# Design and Analysis of Algorithms Part II: Dynamic Programming Lecture 15: Rod-Cutting

盛浩

shenghao@buaa.edu.cn

北京航空航天大学计算机学院

北航《算法设计与分析》

## 动态规划篇概述



- 在算法课程第二部分"动态规划"主题中,我们将主要聚焦于如下 经典问题:
  - 0-1 Knapsack (0-1背包问题)
  - Maximum Contiguous Subarray II (最大连续子数组 II)
  - Longest Common Subsequences (最长公共子序列)
  - Longest Common Substrings (最长公共子串)
  - Minimum Edit Distance (最小编辑距离)
  - Rod-Cutting (钢条切割)
  - Chain Matrix Multiplication (矩阵链乘法)

## 动态规划篇概述



- 在算法课程第二部分"动态规划"主题中,我们将主要聚焦于如下 经典问题:
  - 0-1 Knapsack (0-1背包问题)
  - Maximum Contiguous Subarray II (最大连续子数组 II)
  - Longest Common Subsequences (最长公共子序列)
  - Longest Common Substrings (最长公共子串)
  - Minimum Edit Distance (最小编辑距离)
  - Rod-Cutting (钢条切割)
  - Chain Matrix Multiplication (矩阵链乘法)



## • 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 <b>p</b>	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 <b>p</b>	0	1	5	8	9	10	17	17	20	24	24

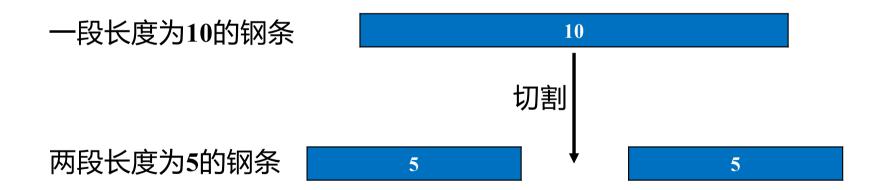
一段长度为10的钢条



## • 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 <b>p</b>	0	1	5	8	9	10	17	17	20	24	24

切割	方案	总收益
方案1	<b>{10}</b>	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 <b>p</b>	0	1	5	8	9	10	17	17	20	24	24

切割	方案	总收益
方案1	<b>{10}</b>	24
方案2	<b>{5,5}</b>	10+10=20

4.0	
10	
10	

<b>5</b>	5
3	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 <b>p</b>	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
方案3	{2,2,6}	5+5+17=27	2 2 6



## • 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 <b>p</b>	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
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?

# 问题定义



• 形式化定义

## 钢条切割问题

**Rod Cutting Problem** 

#### 输入

钢条长度n

# 问题定义



## • 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

## 问题定义



# • 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 $T=< c_1, c_2, ..., c_m >$ ,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 $T=<c_1,c_2,...,c_m>$ ,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

10

不切: p[10]



- 假设至多切割1次
  - 枚举所有可能的切割位置

。 不切: p[10]

。 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

。 不切: p[10]

。 切割:p[i] + p[10 − i]

10			
1	9		
2	8		



- 假设至多切割1次
  - 枚举所有可能的切割位置

。 不切: p[10]

。 切割:p[i] + p[10 - i]

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



- 假设至多切割1次
  - 枚举所有可能的切割位置

。 不切:p[10]

