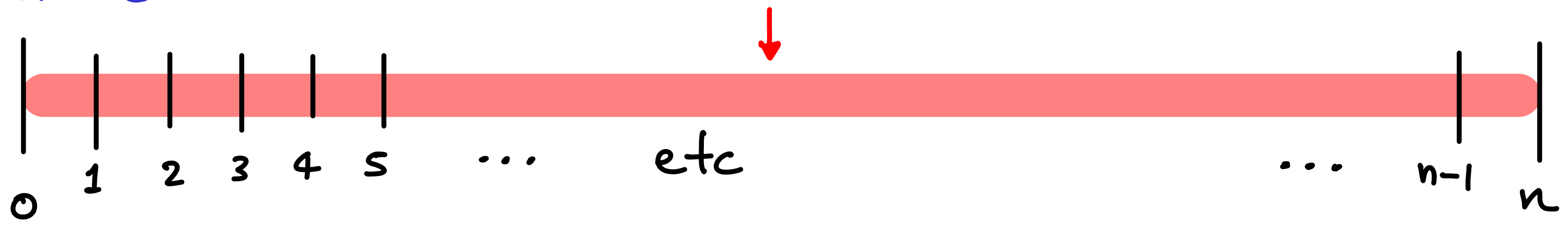
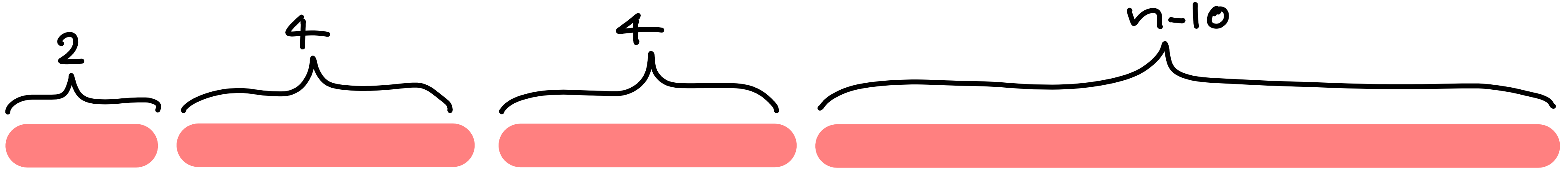
DYNAMIC PROGRAMMING – ROD CUTTING



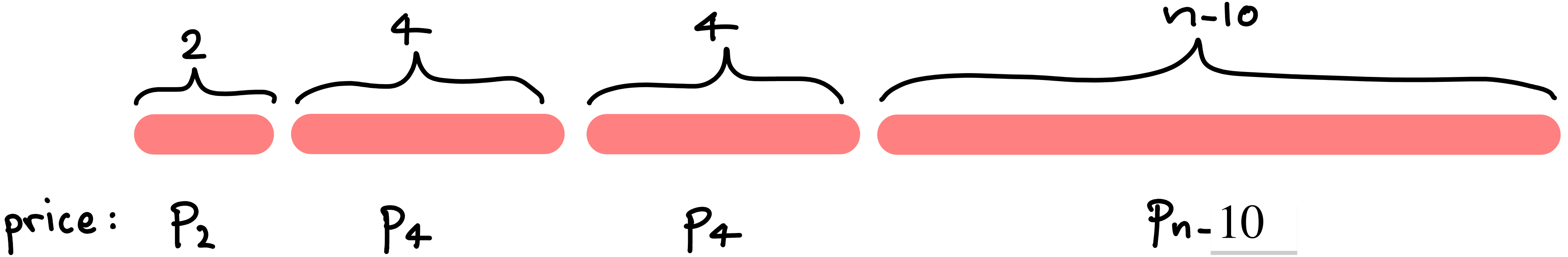
DYNAMIC PROGRAMMING - ROD CUTTING

You can cut at any integer position, for free.



DYNAMIC PROGRAMMING – ROD CUTTING

You can cut at any integer position, for free.



Every resulting piece will be sold, at a predefined price.

