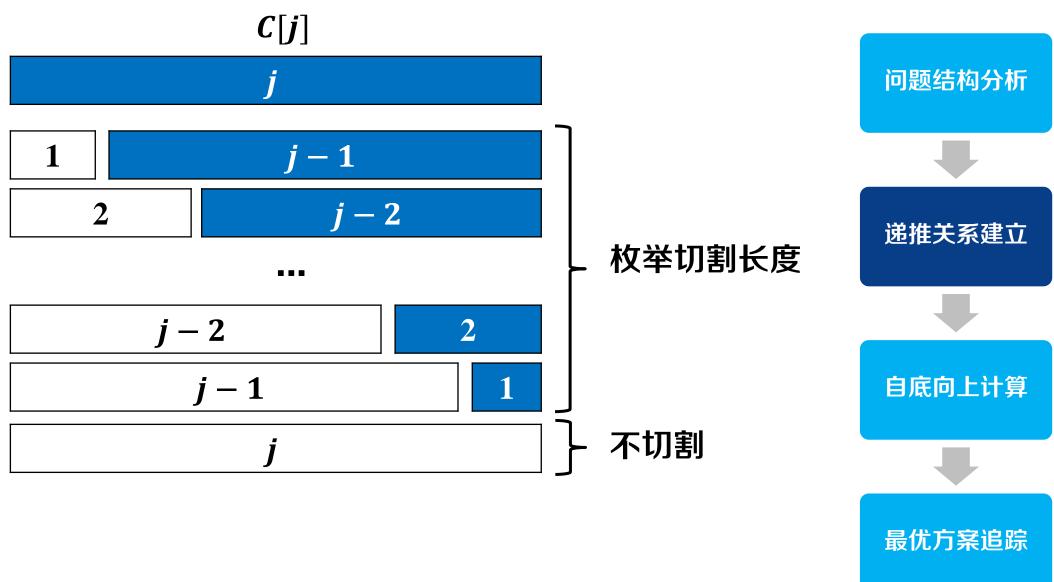


• $C[10] = \max_{1 \le i \le 9} \{p[i] + C[10 - i], p[10]\}$



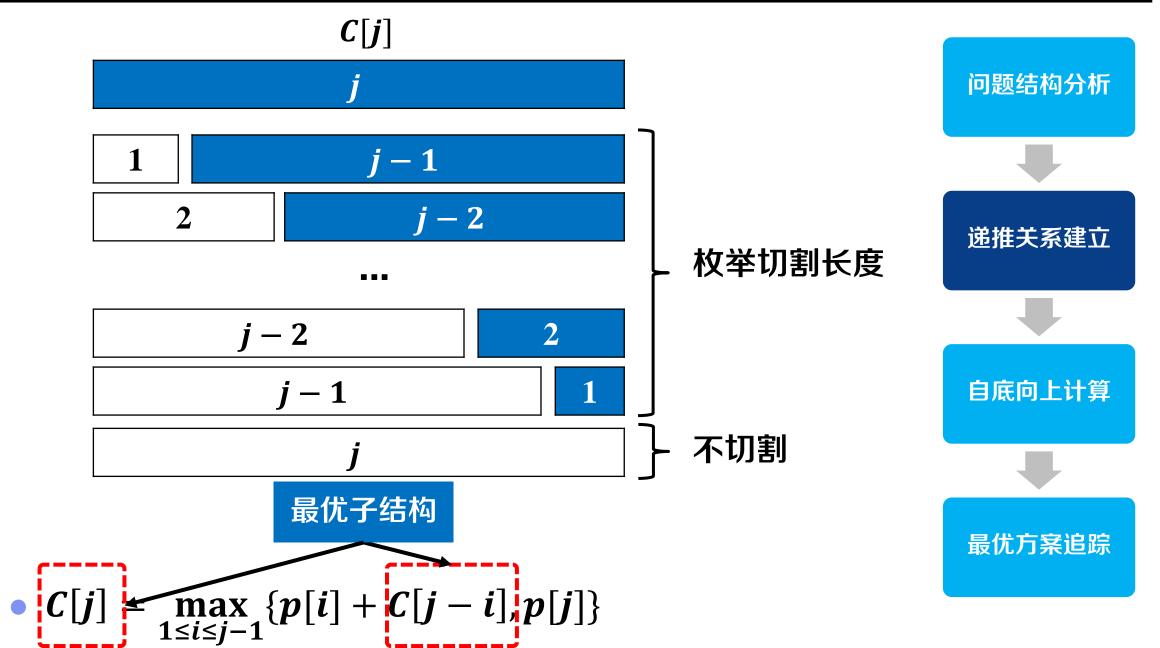






• $C[j] = \max_{1 \le i \le j-1} \{p[i] + C[j-i], p[j]\}$







• 对于每个钢条长度j

•
$$C[j] = \max_{1 \le i \le j-1} \{p[i] + C[j-i], p[j]\}$$

问题结构分析



递推关系建立



自底向上计算





- 对于每个钢条长度j
 - $C[j] = \max_{1 \le i \le j-1} \{p[i] + C[j-i], p[j]\}$

问题结构分析



递推关系建立



自底向上计算



最优方案追踪

p[j]

