Workshop #5
CSCI 432

Rod Cutting

length	1	2	3	4
profit	$1	$5	$8	$9



Given a rod of length n and a table of prices,
determine the maximum profit obtainable
by cutting the rod up and selling the pieces.

Rod Cutting

length	1	2	3	4
profit	$1	$5	$8	$9



profit = $9

Given a rod of length n and a table of prices, determine the maximum profit obtainable by cutting the rod up and selling the pieces.

Rod Cutting

length	1	2	3	4
profit	$1	$5	$8	$9



profit = $9



profit = $9

Given a rod of length n and a table of prices, determine the maximum profit obtainable by cutting the rod up and selling the pieces.

Rod Cutting

length	1	2	3	4
profit	$1	$5	$8	$9



profit = $9



profit = $9



profit = $9

Given a rod of length n and a table of prices,
determine the maximum profit obtainable
by cutting the rod up and selling the pieces.

Rod Cutting

length	1	2	3	4
profit	$1	$5	$8	$9



profit = $9



profit = $9



profit = $10



profit = $7



profit = $4

Given a rod of length n and a table of prices,
determine the maximum profit obtainable
by cutting the rod up and selling the pieces.

Rod Cutting

Assume there is a profit ≥ 0 for each possible length.

length	1	2	3	4
profit	$1	$5	$8	$9



profit = $9



profit = $9



profit = $10



profit = $7



profit = $4

Given a rod of length n and a table of prices, determine the maximum profit obtainable by cutting the rod up and selling the pieces.

Rod Cutting

Can we use dynamic programming?

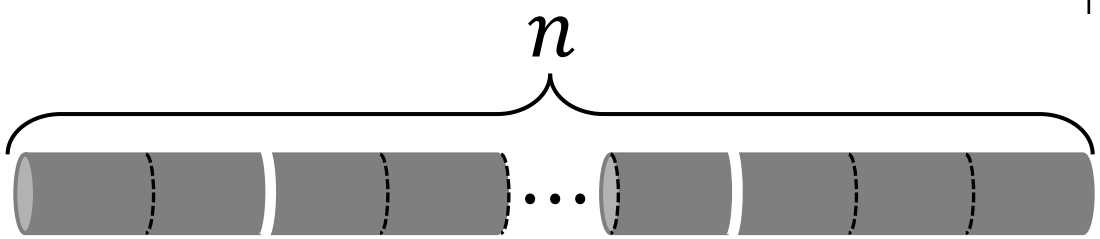
Rod Cutting

Does the rod cutting problem have optimal substructure?

Rod Cutting

Assume there is a profit ≥ 0
for each possible length.

length	1	2	3	4	…	n
profit	\$1	\$5	\$8	\$9	…	\$?

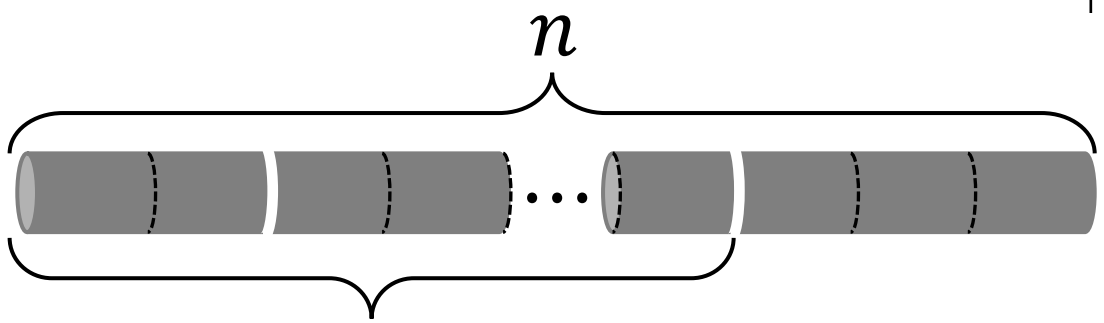


Does the rod cutting problem have optimal substructure?