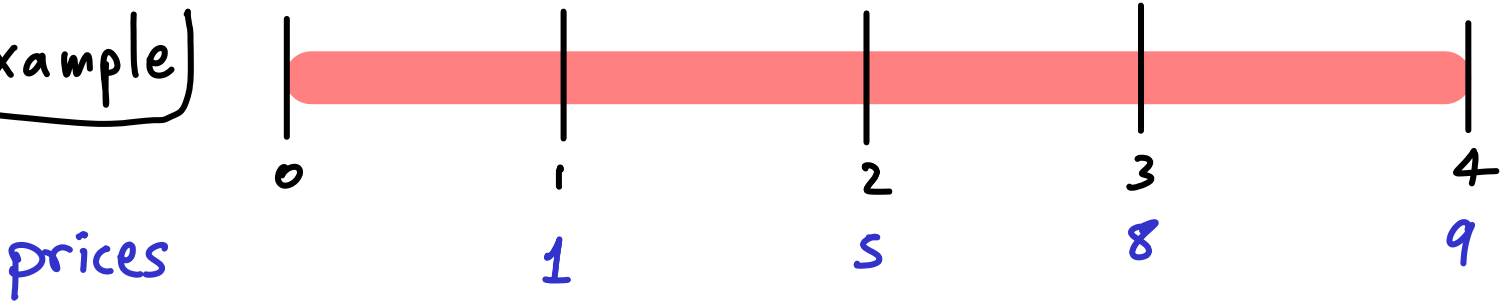
– Maximize profit –

We have the option to cut at every position $\{1 \ldots n-1\}$

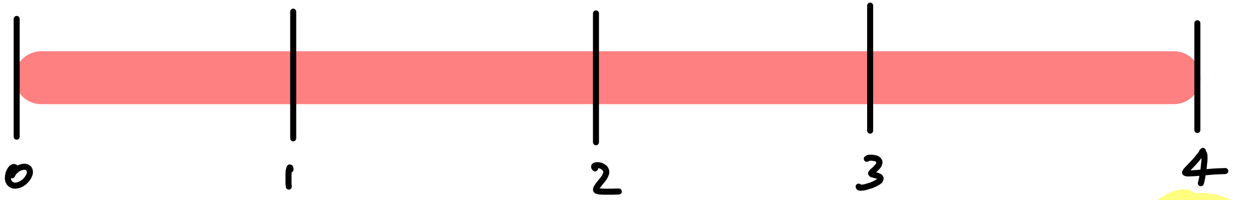
We have the option to cut at every position $\{1 \ldots n-1\}$

"Naive" counting : $\{0,1\}^{n-1} \rightarrow 2^{n-1}$ solutions : pick best.

example



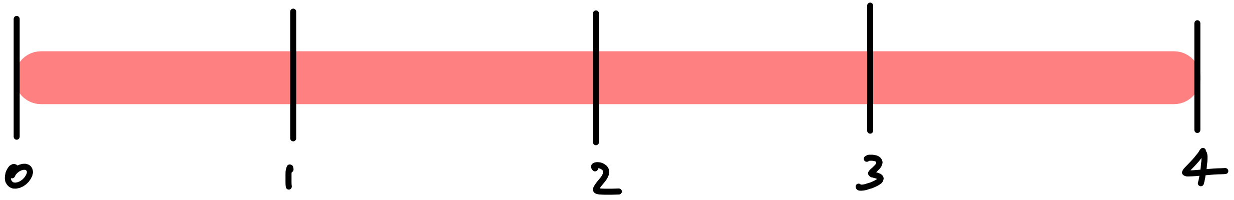
example



prices 1 5 8 9

option 1) Don't cut : profit = $1 \times p_4$ = 9

example



prices 1 5 8 9

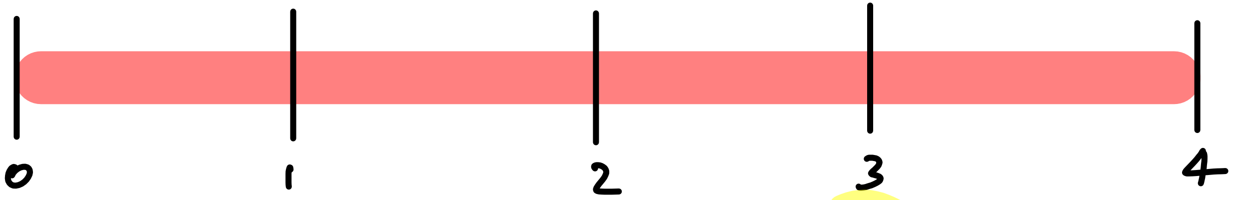
option 1) Don't cut : profit $= 1 \times p_4 = 9$

2) Cut once : (2.1) cut at 1 :

(2.2) cut at 2 :

(2.3) cut at 3 :

example



prices   1   5   8   9

option

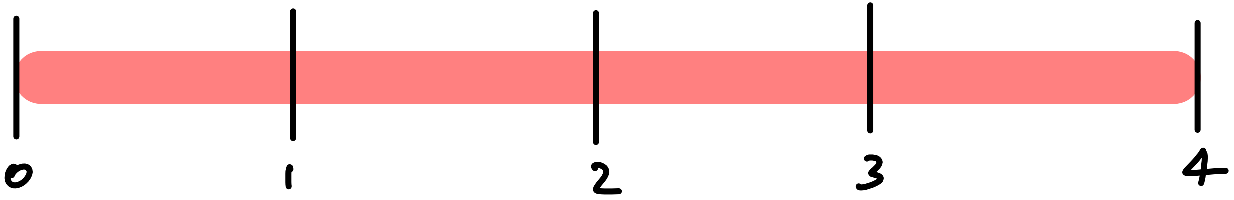
1) Don't cut : profit = $1 \times p_4 = 9$

2) Cut once : (2.1) cut at 1 : profit = $\overset{p_1}{1} + \overset{p_3}{8} = 9$

(2.2) cut at 2 :

(2.3) cut at 3 : profit = $8 + 1 = 9$

example



prices $\quad 1 \quad 5 \quad 8 \quad 9$

option 1) Don't cut : profit $= 1 \times p_4 = 9$

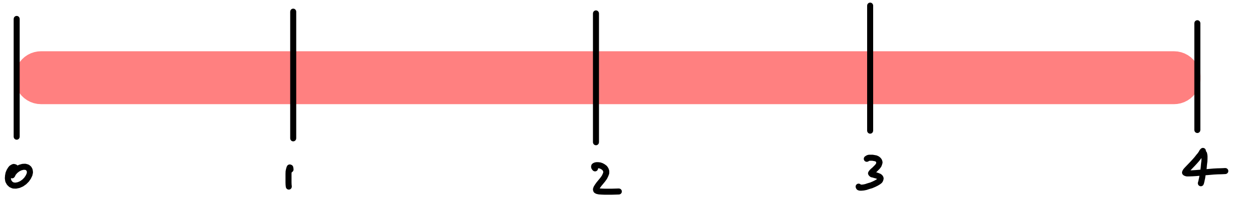
2) Cut once :