C[j]

max

C[j-1]	C[j-2]	C[j-3]	•••	C [1]
+	+	+	+	+
p [1]	p [2]	p [3]	•••	p[j-1]



• 对于每个钢条长度j

•
$$C[j] = \max_{1 \le i \le j-1} \{ p[i] + C[j-i], p[j] \}$$





递推关系建立



自底向上计算



最优方案追踪

C[j]

max

	C [1]	•••	C[j-3]	芰 [j-2]	枚举切割长原
	+	+	+	+	+
p[j]	p[j-1]	• • •	p [3]	p [2]	<i>p</i> [1]

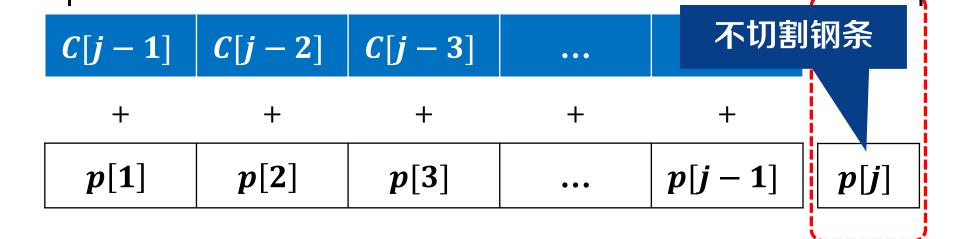


- 对于每个钢条长度j
 - $C[j] = \max_{1 \le i \le j-1} \{p[i] + C[j-i], p[j]\}$



C[j]

max



自底向上计算:确定计算顺序



已知钢条价格

p[] p[1] p[2] ... p[n]

- 初始化
 - C[0] = 0 切割长度为0的钢条,总收益为0



问题结构分析



递推关系建立



自底向上计算



自底向上计算:确定计算顺序



已知钢条价格

p[] p[1] p[2] ... p[n]

- 初始化
 - C[0] = 0 切割长度为0的钢条,总收益为0
- 递推公式
 - $C[j] = \max_{1 \le i \le j-1} \{p[i] + C[j-i], p[j]\}$

自底向上计算

	0	1	2	•••	\boldsymbol{n}
C []	0 —				→ ★

问题结构分析



递推关系建立



自底向上计算





已知钢条价格

p[] p[1] p[2] ... p[n]