。 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		8	2
4		9	1

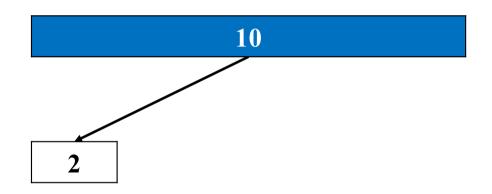


• 假设至多切割2次

10

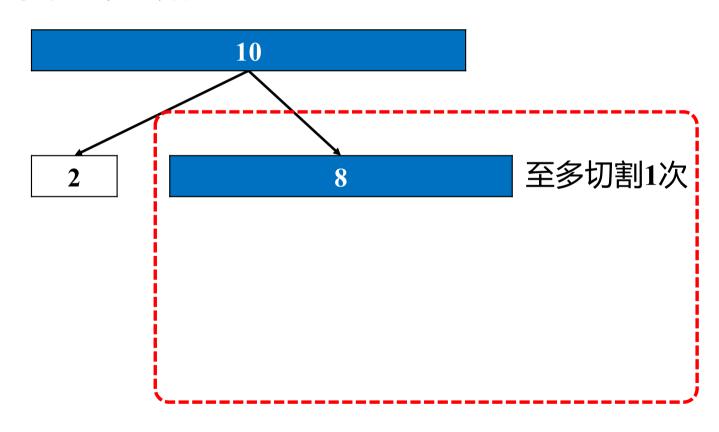


- 假设至多切割2次
  - 先将钢条切割出一段



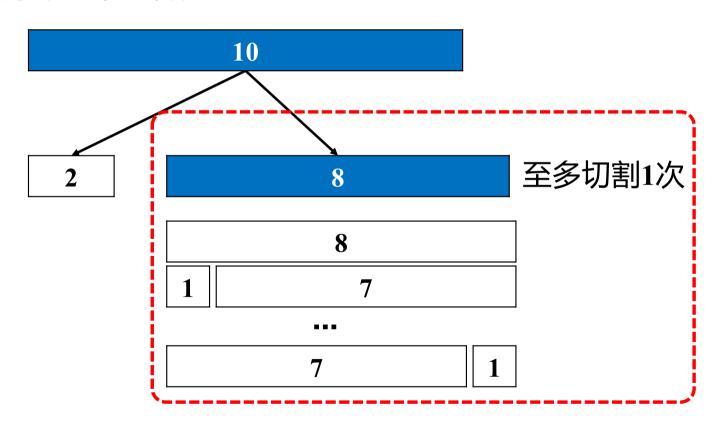


- 假设至多切割2次
  - 先将钢条切割出一段
  - 在剩余钢条中继续切割





- 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



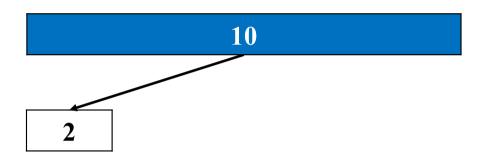


• 原始问题不限制切割次数