(2.1) cut at 1 : profit $= \overset{p_1}{1} + \overset{p_3}{8} = 9$

(2.2) cut at 2 : profit $= 5 + 5 = 10$

(2.3) cut at 3 : profit $= 8 + 1 = 9$

example



prices $\quad 1 \quad 5 \quad 8 \quad 9$

option

1) Don't cut : profit = $1 \times p_4 = 9$

2) Cut once :

(2.1) cut at 1 : profit = $\overset{p_1}{1} + \overset{p_3}{8} = 9$

(2.2) cut at 2 : profit = $5 + 5 = 10$

(2.3) cut at 3 : profit = $8 + 1 = 9$

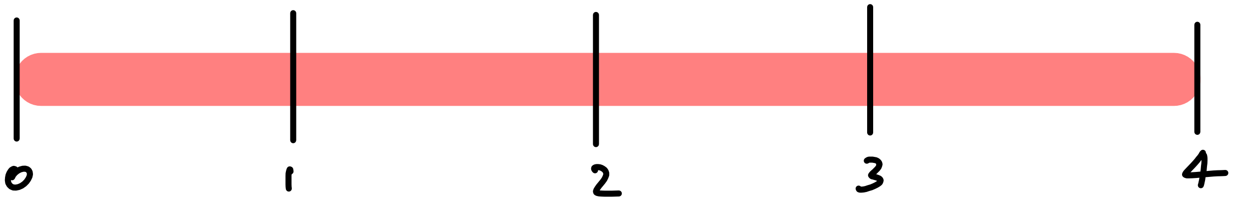
3) Cut twice :

(3.1) cut at 1 & 2 :

(3.2) cut at 1 & 3 :

(3.1) cut at 2 & 3 :

example



prices: 1, 5, 8, 9

option

1) Don't cut : profit $= 1 \times p_4 = 9$

2) Cut once :

(2.1) cut at 1 : profit $= \overset{p_1}{1} + \overset{p_3}{8} = 9$

(2.2) cut at 2 : profit $= 5 + 5 = 10$

(2.3) cut at 3 : profit $= 8 + 1 = 9$

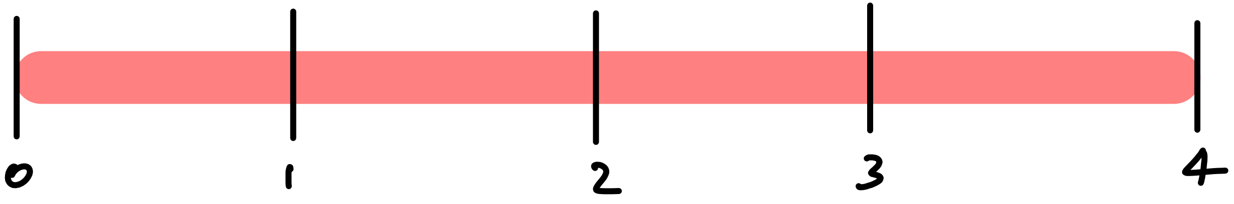
3) Cut twice :

(3.1) cut at 1 & 2 : profit $= 1 + 1 + 5 = 7$

(3.2) cut at 1 & 3 : profit $= 1 + 5 + 1 = 7$

(3.1) cut at 2 & 3 : profit $= 5 + 1 + 1 = 7$

example



prices 1 5 8 9

option

1) Don't cut : profit = $1 \times p_4 = 9$

2) Cut once : (2.1) cut at 1 : profit = $\overset{p_1}{1} + \overset{p_3}{8} = 9$

(2.2) cut at 2 : profit = $5 + 5 = \boxed{10}$

(2.3) cut at 3 : profit = $8 + 1 = 9$

3) Cut twice : (3.1) cut at 1 & 2 : profit = $1 + 1 + 5 = 7$

(3.2) cut at 1 & 3 : profit = $1 + 5 + 1 = 7$

(3.1) cut at 2 & 3 : profit = $5 + 1 + 1 = 7$

4) Cut 3 times : profit = $4 \times p_1 = 4$

$$\$(n) = \max \begin{cases} \text{no cut} \quad \cdots\cdots\cdots \quad \text{done} \\ \left.\begin{array}{l} \text{cut at position } 1 \\ \text{cut at position } 2 \\ \text{'' \quad '' \quad '' \quad } 3 \\ \quad\quad\quad \vdots \\ \text{cut at position } n-2 \\ \text{cut at position } n-1 \end{array}\right\} \text{ \& recurse} \end{cases}$$

$$\$(n) = \max \begin{cases} \text{no cut} \longrightarrow p_n \\ \text{cut at position } 1 \longrightarrow p_1 + \$(n-1) \\ \text{cut at position } 2 \longrightarrow \\ \text{" " " } 3 \longrightarrow \\ \vdots \\ \text{cut at position } n-2 \longrightarrow \\ \text{cut at position } n-1 \longrightarrow \end{cases}$$

$$
\$(n) = \max \begin{cases}
\text{no cut} \longrightarrow p_n \\
\text{cut at position } 1 \longrightarrow p_1 + \$(n-1) \\
\text{cut at position } 2 \longrightarrow \$(2) + \$(n-2) \\
\text{'' \quad '' \quad ''} \quad 3 \longrightarrow \\
\quad \vdots \\
\text{cut at position } n-2 \longrightarrow \\
\text{cut at position } n-1 \longrightarrow
\end{cases}
$$