I.e. Does an optimal solution to an instance
imply an optimal solution to a smaller instance?

Rod Cutting

length	1	2	3	4	…	n
profit	\$1	\$5	\$8	\$9	…	\$?



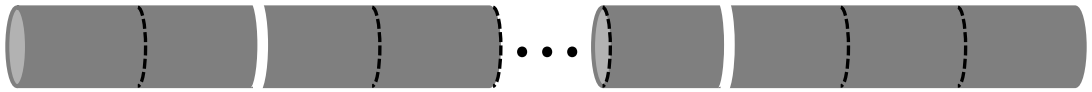
Optimal partition
for size = $n - 3$

Does the rod cutting problem have optimal substructure?

Yes! Given an optimal partition for n, every subset of that partition must also be optimal for its size!

Rod Cutting

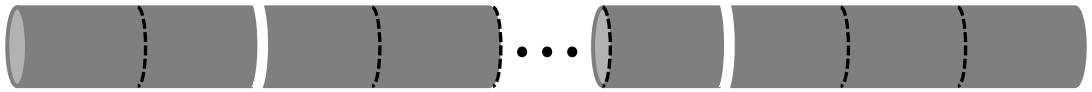
length	1	2	3	4	…	n
profit	\$1	\$5	\$8	\$9	…	\$?



$M[n] =$ optimal profit from partitioning rod of length n.
$p_i =$ profit for rod of length i.

Rod Cutting

length	1	2	3	4	…	n
profit	\$1	\$5	\$8	\$9	…	\$?

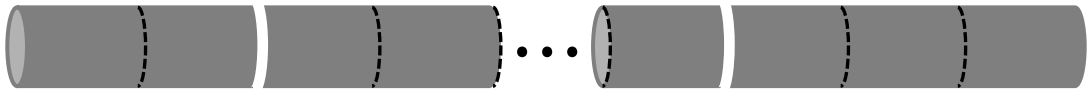


$M[n]$ = optimal profit from partitioning rod of length n.
p_i = profit for rod of length i.

$$M[n] = ?$$

Rod Cutting

length	1	2	3	4	…	n
profit	\$1	\$5	\$8	\$9	…	\$?



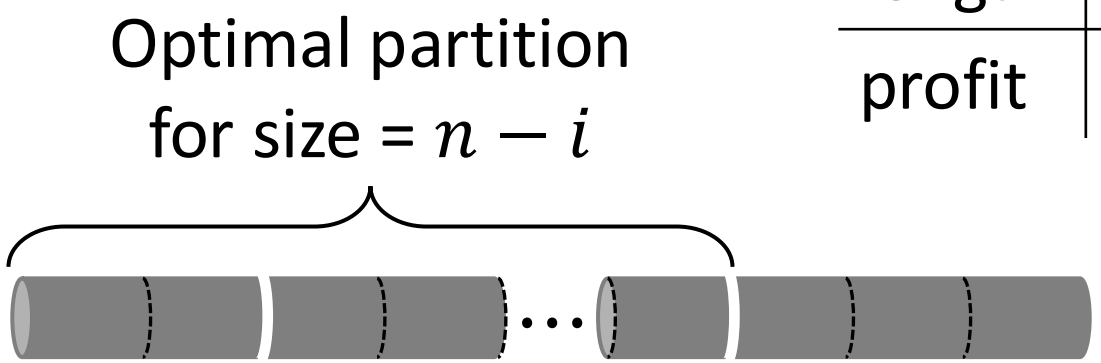
$M[n]$ = optimal profit from partitioning rod of length n.
p_i = profit for rod of length i.

$$M[n] = ?$$

Suppose length i is in the optimal solution.

Rod Cutting

length	1	2	3	4	…	n
profit	\$1	\$5	\$8	\$9	…	\$?