问题结构分析



递推关系建立



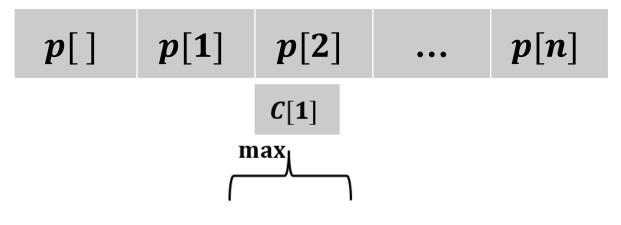
自底向上计算

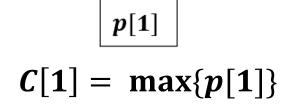


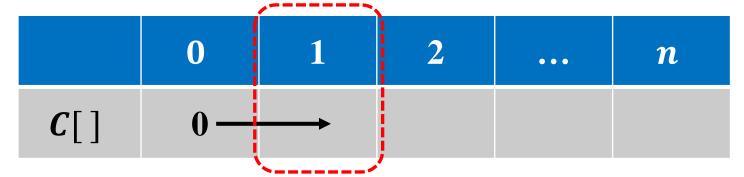
最优方案追踪



已知钢条价格







问题结构分析



递推关系建立

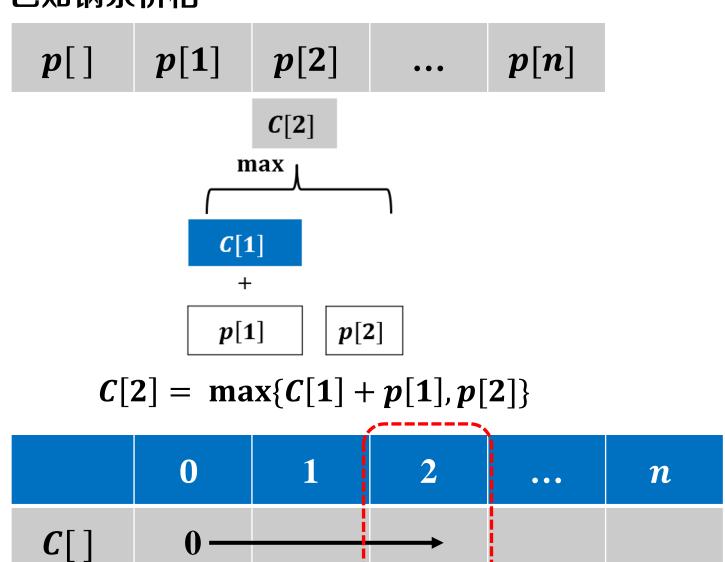


自底向上计算





已知钢条价格



问题结构分析



递推关系建立

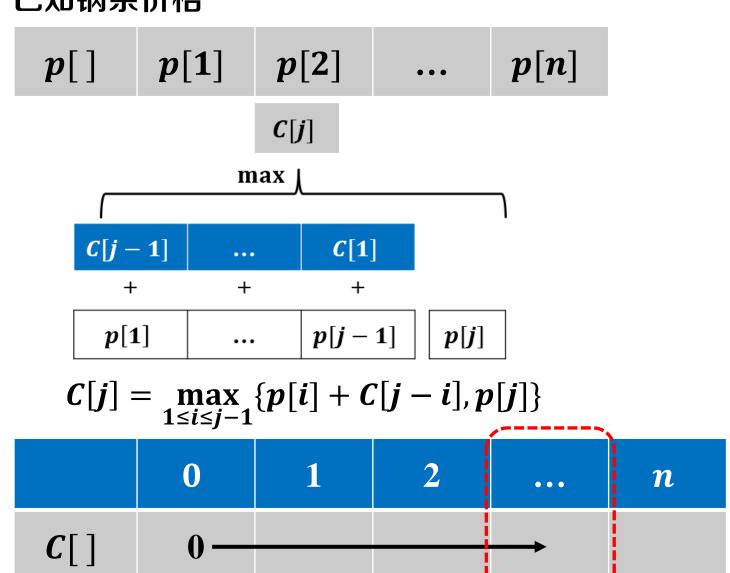