$$\$(n) = \max \begin{cases} \text{no cut} \longrightarrow p_n \\ \text{cut at position } 1 \longrightarrow p_1 + \$(n-1) \\ \text{cut at position } 2 \longrightarrow \$(2) + \$(n-2) \\ \text{"} \quad \text{"} \quad \text{"} \quad 3 \longrightarrow \$(3) + \$(n-3) \\ \quad \vdots \qquad\qquad\qquad \vdots \\ \text{cut at position } n-2 \longrightarrow \$(n-2) + \$(2) \\ \text{cut at position } n-1 \longrightarrow \$(n-1) + p_1 \end{cases}$$

$$
\$(n) = \max \begin{cases} \text{no cut} \longrightarrow p_n \\ \text{cut at position } 1 \longrightarrow p_1 + \$(n-1) \\ \text{cut at position } 2 \longrightarrow \$(2) + \$(n-2) \\ \text{'' '' '' } 3 \longrightarrow \$(3) + \$(n-3) \\ \vdots \qquad\qquad \vdots \\ \text{cut at position } n-2 \longrightarrow \$(n-2) + \$(2) \\ \text{cut at position } n-1 \longrightarrow \$(n-1) + p_1 \end{cases}
$$

$$
\max \begin{cases} p_{n-1} \\ p_1 + \$(n-2) \\ \$(2) + \$(n-3) \\ \$(3) + \$(n-4) \\ \vdots \\ \$(n-3) + \$(2) \\ \$(n-2) + p_1 \end{cases}
$$

AWFUL if just using recursion

Change description of solution:

Cut at i

keep size(i)

+ recurse on <u>n-i</u>

Change description of solution:

$$\$(n) = \max \begin{cases} \text{no cut} \longrightarrow p_n \\ \text{cut at position } 1 \longrightarrow p_1 + \$(n-1) \\ \text{cut at position } 2 \longrightarrow p_2 + \$(n-2) \\ \text{" \quad " \quad " \quad } 3 \longrightarrow p_3 + \$(n-3) \\ \vdots \\ \text{cut at position } n-2 \longrightarrow p_{n-2} + \$(2) \\ \text{cut at position } n-1 \longrightarrow p_{n-1} + \$(1) \end{cases}$$

leftmost cut

leftover

Cut at i

keep size(i)

+ recurse on $\underline{n-i}$

Change description of solution:

$$\$(n) = \max \begin{cases} \text{no cut} & \longrightarrow p_n + \$(0) \\ \text{cut at position } 1 & \longrightarrow p_1 + \$(n-1) \\ \text{cut at position } 2 & \longrightarrow p_2 + \$(n-2) \\ \text{"} \quad \text{"} \quad \text{"} \quad 3 & \longrightarrow p_3 + \$(n-3) \\ \vdots & \\ \text{cut at position } n-2 & \longrightarrow p_{n-2} + \$(2) \\ \text{cut at position } n-1 & \longrightarrow p_{n-1} + \$(1) \end{cases}$$

Cut at i

keep size(i)

+ recurse on n-i

$$\$(n) = \max_{1 \leq i \leq n} \left\{ p_i + \$(n-i) \right\}$$

$$\$(0) = 0$$

Change description of solution:

$$\$(n) = \max \begin{cases} \text{no cut} \longrightarrow p_n + \$(0) \\ \text{cut at position } 1 \longrightarrow p_1 + \$(n-1) \\ \text{cut at position } 2 \longrightarrow p_2 + \$(n-2) \\ \text{" \quad " \quad " \quad } 3 \longrightarrow p_3 + \$(n-3) \\ \vdots \\ \text{cut at position } n-2 \longrightarrow p_{n-2} + \$(2) \\ \text{cut at position } n-1 \longrightarrow p_{n-1} + \$(1) \end{cases}$$

Cut at i

keep size(i)

+ recurse on $\underline{n-i}$

Compute & store all $\$(n-i)$

$$\$(n) = \max_{1 \leq i \leq n} \{ p_i + \$(n-i) \} \rightarrow T(n) = \Theta(n)$$

$$\$(0) = 0$$

Change description of solution:

$$\$(n) = \max \begin{cases} \text{no cut} \longrightarrow p_n + \$(0) \\ \text{cut at position } 1 \longrightarrow p_1 + \$(n-1) \\ \text{cut at position } 2 \longrightarrow p_2 + \$(n-2) \\ \text{'' '' '' } 3 \longrightarrow p_3 + \$(n-3) \\ \vdots \\ \text{cut at position } n-2 \longrightarrow p_{n-2} + \$(2) \\ \text{cut at position } n-1 \longrightarrow p_{n-1} + \$(1) \end{cases}$$

Cut at i

keep size(i)

+ recurse on $\underline{n-i}$

$$\$(n) = \max_{1 \le i \le n} \{ p_i + \$(n-i) \}$$

$$\$(0) = 0$$

Compute & store all $\$(n-i)$

$\downarrow$

$\rightarrow T(n) = \Theta(n)$

TOTAL $= \Theta(n^2)$

Example

size :	1	2	3	4	5	6	7	8
price :	1	5	8	9	10	17	17	20

$$\$(n) = \max_{1 \le i \le n} \{ p_i + \$(n-i) \}$$

Example size : 1 2 3 4 5 6 7 8

size :	1	2	3	4	5	6	7	8
price :	1	5	8	9	10	17	17	20
\$:	1							

$$\$(n) = \max_{1 \leq i \leq n} \{ p_i + \$(n-i) \}$$

$\$(1) = 1$

Example

size :	1	2	3	4	5	6	7	8
price :	1	5	8	9	10	17	17	20
\$:	1							

$$\$(n) = \max_{1 \leq i \leq n} \{ p_i + \$(n-i) \}$$

$\$(1) = 1$

$\$(2) = \underline{1 + \$(1)}$ OR $5 + \emptyset$

Example

size :	1	2	3	4	5	6	7	8
price :	1	5	8	9	10	17	17	20
\$:	1	5						

$$\$(n) = \max_{1 \le i \le n} \{ p_i + \$(n-i) \}$$

$\$(1) = 1$

$\$(2) = 1 + \$(1)$ OR $\underline{5 + \emptyset} \rightarrow 5$ (don't cut)

