**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 pi is the price of a rod of length i (for i = 1, 2, ..., n).

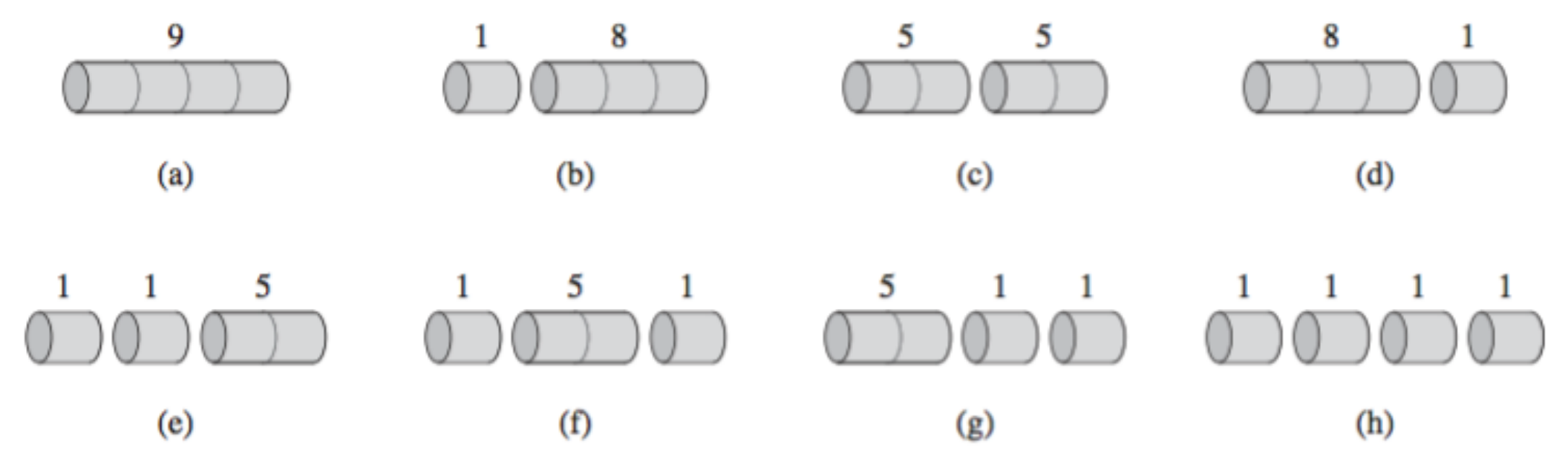
**Goal**

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

**Example**

Givenn = 4 and pi = {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 p4 = 9.
* If we cut it into 4 pieces of length 1, we earn 4 · p1 = 4.
* If we cut it into 2 pieces of length 1 and a piece of length 2, we earn 2 · p1 + p2 = 9
* If we cut it into 2 pieces of length 2, we can earn 2 · p2 = 10



There are more options (shown above), but the maximum revenue is 10. In general, rod of length n can be cut in 2n−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 calculate the maximum revenue rn in terms of optimal revenues for all combinations. Example for a rod of length 4:

r4 = max(p1 + r3, p2 + r2, p3 + r1, p4 + r0)

= max(1 + 8, 5 + 5, 8 + 1, 9 + 0)

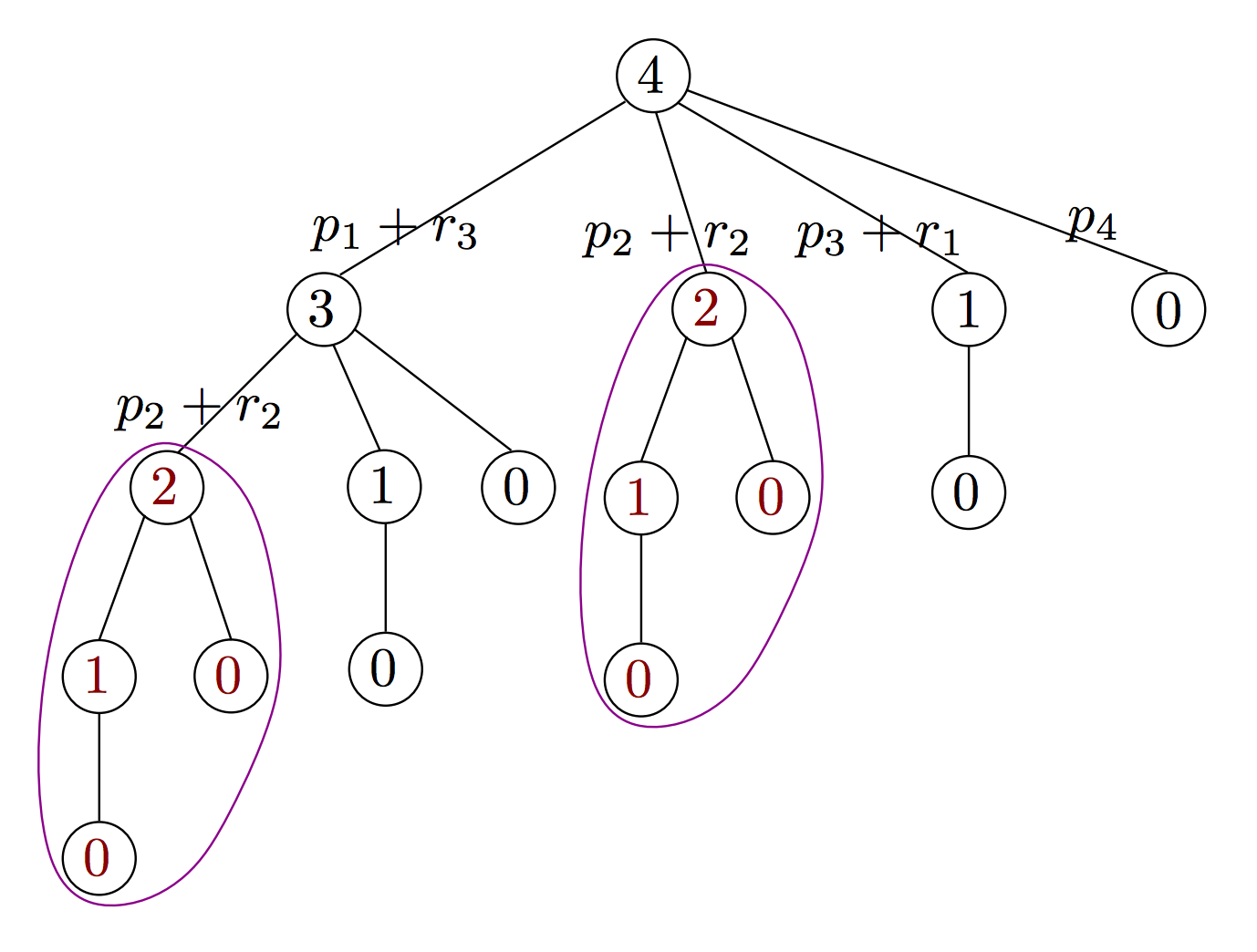
= max(9, 10, 9, 9)

= 10

Another approach: With r0 = 0, rn = max(pi + rn−i) for 1 ≤ i ≤ 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. Once you have the algorithm working, using additional storage, output the number of cuts required.

pricesi = {0, 1, 5, 8, 9, 10, 17, 17, 20}

**Max revenue for length** n**:** 0 1 5 8 10 13 17 18 22

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