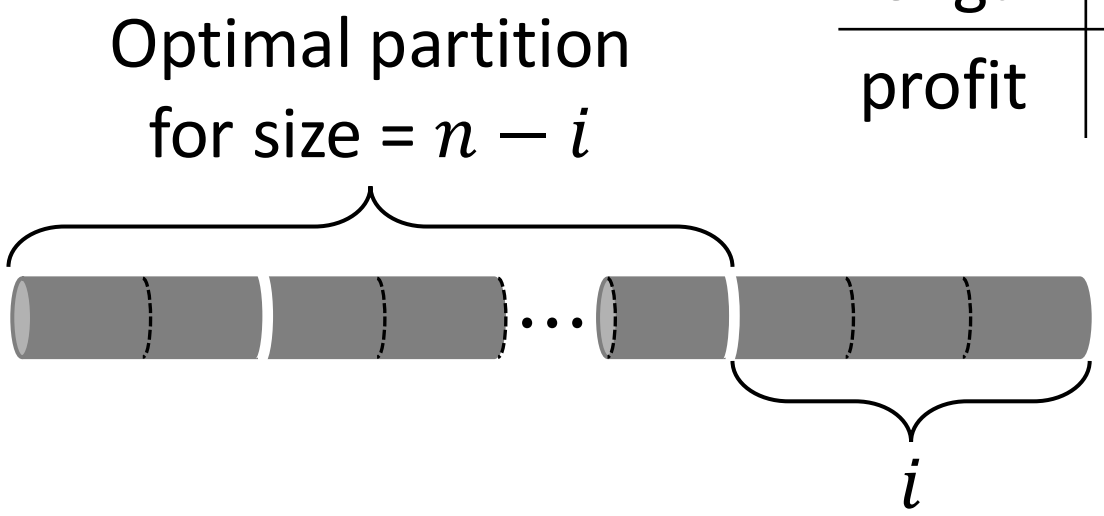
$M[n]$ = optimal profit from partitioning rod of length n.
p_i = profit for rod of length i.

$$M[n] = M[n-i]$$

Suppose length i is in the optimal solution.

Rod Cutting

length	1	2	3	4	…	n
profit	\$1	\$5	\$8	\$9	…	\$?



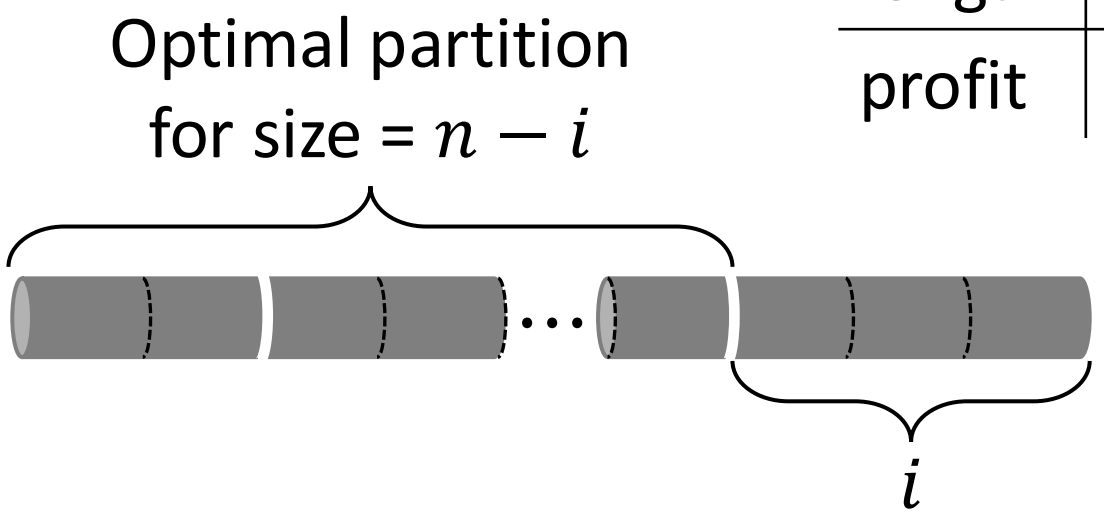
$M[n]$ = optimal profit from partitioning rod of length n.
p_i = profit for rod of length i.

$$M[n] = M[n-i] + i$$

Suppose length i is in the optimal solution.