Example

size :	1	2	3	4	5	6	7	8
price :	1	5	8	9	10	17	17	20
$:	1	5						

$$\$(n) = \max_{1 \le i \le n} \{ p_i + \$(n-i) \}$$

$\$(1) = 1$

$\$(2) = 1 + \$(1)$ OR $5 + \emptyset \rightarrow 5$

$\$(3) = 1 + \$(2)$ OR $5 + \$(1)$ OR 8

Example

size :	1	2	3	4	5	6	7	8
price :	1	5	8	9	10	17	17	20
\$:	1	5	8					

$$\$(n) = \max_{1 \le i \le n} \{ p_i + \$(n-i) \}$$

$\$(1) = 1$

$\$(2) = 1 + \$(1)$ OR $\underline{5 + \emptyset} \rightarrow 5$

$\$(3) = 1 + \$(2)$ OR $5 + \$(1)$ OR $8 \rightarrow 6$ OR 6 OR $\underline{8} \rightarrow 8$

(don't cut)

Example

size :	1	2	3	4	5	6	7	8
price :	1	5	8	9	10	17	17	20
\$:	1	5	8					

$$\$(n) = \max_{1 \le i \le n} \{ p_i + \$(n-i) \}$$

$\$(1) = 1$

$\$(2) = 1 + \$(1)$ OR $\underline{5 + \emptyset} \rightarrow 5$

$\$(3) = 1 + \$(2)$ OR $5 + \$(1)$ OR $8 \rightarrow 6$ OR 6 OR $\underline{8} \rightarrow 8$

$\$(4) = 1 + \$(3)$ OR $5 + \$(2)$ OR $8 + \$(1)$ OR 9

Example

size :	1	2	3	4	5	6	7	8
price :	1	5	8	9	10	17	17	20
\$:	1	5	8	10				

$$\$(n) = \max_{1 \le i \le n} \{ p_i + \$(n-i) \}$$

$\$(1) = 1$

$\$(2) = 1 + \$(1)$ OR $\underline{5 + \emptyset} \rightarrow 5$

$\$(3) = 1 + \$(2)$ OR $5 + \$(1)$ OR $8 \rightarrow 6$ OR 6 OR $\underline{8} \rightarrow 8$

$\$(4) = 1 + \$(3)$ OR $\underline{5 + \$(2)}$ OR $8 + \$(1)$ OR $9 \rightarrow \max\{9, \underline{10}, 9, 9\} \rightarrow 10$

(cut 2 pieces of size 2)

Example

size :	1	2	3	4	5	6	7	8
price :	1	5	8	9	10	17	17	20
$:	1	5	8	10				

$$\$(n) = \max_{1 \le i \le n} \{ p_i + \$(n-i) \}$$

$\$(1) = 1$

$\$(2) = 1 + \$(1)$ OR $\underline{5 + \emptyset} \rightarrow 5$

$\$(3) = 1 + \$(2)$ OR $5 + \$(1)$ OR $8 \rightarrow 6$ OR 6 OR $\underline{8} \rightarrow 8$

$\$(4) = 1 + \$(3)$ OR $\underline{5 + \$(2)}$ OR $8 + \$(1)$ OR $9 \rightarrow \max\{9, \underline{10}, 9, 9\} \rightarrow 10$

$\$(5) = 1 + \$(4)$ OR $5 + \$(3)$ OR $8 + \$(2)$ OR $9 + \$(1)$ OR 10

Example

size :	1	2	3	4	5	6	7	8
price :	1	5	8	9	10	17	17	20
\$:	1	5	8	10	13			

$$\$(n) = \max_{1 \le i \le n} \{ p_i + \$(n-i) \}$$

$\$(1) = 1$

$\$(2) = 1 + \$(1)$ OR $\underline{5 + \emptyset}$ $\rightarrow 5$

$\$(3) = 1 + \$(2)$ OR $5 + \$(1)$ OR 8 $\rightarrow 6$ OR 6 OR $\underline{8} \rightarrow 8$

$\$(4) = 1 + \$(3)$ OR $\underline{5 + \$(2)}$ OR $8 + \$(1)$ OR 9 $\rightarrow \max\{9, \underline{10}, 9, 9\} \rightarrow 10$

$\$(5) = 1 + \$(4)$ OR $5 + \$(3)$ OR $\underline{8 + \$(2)}$ OR $9 + \$(1)$ OR 10 $\rightarrow \max\{11, 13, \underline{13}, 10, 10\}$

(cut a 3 & deal with leftovers: size 2
↳ don't cut again)

Example

size:	1	2	3	4	5	6	7	8
price:	1	5	8	9	10	17	17	20
\$:	1	5	8	10	13			

$$\$(n) = \max_{1 \le i \le n} \{ p_i + \$(n-i) \}$$

$\$(1) = 1$

$\$(2) = 1 + \$(1)$ OR $\underline{5 + \emptyset} \rightarrow 5$

$\$(3) = 1 + \$(2)$ OR $5 + \$(1)$ OR $8 \rightarrow 6$ OR 6 OR $\underline{8} \rightarrow 8$