自底向上计算





已知钢条价格



问题结构分析



递推关系建立

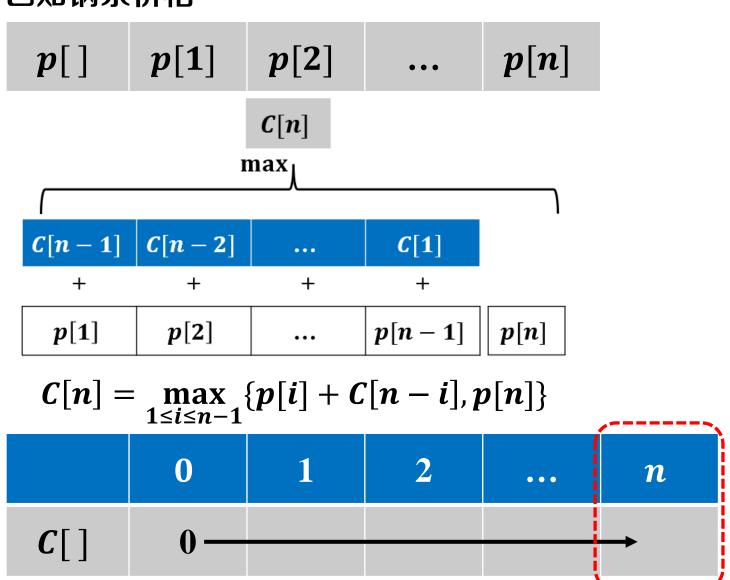


自底向上计算





已知钢条价格



问题结构分析



递推关系建立

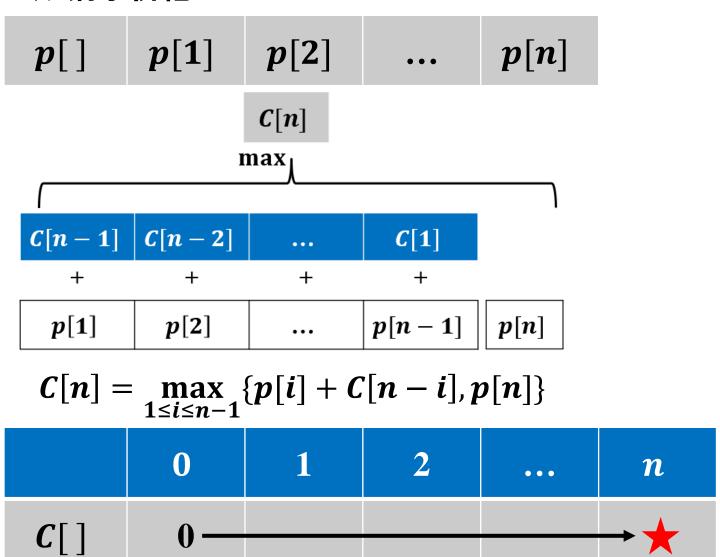


自底向上计算





已知钢条价格



问题结构分析



递推关系建立



自底向上计算



最优方案追踪: 记录决策过程



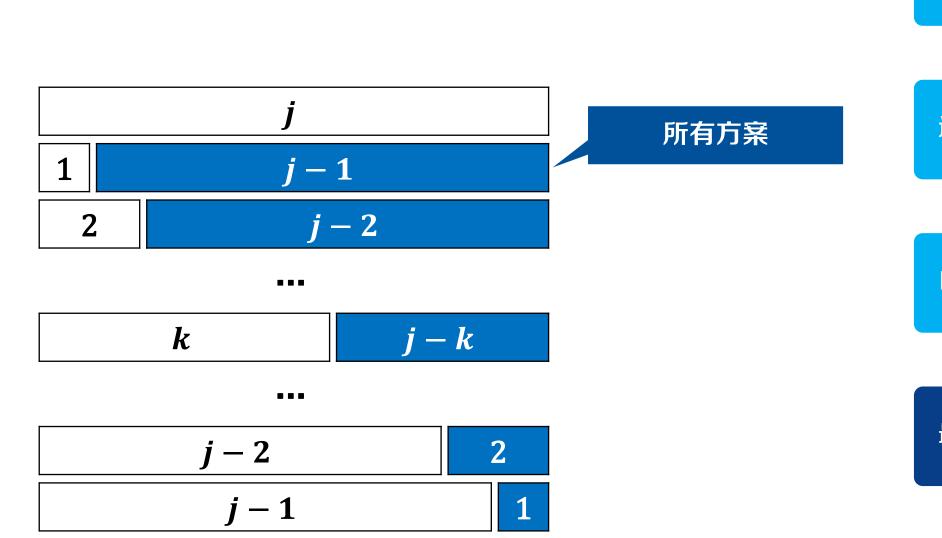
构造追踪数组rec[1..n]



最优方案追踪: 记录决策过程



构造追踪数组rec[1..n]



