

15

15.1-2

No it cannot always produce an optimal solution. Consider the following example.

l_i	1	2	3
p_i	1	50	72
$\frac{p_i}{l_i}$	1	25	24

For $n = 3$ the greedy approach cut the rod in 2 pieces. The length of one of them is 2 and the other's is 1. So the profit is $50\$ + 1\$ = 51\$$. But the optimal solution is to keep the rod intact so the profit is 72\$.

15.1-3

We can keep the rod intact so we don't need to incur the fixed cost c or we can have at least one cut. We need to choose the best solution among all of them:

$$r(i) = \begin{cases} \max_{1 \leq k < n} (p_i, r(i-k) + p_k - c) & i > 0 \\ 0 & i = 0 \end{cases}$$

So the solution is $r(n)$. We have n distinct subproblem. In each step we need to choose between keeping the rod intact or have at least one cut which divide the rod into two pieces. The length of one of them is k and the other's $n - k$. We don't know the exact value of k so we need to try all possible values. This can be done in $O(n)$. Therefore the overall running time is $O(n^2)$

```
1: function F(p, n, c)
2:   let r[0..n] be a new array
3:   r[0]  $\leftarrow$  0
4:   for j from 1 to n do
5:     q  $\leftarrow$  p[j]
6:     for i from 1 to j - 1 do
7:       q = max(q, r[j-i] + p[i] - c)
8:     end for
9:     r[j] = q
10:  end for
11:  return r[n]
12: end function
```