$\$(4) = 1 + \$(3)$ OR $\underline{5 + \$(2)}$ OR $8 + \$(1)$ OR $9 \rightarrow \max\{9, \underline{10}, 9, 9\} \rightarrow 10$

$\$(5) = 1 + \$(4)$ OR $5 + \$(3)$ OR $\underline{8 + \$(2)}$ OR $9 + \$(1)$ OR $10 \rightarrow \max\{11, 13, \underline{13}, 10, 10\}$

$\$(6) = \{1 + 13,\ 5 + 10,\ 8 + 8,\ 9 + 5,\ 10 + 1,\ 17\}$

Example

size :	1	2	3	4	5	6	7	8
price :	1	5	8	9	10	17	17	20
\$:	1	5	8	10	13	17		

$$\$(n) = \max_{1 \le i \le n} \{ p_i + \$(n-i) \}$$

$\$(1) = 1$

$\$(2) = 1 + \$(1)$ OR $\underline{5 + \emptyset} \rightarrow 5$

$\$(3) = 1 + \$(2)$ OR $5 + \$(1)$ OR $8 \rightarrow 6$ OR 6 OR $\underline{8} \rightarrow 8$

$\$(4) = 1 + \$(3)$ OR $\underline{5 + \$(2)}$ OR $8 + \$(1)$ OR $9 \rightarrow \max\{9, \underline{10}, 9, 9\} \rightarrow 10$

$\$(5) = 1 + \$(4)$ OR $5 + \$(3)$ OR $\underline{8 + \$(2)}$ OR $9 + \$(1)$ OR $10 \rightarrow \max\{11, 13, \underline{13}, 10, 10\}$

$\$(6) = \{1 + 13,\ 5+10,\ 8+8,\ 9+5,\ 10+1,\ \underline{17}\} \rightarrow 17$

(don't cut)

Example

size :	1	2	3	4	5	6	7	8
price :	1	5	8	9	10	17	17	20
$:	1	5	8	10	13	17		

$$\$(n) = \max_{1 \le i \le n} \{ p_i + \$(n-i) \}$$

$\$(1) = 1$

$\$(2) = 1 + \$(1)$ OR $\underline{5 + \emptyset} \rightarrow 5$

$\$(3) = 1 + \$(2)$ OR $5 + \$(1)$ OR $8 \rightarrow 6$ OR 6 OR $\underline{8} \rightarrow 8$

$\$(4) = 1 + \$(3)$ OR $\underline{5 + \$(2)}$ OR $8 + \$(1)$ OR $9 \rightarrow \max\{9, \underline{10}, 9, 9\} \rightarrow 10$

$\$(5) = 1 + \$(4)$ OR $5 + \$(3)$ OR $\underline{8 + \$(2)}$ OR $9 + \$(1)$ OR $10 \rightarrow \max\{11, 13, \underline{13}, 10, 10\}$

$\$(6) = \{1+13,\ 5+10,\ 8+8,\ 9+5,\ 10+1,\ \underline{17}\} \rightarrow 17$

$\$(7) = \{1+17,\ 5+13,\ 8+10,\ 9+8,\ 10+5,\ 17+1,\ 17\}$

Example

size :	1	2	3	4	5	6	7	8
price :	1	5	8	9	10	17	17	20
\$:	1	5	8	10	13	17	18	

$$\$(n) = \max_{1 \le i \le n} \{ p_i + \$(n-i) \}$$

$\$(1) = 1$

$\$(2) = 1 + \$(1)$ OR $\underline{5 + \emptyset} \rightarrow 5$

$\$(3) = 1 + \$(2)$ OR $5 + \$(1)$ OR $8 \rightarrow 6$ OR 6 OR $\underline{8} \rightarrow 8$

$\$(4) = 1 + \$(3)$ OR $\underline{5 + \$(2)}$ OR $8 + \$(1)$ OR $9 \rightarrow \max\{9, \underline{10}, 9, 9\} \rightarrow 10$

$\$(5) = 1 + \$(4)$ OR $5 + \$(3)$ OR $\underline{8 + \$(2)}$ OR $9 + \$(1)$ OR $10 \rightarrow \max\{11, 13, \underline{13}, 10, 10\}$

$\$(6) = \{1+13,\ 5+10,\ 8+8,\ 9+5,\ 10+1,\ \underline{17}\} \rightarrow 17$

$\$(7) = \{\underline{1+17},\ \underline{5+13},\ \underline{8+10},\ 9+8,\ 10+5,\ \underline{17+1},\ 17\} \rightarrow 18$

cut 1 & 6

or 2 & 5 → 2,2,3

or 3 & 4 → 3,2,2

or 6 & 1

Example

size:	1	2	3	4	5	6	7	8
price:	1	5	8	9	10	17	17	20
\$:	1	5	8	10	13	17	18	

$$\$(n) = \max_{1 \le i \le n} \{ p_i + \$(n-i) \}$$

$\$(1) = 1$

$\$(2) = 1 + \$(1)$ OR $\underline{5 + \emptyset} \rightarrow 5$

$\$(3) = 1 + \$(2)$ OR $5 + \$(1)$ OR $8 \rightarrow 6$ OR 6 OR $\underline{8} \rightarrow 8$

$\$(4) = 1 + \$(3)$ OR $\underline{5 + \$(2)}$ OR $8 + \$(1)$ OR $9 \rightarrow \max\{9, \underline{10}, 9, 9\} \rightarrow 10$