问题结构分析



递推关系建立



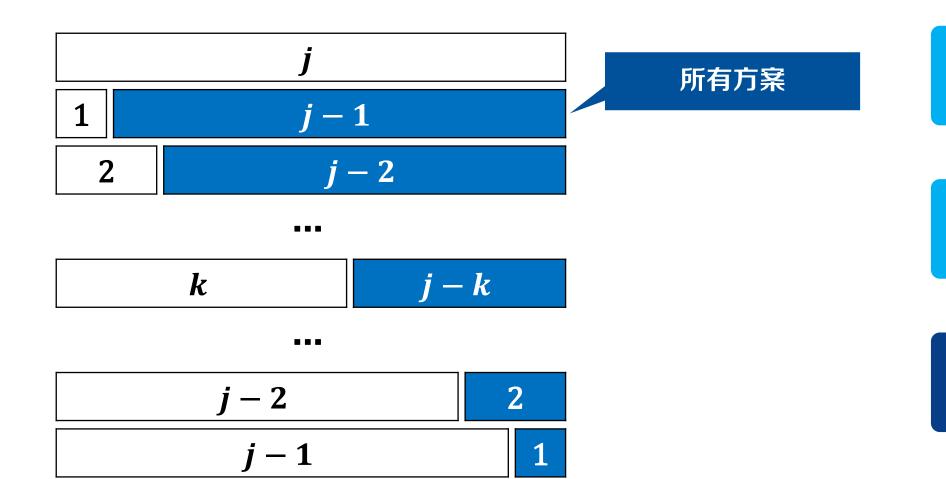
自底向上计算



最优方案追踪:记录决策过程



- 构造追踪数组rec[1..n]
- rec[j]: 记录长度为j钢条的最优切割方案



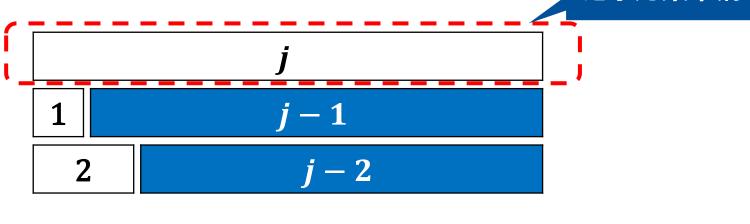
自底向上计算

最优方案追踪:记录决策过程



- 构造追踪数组rec[1..n]
- rec[j]: 记录长度为j钢条的最优切割方案
 - 不切: rec[j] = j

记录方案中钢条长度





•••

$$\begin{array}{c|c} j-2 & 2 \\ \hline j-1 & 1 \end{array}$$

问题结构分析



递推关系建立



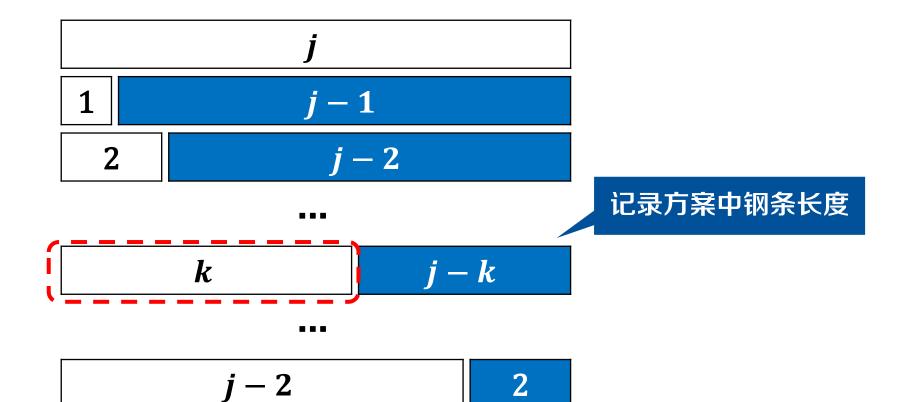
自底向上计算



最优方案追踪:记录决策过程



- 构造追踪数组rec[1..n]
- rec[j]: 记录长度为j钢条的最优切割方案
 - 不切: rec[j] = j 切割: rec[j] = k



问题结构分析



递推关系建立

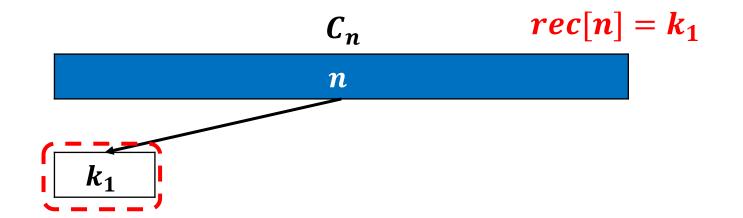


自底向上计算





- 根据追踪数组,递归输出方案
 - 輸出长度为k₁的钢条



问题结构分析



递推关系建立

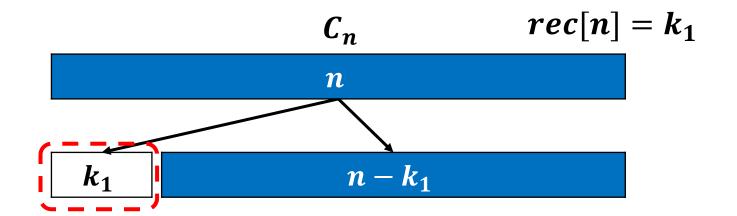


自底向上计算





- 根据追踪数组,递归输出方案
 - 输出长度为k₁的钢条



问题结构分析



递推关系建立

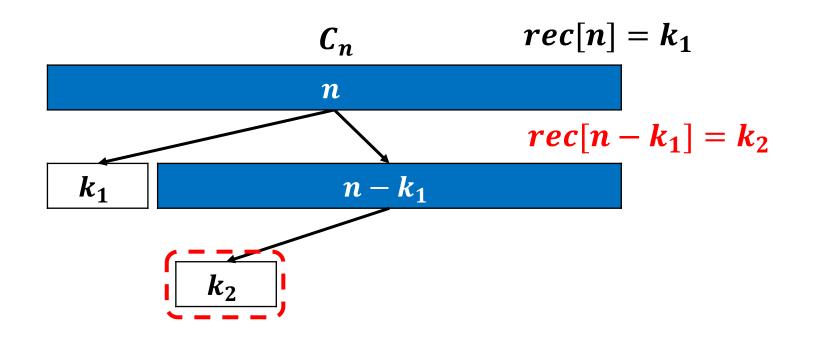


自底向上计算





- 根据追踪数组,递归输出方案
 - 输出长度为 k_2 的钢条



问题结构分析



递推关系建立

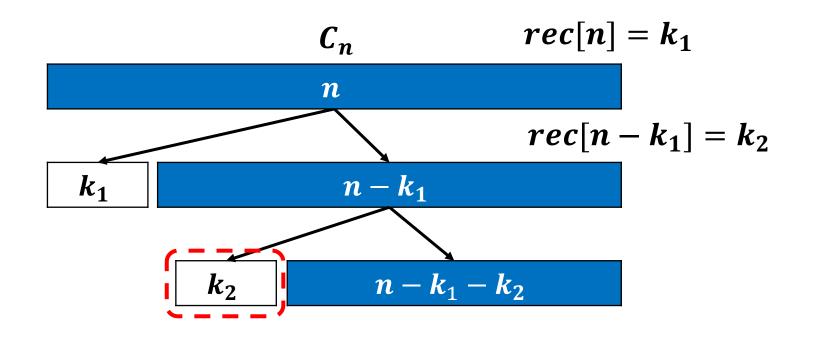


自底向上计算





- 根据追踪数组,递归输出方案
 - 输出长度为 k_2 的钢条



问题结构分析



递推关系建立

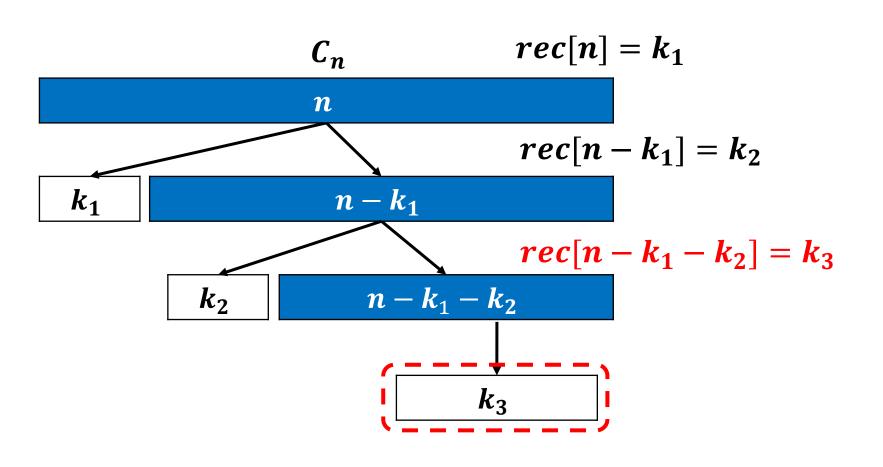


自底向上计算





- 根据追踪数组,递归输出方案
 - 输出长度为k₃的钢条



问题结构分析



递推关系建立

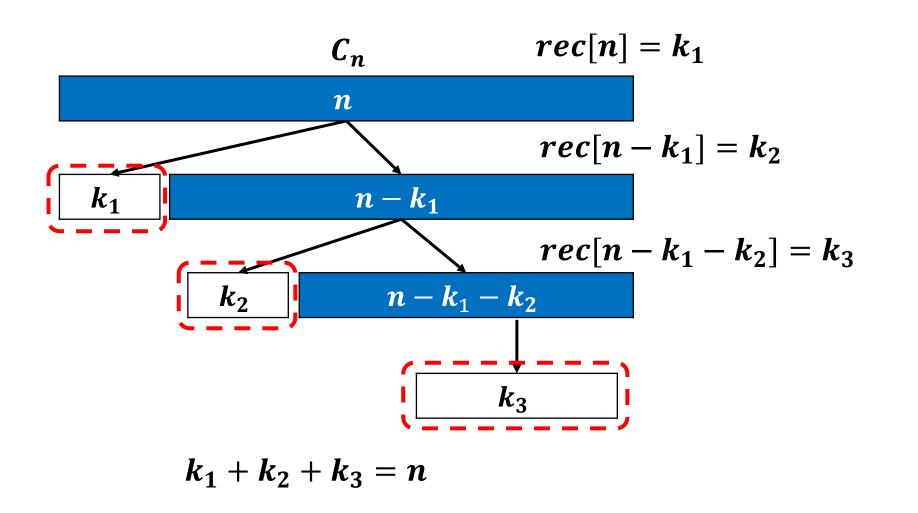


自底向上计算





- 根据追踪数组,递归输出方案
 - 递归出口:输出的钢条总长度已达*n*



问题结构分析



递推关系建立



自底向上计算





n = 10

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	Ř	刀始化								
rec	0	1	2	3	4	5	6	7	8	9	10



$$n = 10$$

	1									
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 1$$

$$\begin{array}{c|c} i & 1 \\ \hline & p[j] = p[1] \\ \hline \end{array}$$

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0										
rec	0	1	2	3	4	5	6	7	8	9	10



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 1$$

$$\max\{p[i] + C[j-i], p[j]\}$$

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0										
rec	0	1	2	3	4	5	6	7	8	9	10



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 1$$

$$\max\{p[i] + C[j-i], p[j]\}$$

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1									
rec	0	1	2	3	4	5	6	7	8	9	10
		1									



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 2$$

$$\begin{array}{c|c} i & 1 \\ \hline \end{array} p[1] + C[1]$$

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1									
rec	0	1	2	3	4	5	6	7	8	9	10
		1									



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i										

$$j = 2$$

i	1	2	p[j] = p[2]
	2	5	

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1									
rec	0	1	2	3	4	5	6	7	8	9	10
		1									



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 2$$

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1									
rec	0	1	2	3	4	5	6	7	8	9	10
		1									



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 2$$

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5								
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2								



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i										

$$j = 3$$

i	1	p[1] + C[[2]
	6		

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5								