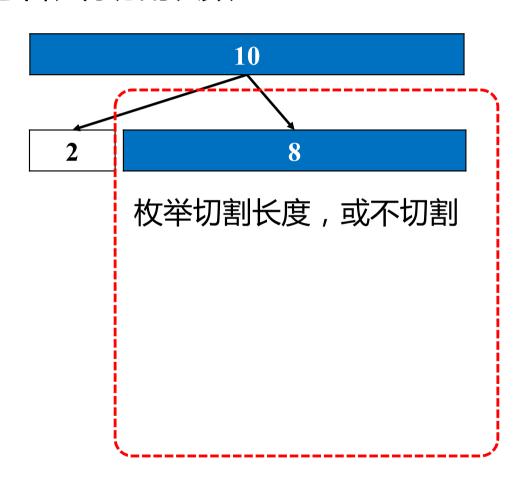
10

枚举切割长度,或不切割

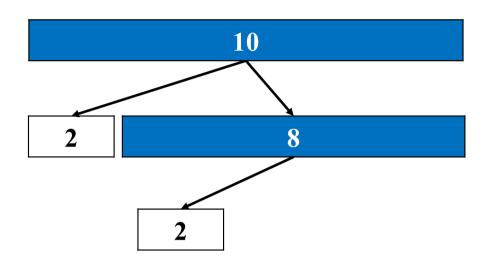




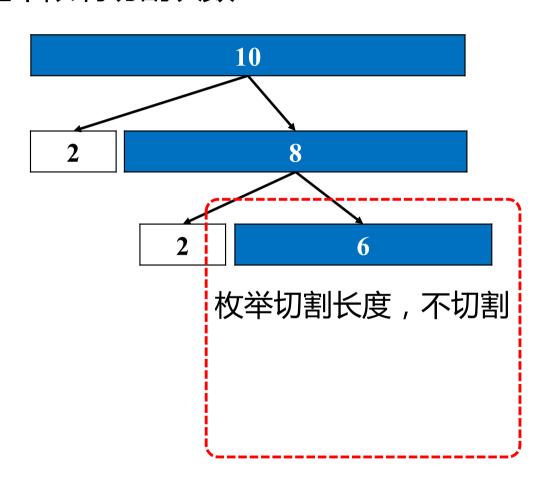




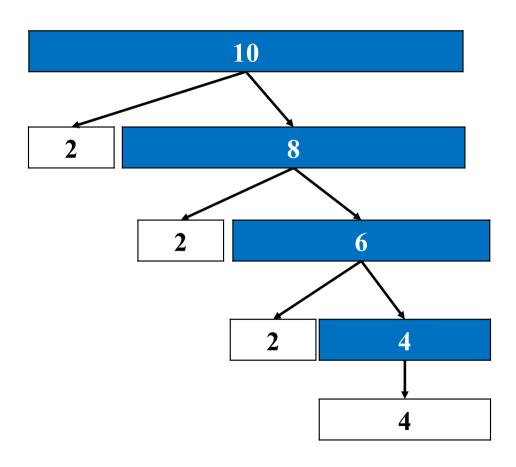






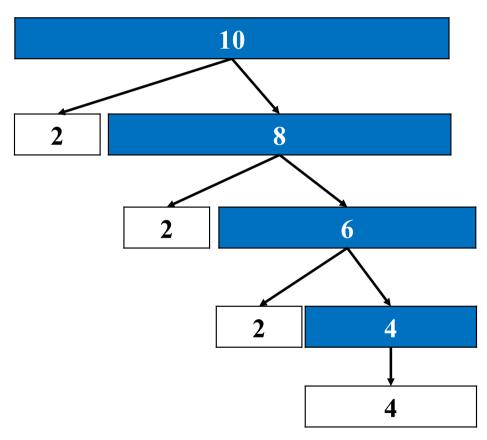








• 原始问题不限制切割次数



• 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



- 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



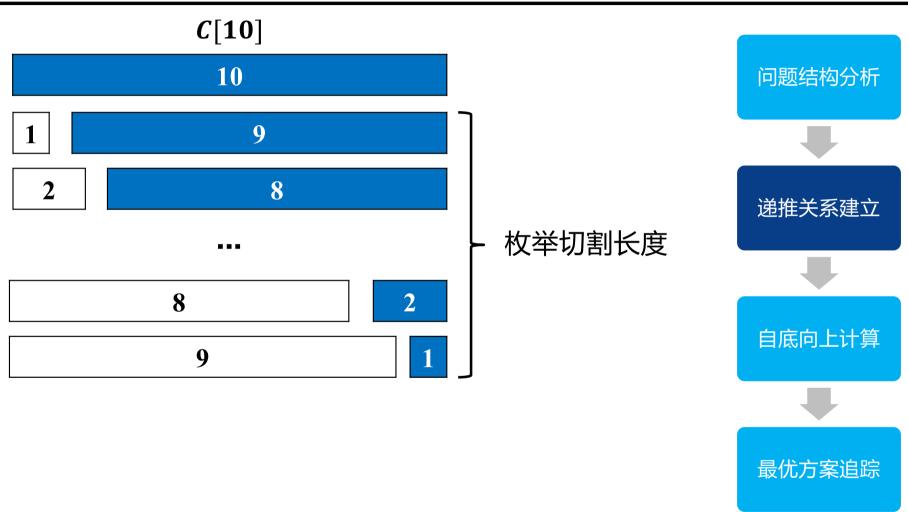