$\$(5) = 1 + \$(4)$ OR $5 + \$(3)$ OR $\underline{8 + \$(2)}$ OR $9 + \$(1)$ OR $10 \rightarrow \max\{11, 13, \underline{13}, 10, 10\}$

$\$(6) = \{1 + 13, 5 + 10, 8 + 8, 9 + 5, 10 + 1, \underline{17}\} \rightarrow 17$

$\$(7) = \{\underline{1 + 17}, \underline{5 + 13}, \underline{8 + 10}, 9 + 8, 10 + 5, \underline{17 + 1}, 17\} \rightarrow 18$

$\$(8) = \{1 + 18, 5 + 17, 8 + 13, 9 + 10, 10 + 8, 17 + 5, 17 + 1, 20\}$

Example

size :	1	2	3	4	5	6	7	8
price :	1	5	8	9	10	17	17	20
$:	1	5	8	10	13	17	18	$\boxed{22}$

$$\$(n) = \max_{1\le i\le n}\{p_i + \$(n-i)\}$$

DONE

$\$(1) = 1$

$\$(2) = 1 + \$(1)$ OR $\underline{5+\emptyset} \to 5$

$\$(3) = 1 + \$(2)$ OR $5 + \$(1)$ OR $8 \to 6$ OR 6 OR $\underline{8} \to 8$

$\$(4) = 1 + \$(3)$ OR $\underline{5 + \$(2)}$ OR $8 + \$(1)$ OR $9 \to \max\{9, \underline{10}, 9, 9\} \to 10$

$\$(5) = 1 + \$(4)$ OR $5 + \$(3)$ OR $\underline{8 + \$(2)}$ OR $9 + \$(1)$ OR $10 \to \max\{11, 13, \underline{13}, 10, 10\}$

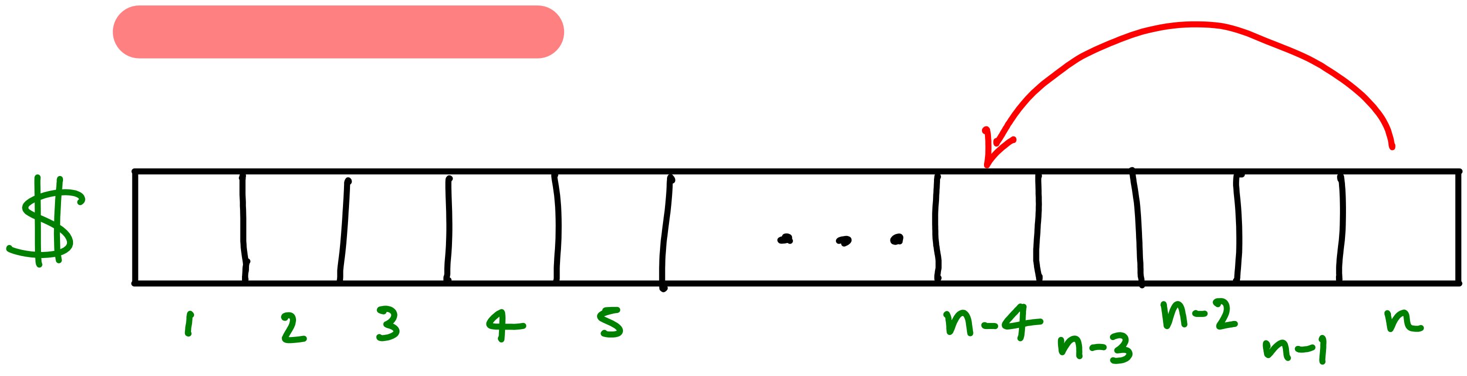
$\$(6) = \{1+13,\ 5+10,\ 8+8,\ 9+5,\ 10+1,\ \underline{17}\} \to 17$

$\$(7) = \{\underline{1+17},\ \underline{5+13},\ \underline{8+10},\ 9+8,\ 10+5,\ \underline{17+1},\ 17\} \to 18$

$\$(8) = \{1+18,\ \underline{5+17},\ 8+13,\ 9+10,\ 10+8,\ \underline{17+5},\ 17+1,\ 20\} \to 22$

cut 2 & 6 = 6 & 2

Retrieving cuts (not just profit)



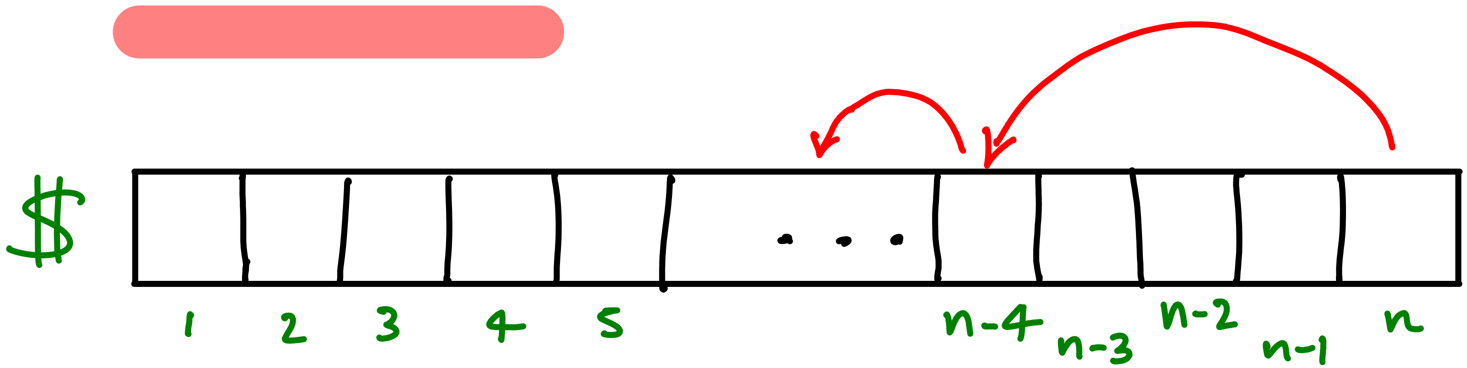
ex:

$$\$(n) = \max\{\cdot\} = p_4 + \$(n-4)$$

When computing $\$(n)$ we also see where we should cut

e.g. cut at p_4, leftovers: $\$(n-4)$, so follow pointer & report 4

Retrieving cuts (not just profit)



ex:

