The rod-cutting problem is as follows. Given a rod of length $n \in \mathbb{N}$ and a price function $p: \mathbb{N} \to \mathbb{R}$, determine the maximum value of the rod if you can make any number of integer length cuts.

How do we determine what cuts to make?

The trick is no matter how you cut the rod, there is always a first cut.

Thus we get the recursive definition

$$V(n) = \begin{cases} 0 & n = 0 \\ \max_{1 \le \ell \le n} \left(p(\ell) + V(n-\ell) \right) & \text{otherwise} \end{cases}$$

RC (n, p)

Let $V: \mathbb{Z}_{\ge 0} \to \mathbb{R}$ be a map
$V(0) = 0$
For $i = 1$ to n
$$V(i) = \max_{1 \le \ell \le i} \left(p(\ell) + V(n-\ell) \right)$$

Return $V(n)$

The runtime of the rod-cutting algorithm is $\approx O(n^2)$ (the min hides another loop).

Remember, though, that the input size is $m = \log n$, so the runtime of interest is actually more complicated by less important, $\Theta(m 2^{2m})$, Each loop contributes a 2^m factor, and the additions/comparisons are linear.