自底向上计算



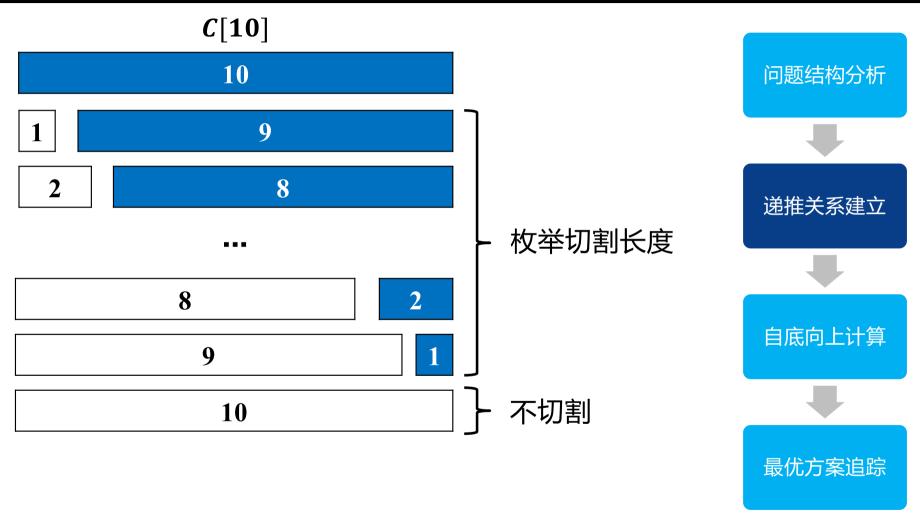
最优方案追踪

# 递推关系建立:分析最优(子)结构

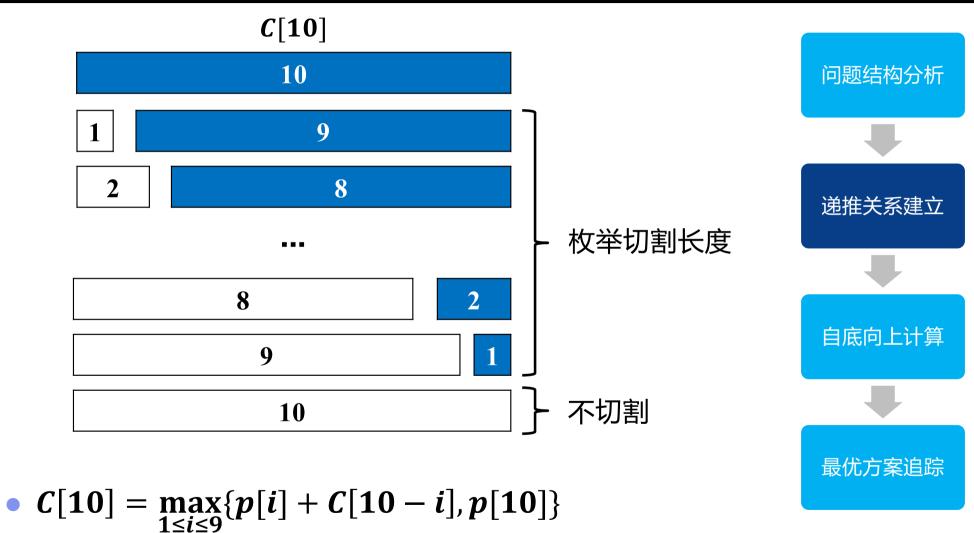




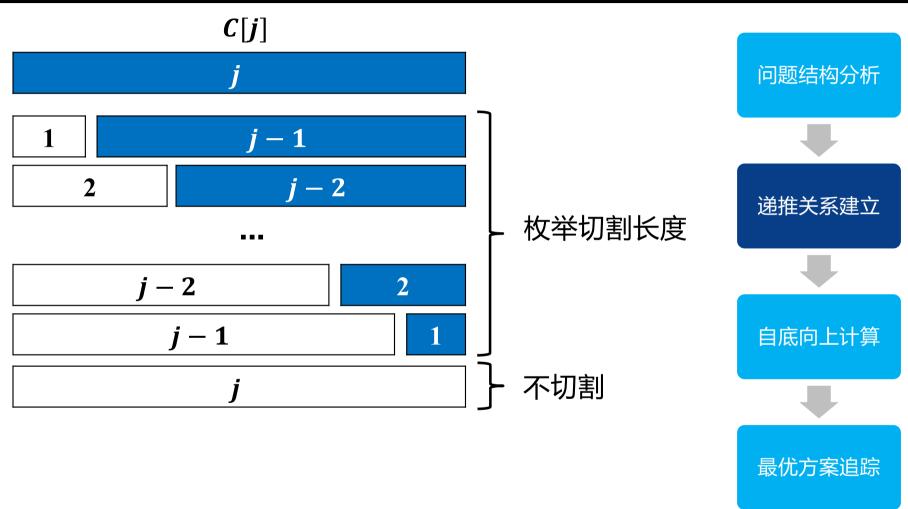




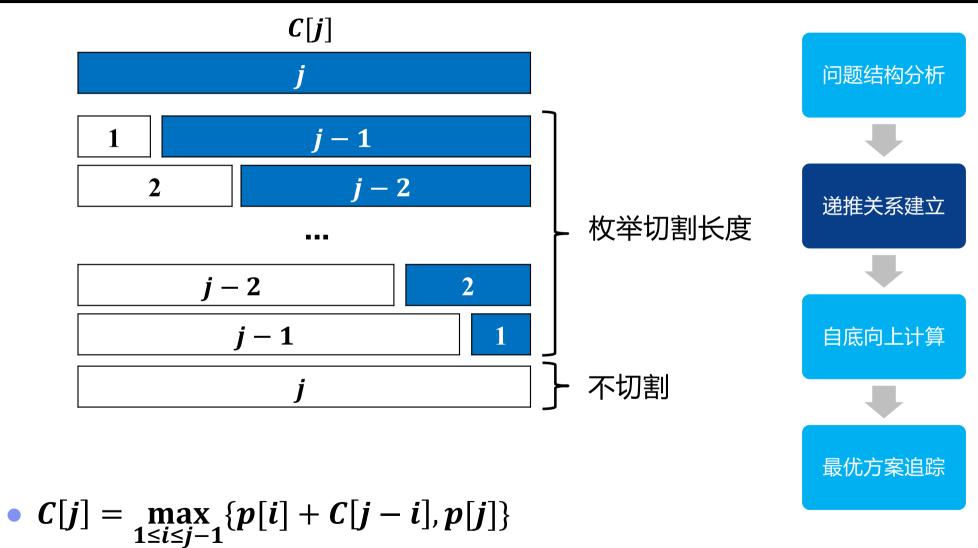




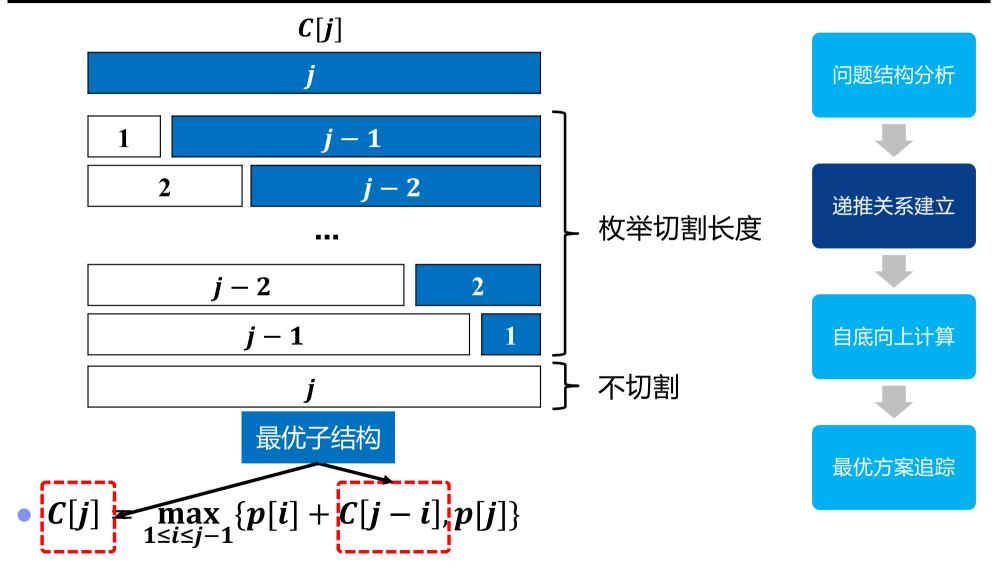














- 对于每个钢条长度j
  - $C[j] = \max_{1 \le i \le j-1} \{p[i] + C[j-i], p[j]\}$

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;

p[2]