$\$(n) = \max\{\cdot\} =$

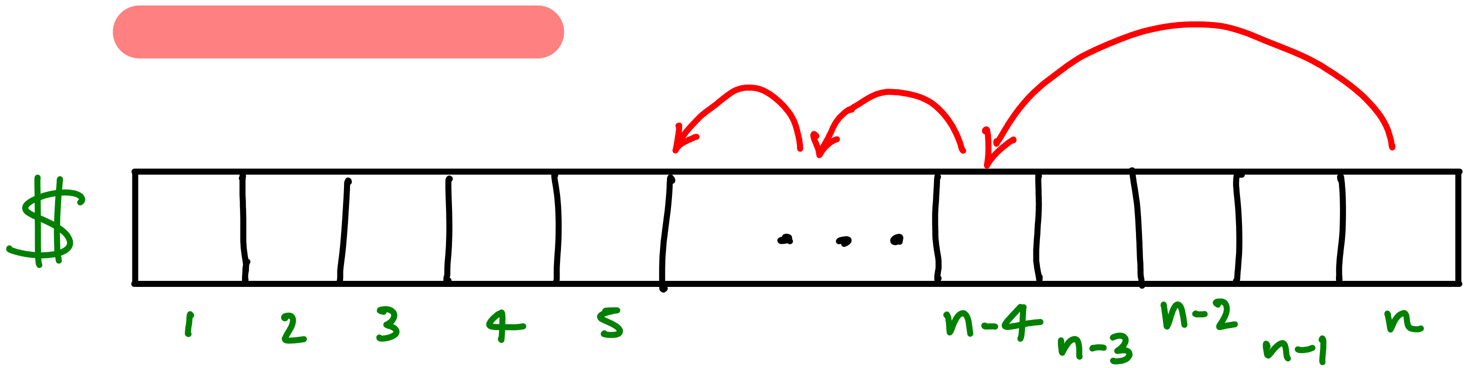
$p_4 + \$(n-4)$

When computing $\$(n)$ we also see where we should cut

e.g. cut at p_4, leftovers: $\$(n-4)$, so follow pointer & report 4

e.g. $\$(n-17)$, report $17-4 = 13$

Retrieving cuts (not just profit)



ex:

$\$(n) = \max\{\cdot\} = p_4 + \$(n-4)$

When computing $\$(n)$ we also see where we should cut

e.g. cut at p_4, leftovers: $\$(n-4)$, so follow pointer & report 4

↳ e.g. $\$(n-17)$, report $17-4=13$

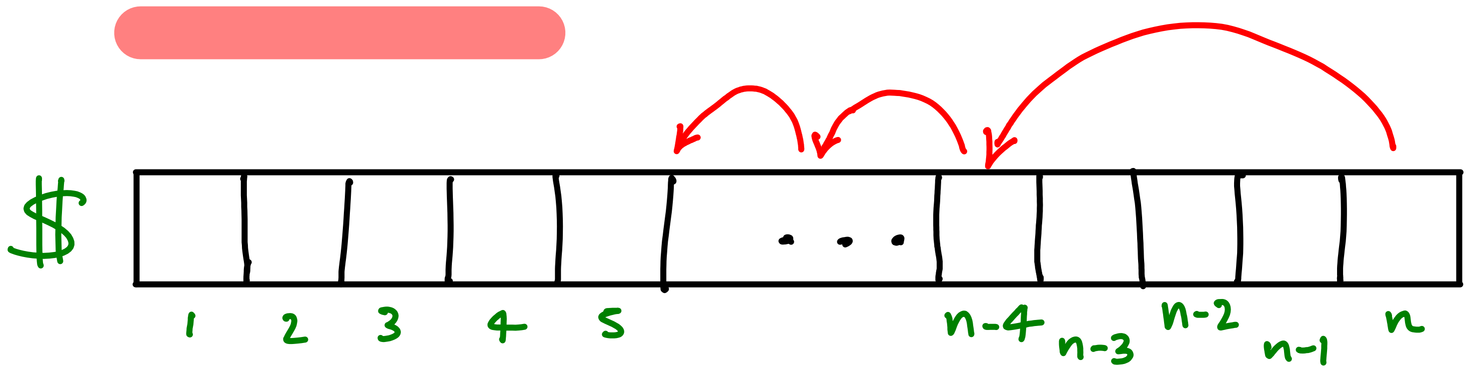
↳ e.g. $\$(n-50)$, report $50-17=33$ etc

Retrieving cuts (not just profit)

ex:

$\$(n) = \max\{\cdot\} =$

$p_4 + \$(n-4)$



When computing $\$(n)$ we also see where we should cut

e.g. cut at p_4, leftovers: $\$(n-4)$, so follow pointer & report 4

↪ e.g. $\$(n-17)$, report $17-4=13$

↪ e.g. $\$(n-50)$, report $50-17=33$ etc