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2								



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i										

$$j = 3$$

i	1	2	p[2] + C[1]
	6	6	

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5								
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2								



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 3$$

i	1	2	3	p[j] = p[3]
	6	6	8	

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5								
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2								



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 3$$

i	1	2	3	
	6	6	8	

$$\max\{p[i] + C[j-i], p[j]\}$$

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5								
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2								



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 3$$

i	1	2	3	max
	6	6	8	

$$\max\{p[i] + C[j-i], p[j]\}$$

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8							
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3							



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i										

$$j = 4$$

$$\begin{array}{c|c} i & 1 \\ \hline 9 & \end{array}$$

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8							
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3							



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 4$$

i	1	2	p[2] + C[2]
	9	10	

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8							
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3							



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 4$$

i	1	2	3	p[3] + C[1]
	9	10	9	

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	, 5	8							
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3							



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 4$$

i	1	2	3	4	p[j]
	9	10	9	9	

$$p[j] = p[4]$$

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8							
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3							



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 4 \qquad \max\{p[i] + C[j-i], p[j]\}$$

i	1	2	3	4
	9	10	9	9

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8							
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3							



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 4 \qquad \max\{p[i] + C[j-i], p[j]\}$$

i	1	2	3	4
	9	10	9	9

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10						
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2						



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	_10_	17	17	20	24	24

$$j = 5$$

i	1	2	3	4	5
	11	13	13	10	10

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10						
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2						



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 5 \qquad \max\{p[i] + C[j-i], p[j]\}$$

i	1	2	3	4	5
	11	13	13	10	10

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10						
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2						