**p**[1]

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

问题结构分析



递推关系建立



p[3]

max

C[j]

自底向上计算



最优方案追踪

p[j]

p[j-1]

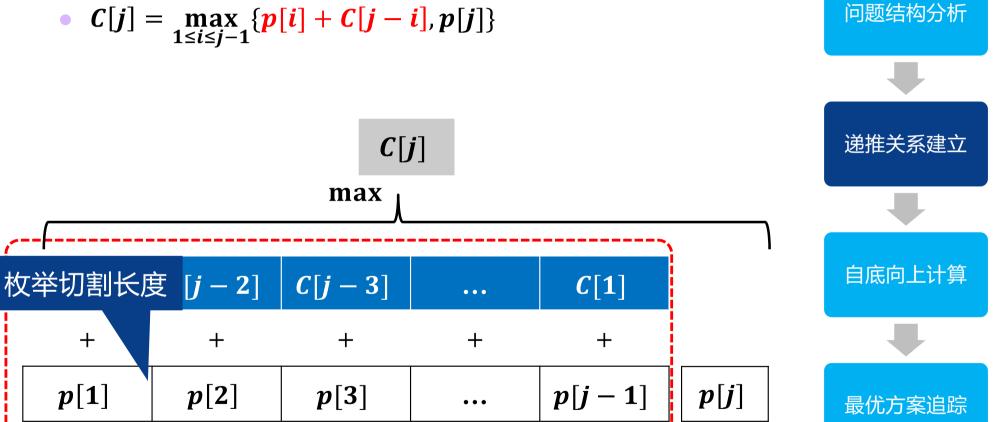


• 对于每个钢条长度;