Rod Cutting

length	1	2	3	4	…	n
profit	\$1	\$5	\$8	\$9	…	\$?



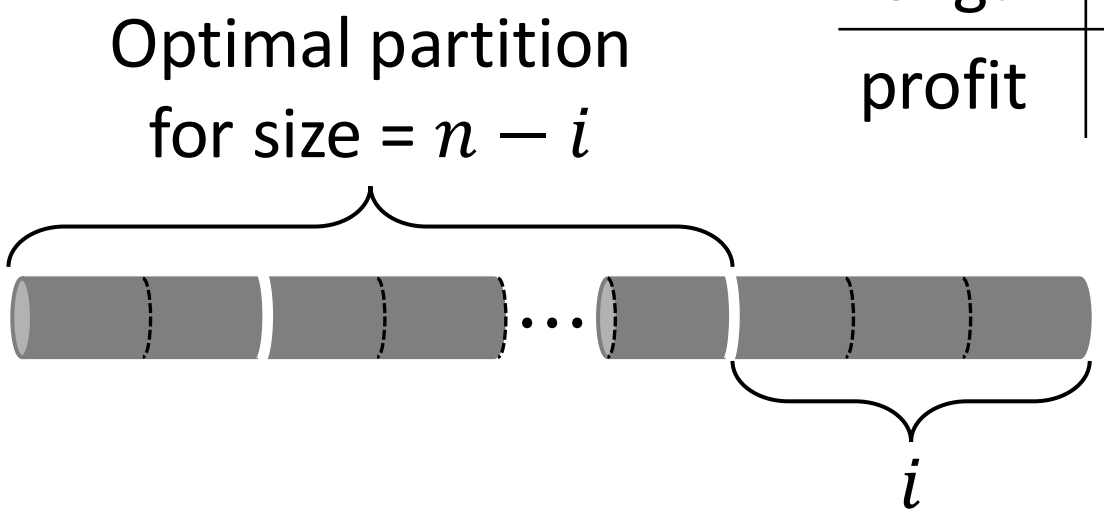
$M[n]$ = optimal profit from partitioning rod of length n.
p_i = profit for rod of length i.

$$M[n] = M[n-i] + p_i$$

Suppose length i is in the optimal solution.

Rod Cutting

length	1	2	3	4	…	n
profit	\$1	\$5	\$8	\$9	…	\$?



$M[n]$ = optimal profit from partitioning rod of length n.
p_i = profit for rod of length i.