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i										

$$j = 5 \qquad \max\{p[i] + C[j-i], p[j]\}$$

i	1	2	3	4	5
	11	13	13	10	10

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10	13					
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2	2					



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 6$$

i	1	2	3	4	5	6
	14	15	16	14	11	17

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10	13					
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2	2					



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 6$$

i	1	2	3	4	5	6
					11	

 $\max\{p[i] + C[j-i], \frac{p[j]}{p[j]}\}$

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10	13					
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2	2					



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 6$$

i	1	2	3	4	5	6
	14	15	16	14	11	17

 $\max\{p[i] + C[j-i], p[j]\}$

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10	13	17				
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2	2	6				



n = 10

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

j = 7

i	1	2	3	4	5	6	7
	18	18	18	17	15	18	17

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10	13	17				
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2	2	6				



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 7 \qquad \max\{p[i] + C[j-i], p[j]\}$$

i	1	2	3	4	5	6	7
	18	18	18	17	15	18	17

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10	13	17				
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2	2	6				



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 7 \qquad \max\{p[i] + C[j-i], p[j]\}$$

i	1	2	3	4	5	6	7
	18	18	18	17	15	18	17

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10	13	17	18			
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2	2	6	1			



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 8$$

i	1	2	3	4	5	6	7	8
	19	22	21	19	18	22	18	20

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10	13	17	18			
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2	2	6	1			



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 8 \qquad \max\{p[i] + C[j-i], p[j]\}$$

i	1	2	3	4	5	6	7	8
	19	22	21	19	18	22	18	20

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10	13	17	18			
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2	2	6	1			



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 8 \qquad \max\{p[i] + C[j-i], p[j]\}$$

i	1	2	3	4	5	6	7	8
	19	22	21	19	18	22	18	20

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10	13	17	18	22		
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	2	•	2	6	1	2		



n = 10

i	1	2	3	4	5	6	7	8	9	10
p_i	1_	5	8	9	10	17	17	20	24	24

j = 9

i	1	2	3	4	5	6	7	8	9
	23	23	25	22	20	25	22	21	24

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0		5	8					22		
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2	2	6	1	2		



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 9 \qquad \max\{p[i] + C[j-i], p[j]\}$$

i	1	2	3	4	5	6	7	8	9
	23	23	25	22	20	25	22	21	24

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10	13	17	18	22		
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2	2	6	1	2		



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i										

$$j = 9 \qquad \max\{p[i] + C[j-i], p[j]\}$$

i	1	2	3	4	5	6	7	8	9
	23	23	25	22	20	25	22	21	24

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10	13	17	18	22	25	
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2	2	6	1	2	3	



n = 10

i	1	2	3	4	5	6	7	8	9	10
p_i	1_	5	8	9_	10	_17_	17	20	24	24

j = 10

i	1	2	3	4	5	6	7	8	9	10
	26	27	26	26	23	27	25	25	25	24

i							7			10
C[i]							18			
rec							7			10
	1	2	3	2	2	6	1	2	3	



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 10 \qquad \max\{p[i] + C[j-i], p[j]\}$$

i	1	2	3	4	5	6	7	8	9	10
	26	27	26	26	23	27	25	25	25	24

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10	13	17	18	22	25	
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2	2	6	1	2	3	



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 10 \qquad \max\{p[i] + C[j-i], p[j]\}$$

i	1	2	3	4	5	6	7	8	9	10
	26	27	26	26	23	27	25	25	25	24

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10	13	17	18	22	25	27
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2	2	6	1	2	3	2



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 10$$

i	1	2	3	4	5	6	7	8	9	10
	26	27	26	26	23	27	25	25	25	24

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10	13	17	18	22	25	27
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2	2	6	1	2	3	2



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 10$$

i	1	2	3	4	5	6	7	8	9	10
	26	27	26	26	23	27	25	25	25	24

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10	13	17	18	22	25	27
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2	2	6	1	2	3	2

最大收益= C[10] = 27



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 10$$

i	1	2	3	4	5	6	7	8	9	10
	26	27	26	26	23	27	25	25	25	24

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10	13	17	18	22	25	27
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2	2	6	1	2	3	2

最大收益= *C*[10] = 27 切割方案= { 2,



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 10$$

i	1	2	3	4	5	6	7	8	9	10
	26	27	26	26	23	27	25	25	25	24

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10	13	17	18	22	25	27
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2	2	6	1	2	3	2

最大收益= *C*[10] = 27 切割方案= { 2, 2



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 10$$

i	1	2	3	4	5	6	7	8	9	10
	26	27	26	26	23	27	25	25	25	24

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10	13	17	18	22	25	27
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2	2	6	1	2	3	2

最大收益= *C*[10] = 27 切割方案= { 2, 2, 6



$$n = 10$$

i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	24	24

$$j = 10$$

i	1	2	3	4	5	6	7	8	9	10
	26	27	26	26	23	27	25	25	25	24

i	0	1	2	3	4	5	6	7	8	9	10
C[i]	0	1	5	8	10	13	17	18	22	25	27
rec	0	1	2	3	4	5	6	7	8	9	10
		1	2	3	2	2	6	1	2	3	2