+

**p**[1]

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

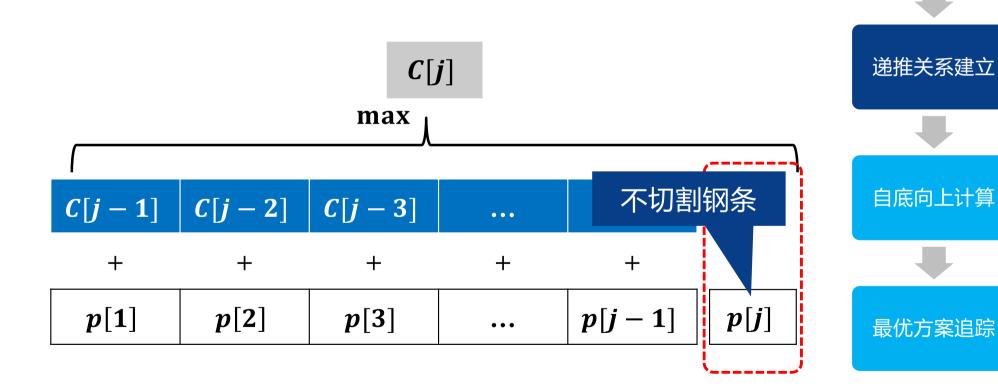




问题结构分析

• 对于每个钢条长度j

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



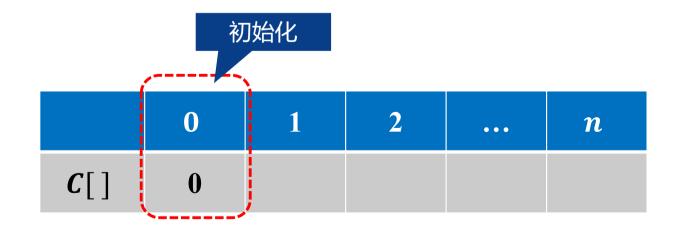
## 自底向上计算:确定计算顺序



#### 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



最优方案追踪

自底向上计算

## 自底向上计算:确定计算顺序

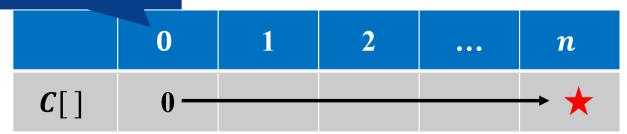


#### 已知钢条价格

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



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



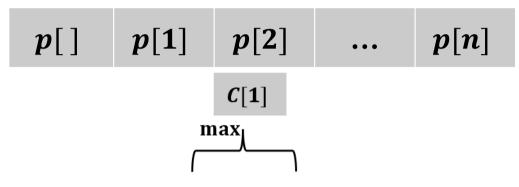
自底向上计算

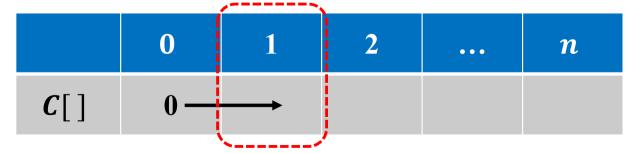


初始化	七				
	U	1	2	•••	n
<b>C</b> []	0 →				



#### 已知钢条价格





问题结构分析



递推关系建立

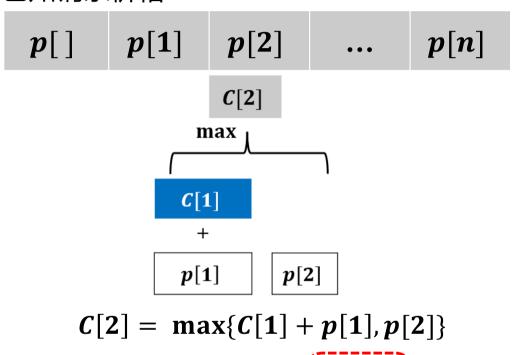


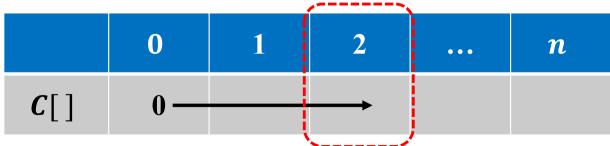
自底向上计算





#### 已知钢条价格





问题结构分析



递推关系建立

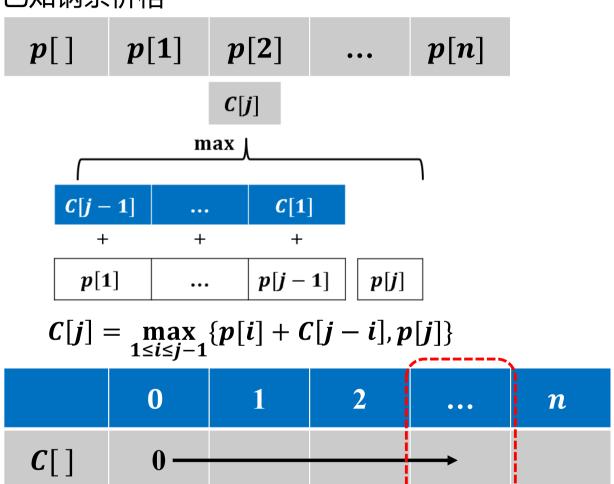


自底向上计算





#### 已知钢条价格



问题结构分析



递推关系建立

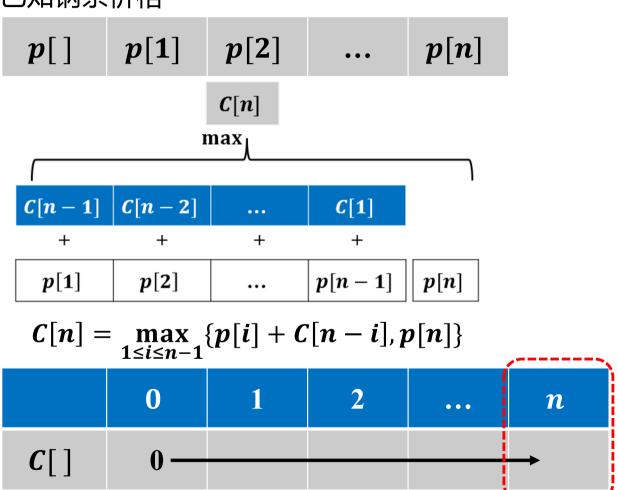


自底向上计算





#### 已知钢条价格



问题结构分析



递推关系建立

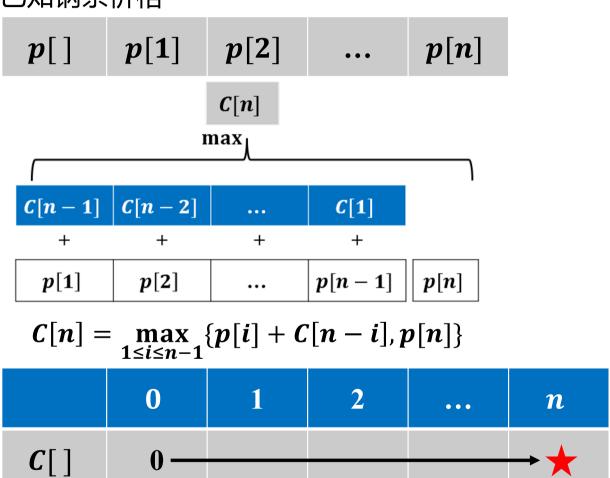


自底向上计算





#### 已知钢条价格



问题结构分析



递推关系建立



自底向上计算





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

j-1

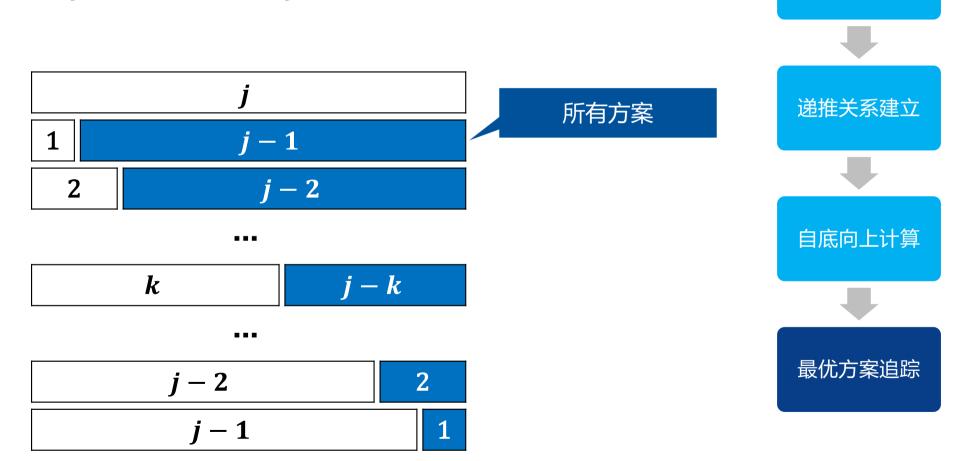


构造追踪数组rec[1..n] 问题结构分析 递推关系建立 所有方案 j-1j-2自底向上计算  $\boldsymbol{k}$ j-k最优方案追踪 j-2



问题结构分析

- 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

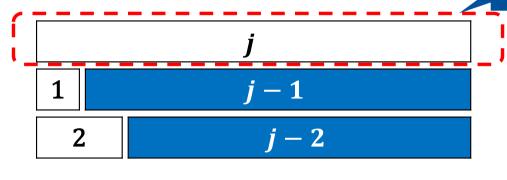




- 构造追踪数组rec[1..n]
- rec[j]: 记录长度为j钢条的最优切割方案

不切: rec[j] = j

记录方案中钢条长度



---



---

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

问题结构分析



递推关系建立

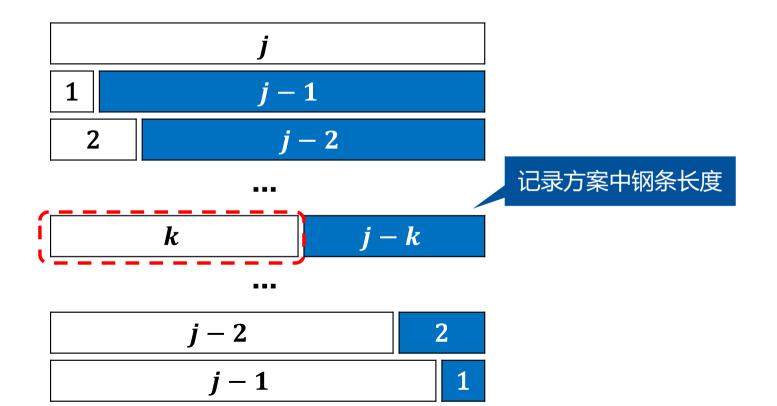


自底向上计算





- 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

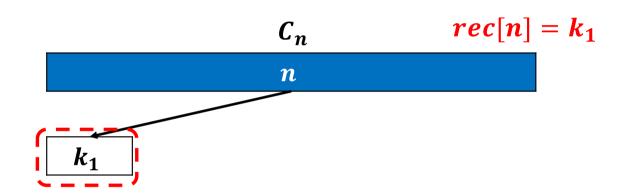


自底向上计算





- 根据追踪数组,递归输出方案
  - 输出长度为 $k_1$ 的钢条



问题结构分析



递推关系建立

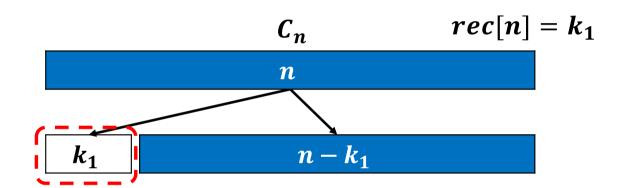


自底向上计算





- 根据追踪数组,递归输出方案
  - 输出长度为 $k_1$ 的钢条



问题结构分析



递推关系建立

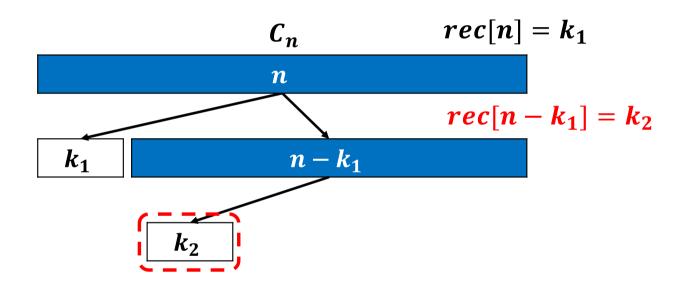


自底向上计算





- 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

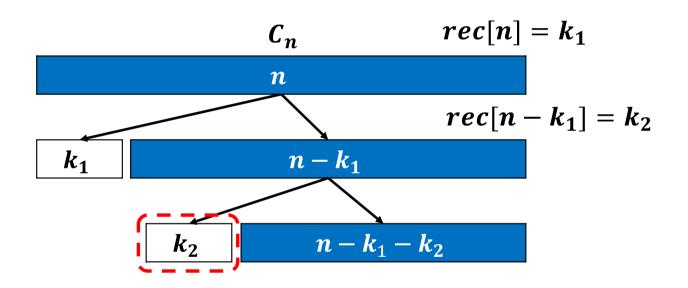


自底向上计算





- 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

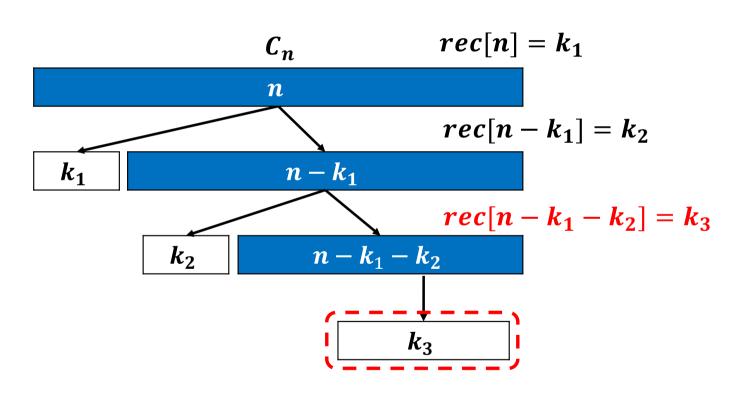


自底向上计算





- 根据追踪数组,递归输出方案
  - 输出长度为 $k_3$ 的钢条



问题结构分析



递推关系建立