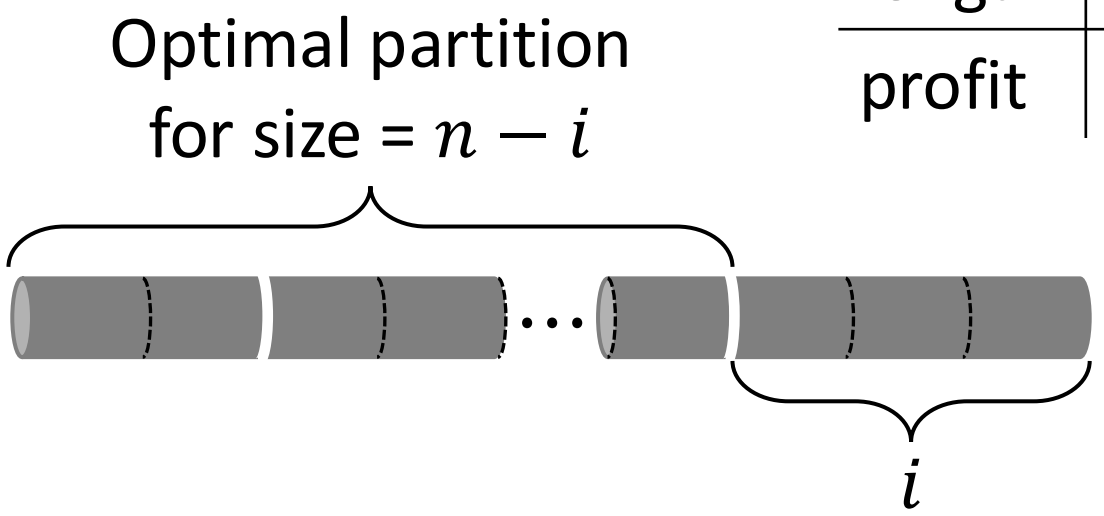
$$M[n] = M[n-i] + p_i$$

Suppose length i is in the optimal solution.

Do we know what i is in the optimal solution?

Rod Cutting

length	1	2	3	4	…	n
profit	\$1	\$5	\$8	\$9	…	\$?



$M[n]$ = optimal profit from partitioning rod of length n.
p_i = profit for rod of length i.

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

Check all possibilities!

Rod Cutting

Coding Plan:

?

Rod Cutting

Coding Plan:

1. Initialize array to hold what we care about.

Rod Cutting

Coding Plan:

1. Initialize array to hold what we care about.
2. Fill out array from small to large using:

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

Rod Cutting

Code?

Coding Plan:

1. Initialize array to hold what we care about.
2. Fill out array from small to large using:

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

Rod Cutting

```
partition(n, p)
   M[0,...,n] = [0,...,0]
   for m = 1 to n
      max = 0
      for length i ≤ m
         if M[m – i] + pi > max
            max = M[m – i] + pi
      M[m] = max
   return M[n]
```

Coding Plan:

1. Initialize array to hold what we care about.
2. Fill out array from small to large using:

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

Rod Cutting

```
partition(n, p)
   M[0,...,n] = [0,...,0]
   for m = 1 to n
      max = 0
      for length i ≤ m
         if M[m – i] + p_i > max
            max = M[m – i] + p_i
      M[m] = max
   return M[n]
```

Coding Plan:

1. Initialize array to hold what we care about.
2. Fill out array from small to large using:

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

Running time: ??

Rod Cutting

```
partition(n, p)
   M[0,...,n] = [0,...,0]
   for m = 1 to n
      max = 0
      for length i ≤ m
         if M[m – i] + p_i > max
            max = M[m – i] + p_i
      M[m] = max
   return M[n]
```

Coding Plan:

1. Initialize array to hold what we care about.
2. Fill out array from small to large using:

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

Running time: $O(n^2)$

Rod Cutting

```
partition(n, p)
   M[0,...,n] = [0,...,0]
   for m = 1 to n
      max = 0
      for length i ≤ m
         if M[m – i] + p_i > max
            max = M[m – i] + p_i
      M[m] = max
   return M[n]
```

Coding Plan:

1. Initialize array to hold what we care about.
2. Fill out array from small to large using:

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

Running time: $O(n^2)$

Rod Cutting

Determine a good strategy to get the actual cuts associated with the max profit, not just the value of the max profit.

```
partition(n, p)
   M[0,...,n] = [0,...,0]
   for m = 1 to n
      max = 0
      for length i ≤ m
         if M[m – i] + p_i > max
            max = M[m – i] + p_i
      M[m] = max
   return M[n]
```

Coding Plan:

1. Initialize array to hold what we care about.
2. Fill out array from small to large using:

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

Running time: $O(n^2)$

Rod Cutting

Determine a good strategy to get the actual cuts associated with the max profit, not just the value of the max profit.

```
partition(n, p)
   M[0,...,n] = [0,...,0]
   C[0,...,n] = [0,...,0]
   for m = 1 to n
      max = 0
      last_cut = 0
      for length i ≤ m
         if M[m – i] + p_i > max
            max = M[m – i] + p_i
            last_cut = i
      M[m] = max
      C[m] = last_cut
```

To get the actual cuts, start at C[n], add C[n] to the cut list, then make L = n – C[n]. Add C[L] to the cut list and update L = L – C[L]. Repeat until L = 0.