Rod Cutting

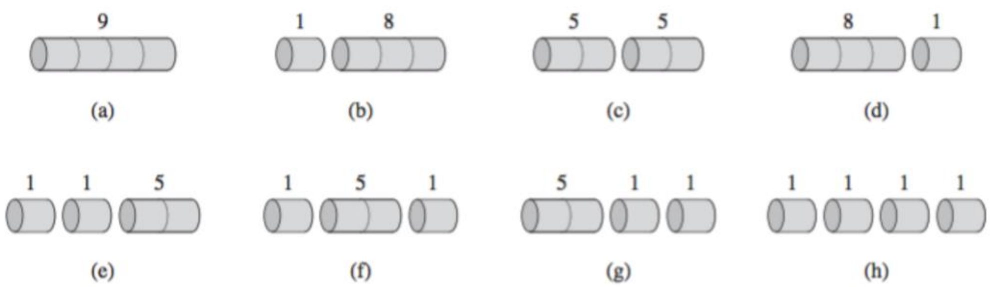
A company buys long steel rods from a manufacturer and cuts them into shorter rods for sale to its customers. If each cut is free, and rods of different lengths can be sold for different amounts, write a program to determine how to best cut the original rods to maximize revenue.

Input
You are given a rod of length n and the company's table of prices p where p_i is the price of a rod of length i (for i = 1, 2, ..., n).

Goal
To determine the maximum revenue r_n, obtainable by cutting up the rod and selling the pieces

Example
Given n = 4 and p_i = {0, 1, 5, 8, 9} (a rod of length 0 is assumed to have a profit of 0).

- If we do not cut the rod, we can earn $p_4 = 9$.
- If we cut it into 4 pieces of length 1, we earn $4 \cdot p_1 = 4$.
- If we cut it into 2 pieces of length 1 and a piece of length 2, we earn $2 \cdot p_1 + p_2 = 7$
- If we cut it into 2 pieces of length 2, we can earn $2 \cdot p_2 = 10$



There are more options (shown above), but 3 of the ways to cut are just permutations of other cuts we already made.

We can compute the maximum revenue (r_n) for rods of length n.
Example for rods of lengths [1..5]:

- r_1: Obvious
- r_2: $\max(p_2,\ p_1+p_1)$
- r_3: $\max(p_3,\ p_1+p_2,\ p_2+p_1,\ p_1+p_1+p_1)$
- r_4: $\max(p_4,\ p_1+p_3,\ p_2+p_2,\ p_3+p_1,\ p_1+p_1+p_2,\ p_1+p_2+p_1,\ p_2+p_1+p_1,\ p_1+p_1+p_1+p_1)$
- r_5: $\max(p_5,\ p_1+p_4, \ldots)$

length i	0	1	2	3	4	5
price p_i	0	1	5	8	9	12

length n	0	1	2	3	4	5
revenue r_n	0	1	5	8	10	13

The maximum revenue is 13. In general, rod of length n can be cut in 2^{n-1} different ways, since we can choose cutting, or not cutting, at all distances i from the left end.

You may observe that we can also calculate the maximum revenue r_n in terms of optimal revenues for all combinations.

$$
\begin{aligned}
r_5 &= \max(r_1 + r_4,\ r_2 + r_3,\ r_3 + r_2,\ r_4 + r_1,\ p_5) \\
&= \max(1 + 10,\ 5 + 8,\ 8 + 5,\ 10 + 1,\ 12) \\
&= \max(11,\ 13,\ 13,\ 11,\ 12) \\
&= 13
\end{aligned}
$$

length i	0	1	2	3	4	5
price p_i	0	1	5	8	9	12

length n	0	1	2	3	4	5
revenue r_n	0	1	5	8	10	13

General simplistic solution:
$r_n = \max(r_1 + r_{n-1}, r_2 + r_{n-2}, \ldots, r_{n-1} + r_1,\ p_i)$

Better solution: rather than adding two r values (e.g. r_2 and r_{n-2}) we can add a p value and an r value (e.g. p_2 and r_{n-2})

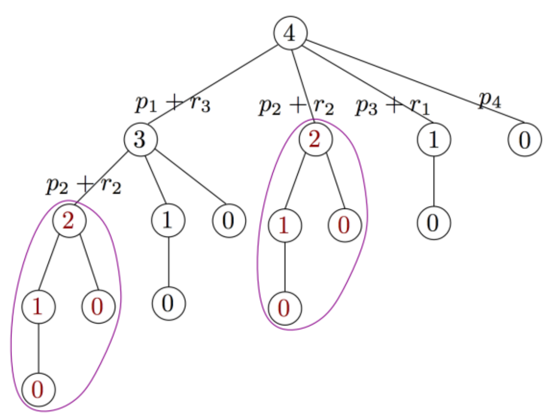
- This approach determines which first cut (i.e. p_i) gives the best revenue.
- This approach gives the same results but is easier to calculate

$$
\begin{aligned}
r_5 &= \max(p_1 + r_4,\ p_2 + r_3,\ p_3 + r_2,\ p_4 + r_1,\ p_5 + r_0) \\
&= \max(1 + 10,\ 5 + 8,\ 8 + 5,\ 9 + 1,\ 12 + 0) \\
&= \max(11,\ 13,\ 13,\ 10,\ 12) \\
&= 13
\end{aligned}
$$

In other words: With $r_0 = 0$, $r_n = \max(p_i + r_{n-i})$ for $1 \le i \le n$. Cut a piece of length i, with remainder of length $n - i$. Only the remainder, and not the first piece, may be further divided.

The optimal value can be found in terms of shorter rods by observing that if we make an optimal cut of length i (and thus also creating a piece of length $n - i$), then both pieces must be optimal (and then these smaller pieces will subsequently be cut). Otherwise, we could make a different cut which would produce a higher revenue, contradicting the assumption that the first cut was optimal.

Begin by (proactively) computing the optimal solutions for smaller rod lengths, and use these values to build solutions to larger rods (in a bottom-up fashion). This problem exhibits the "overlapping sub-problems" property, shown below:



Write a program to devise the maximum revenue for a steel rod of length n.

Input:

```
length = 8
prices_i = {0, 1, 5, 8, 9, 10, 17, 17, 20}
```

Output:

```
Max value for rod of length 8 -> 22
```

Advanced: Once you have the algorithm working, using additional storage, output the number of cuts required.

Number of cuts for length `n`**:** 0 1 2 3 2 2 6 1 2