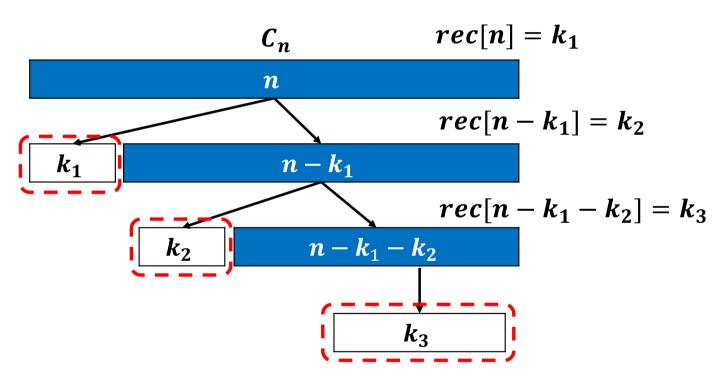


自底向上计算





- 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



$$k_1 + k_2 + k_3 = n$$

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	衫	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

$$j = 1$$

$$i \qquad 1 \qquad p[j] = p[1]$$

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										
	0					_		_			4.0
rec	O O	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

$$j = 1$$

$$i \qquad 1 \qquad \max\{p[i] + C[j-i], p[j]\}$$

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



$$n = 10$$

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

$$j = 2$$

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

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	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

$$j = 2$$

$$i \quad 1 \quad 2 \quad \max\{p[i] + C[j-i], p[j]\}$$

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

$$j = 2$$

$$i \quad 1 \quad 2 \quad \max\{p[i] + C[j-i], p[j]\}$$

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								



$$n = 10$$

	<u>,                                    </u>									
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	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								



$$n = 10$$

		<u> </u>								
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	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
$p_i$										

$$j = 3$$

i	1	2	3	$\max\{p[i] + C[j-i], \frac{p[j]}{p[j]}\}$
	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	$\max\{p[i] + C[j-i], p[j]\}$
	6	6	8	

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

$$j = 4$$

$$\begin{array}{c|c} i & 1 \\ \hline & p[1] + C[3] \end{array}$$

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]
	Q	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] = p[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							
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
$p_i$	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	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	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	2	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	$\max\{p[i] + C[j-i], p[j]\}$
	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	$\max\{p[i] + C[j-i], \frac{p[j]}{p}\}$