最大收益= *C*[10] = 27 切割方案= { 2, 2, 6}

伪代码



```
输入: 钢条价格表p[1..n],钢条长度n
输出: 最大收益C[n],钢条切割方案
//初始化
新建一维数组C[0..n], rec[0..n]
                                                        初始化
C[0] \leftarrow 0
#动态规划
for j \leftarrow 1 to n do
    q \leftarrow p[j]
    rec[j] \leftarrow j
    for i \leftarrow 1 to j - 1 do
       if q < p[i] + C[j-i] then
         q \leftarrow p[i] + C[j-i]
         rec[j] \leftarrow i
       end
    end
    C[j] \leftarrow q
end
```



```
输入: 钢条价格表p[1..n],钢条长度n
输出: 最大收益C[n],钢条切割方案
//初始化
新建一维数组C[0..n], rec[0..n]
C[0] \leftarrow 0
#动态规划
                                                  依次计算子问题
for j \leftarrow 1 to n do
\overline{q} \leftarrow \overline{p}[j]
    rec[j] \leftarrow j
    for i \leftarrow 1 to j - 1 do
        if q < p[i] + C[j-i] then
         q \leftarrow p[i] + C[j-i]
          rec[j] \leftarrow i
        end
    end
    C[j] \leftarrow q
end
```



```
输入: 钢条价格表p[1..n],钢条长度n
输出: 最大收益C[n],钢条切割方案
//初始化
新建一维数组C[0..n], rec[0..n]
C[0] \leftarrow 0
//动态规划
for j \leftarrow 1 to n do
   q \leftarrow p[j]
                                                 不切割钢条
   for i \leftarrow 1 to j - 1 do
       if q < p[i] + C[j-i] then
         q \leftarrow p[i] + C[j-i]
         rec[j] \leftarrow i
       end
   end
   C[j] \leftarrow q
end
```



```
输入: 钢条价格表p[1..n],钢条长度n
输出: 最大收益C[n],钢条切割方案
//初始化
新建一维数组C[0..n], rec[0..n]
C[0] \leftarrow 0
//动态规划
for j \leftarrow 1 to n do
   q \leftarrow p[j]
  for i \leftarrow 1 to j - 1 do
                                                枚举切割长度
 - - -if q < p[i] + C[j-i] + then - -
       q \leftarrow p[i] + C[j-i]
         rec[j] \leftarrow i
       end
   end
   C[j] \leftarrow q
end
```



```
输入: 钢条价格表p[1..n],钢条长度n
输出: 最大收益C[n],钢条切割方案
//初始化
新建一维数组C[0..n], rec[0..n]
C[0] \leftarrow 0
//动态规划
for j \leftarrow 1 to n do
   q \leftarrow p[j]
   rec[j] \leftarrow j
   for i \leftarrow 1 to j - 1 do
      if q < p[i] + C[j-i] then
                                                记录价格和决策
         q \leftarrow p[i] + C[j-i]
          rec[j] \leftarrow i
       end
   end
   C[j] \leftarrow q
end
```

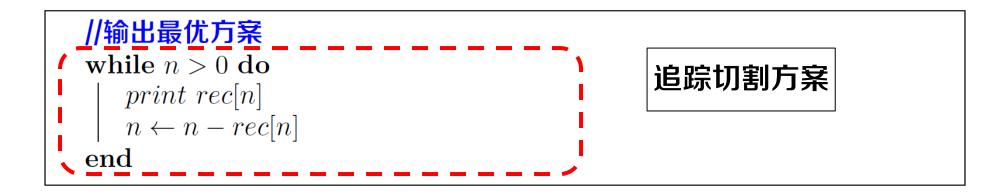
伪代码



```
输入: 钢条价格表p[1..n],钢条长度n
输出: 最大收益C[n],钢条切割方案
//初始化
新建一维数组C[0..n], rec[0..n]
C[0] \leftarrow 0
//动态规划
for j \leftarrow 1 to n do
   q \leftarrow p[j]
   rec[j] \leftarrow j
   for i \leftarrow 1 to j - 1 do
       if q < p[i] + C[j-i] then
        q \leftarrow p[i] + C[j-i]
        rec[j] \leftarrow i
       end
                                               保存子问题的解
end
```

伪代码





时间复杂度分析



• RodCutting(p, n)

```
输入: 钢条价格表p[1..n],钢条长度n
输出: 最大收益C[n],钢条切割方案
//初始化
新建一维数组C[0..n], rec[0..n]
C[0] \leftarrow 0
<u>//动态规划</u>_
for j \leftarrow 1 to n do
    q \leftarrow p[j]
   rec[j] \leftarrow j
  for i \leftarrow 1 \ to \ j-1 \ \mathbf{do}
   if q < p[i] + C[j-i] then
        q \leftarrow p[i] + C[j-i]
          rec[j] \leftarrow i
        end
    end
    C[j] \leftarrow q
end
```

时间复杂度: $O(n^2)$