	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	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		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
		•	<b>-</b>	19				

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				

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



$$n = 10$$

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



$$n = 10$$

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



```
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
- //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
         \operatorname{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

\begin{array}{c|c}
 & \mathbf{for} \ j \leftarrow 1 \ to \ n \ \mathbf{do} \\
 & \overline{q} \leftarrow \overline{p}[j] - - -
\end{array}

                                                                      依次计算子问题
     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
           \operatorname{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
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
         \operatorname{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
                                                                    枚举切割长度
  -\frac{-\mathbf{if} \cdot q < p[\mathbf{i}] + C[\mathbf{j} - i] + \mathbf{then} - - -}{|q \leftarrow p[i] + C[\mathbf{j} - i]}
            rec[j] \leftarrow i
          \operatorname{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 q \leftarrow p[i] + C[j-i]
                                                    记录价格和决策
          rec[j] \leftarrow i
      \operatorname{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
         q \leftarrow p[i] + C[j-i]
          rec[j] \leftarrow i
       \operatorname{end}
                                                 保存子问题的解
```

## 伪代码



```
/ while n>0 do 
| print \ rec[n] 
| n \leftarrow n-rec[n] end
```

#### 时间复杂度分析



```
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>//</u>动态<u>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 do
   if q < p[i] + C[j-i] then
         q \leftarrow p[i] + C[j-i]
          rec[j] \leftarrow i
        \operatorname{end}
    \operatorname{end}
    C[j] \leftarrow q
                                                   时间复杂度:O(n^2)
end
```





