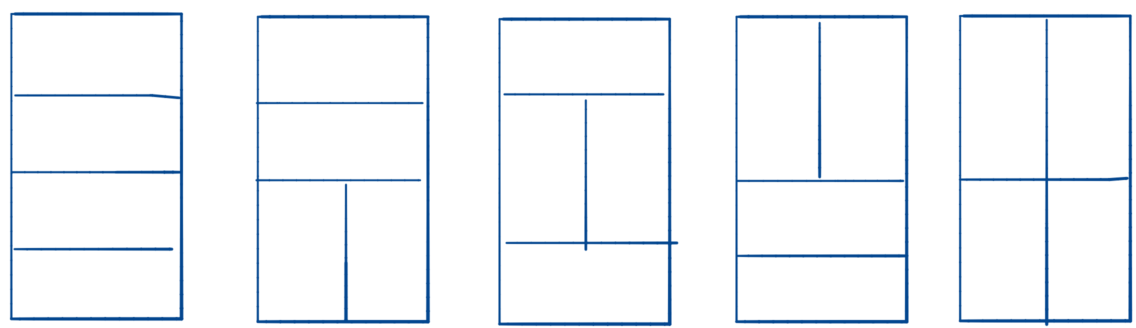
Recitation (IV): Dynamic Programming

Ex 1: Counting Number of Domino Tilings.

Input: n

Output: Number of ways to cover $n \times 2$ grid using domino (1×2 rectangle) tiles.

Exmp: $n = 4$

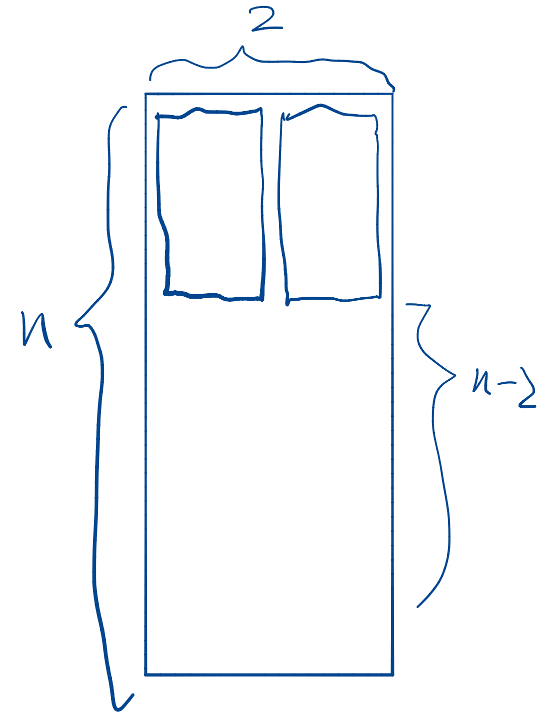


▷ DP Alg

- Subproblem?
  - Consider the direction of the 1st domino.

- Let $opt[i] :=$ # of ways to tile $i \times 2$ grid

$$\Rightarrow \quad opt[i] = \underbrace{opt[i-1]}_{\text{1st domino horizontal}} + \underbrace{opt[i-2]}_{\text{1st domino vertical}}$$

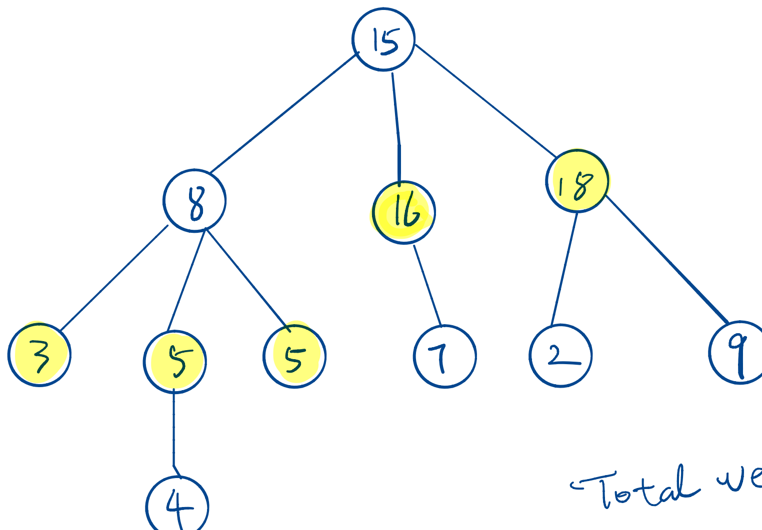


Ex 2: Max-weight Indep Set on Trees

Input: a tree of n nodes, each node i has weight w_i.

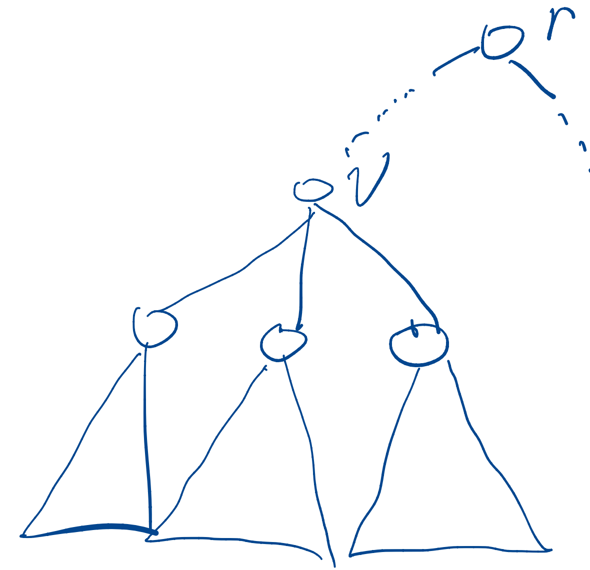
Output: An indep set of the tree with the largest weight

Exmp:



Total weight = 47

▷ DP Alg



- Sub problem?

opt[v, 1] := max wt of indep set rooted at vertex v, but does not include v.

opt[v, 2] := max wt of indep set rooted at vertex v, but does include v.

$$\Rightarrow \quad opt[v,1] = \sum_{u:\text{ child of } v} \max\{opt[u,1], opt[u,2]\}$$

$$opt[v,2] = w_v + \sum_{u:\text{ child of } v} opt[u,1]$$

Ex 3: Coin Change

Input: Seq of int $C_1 < C_2 < C_3 < \cdots < C_n$ as coin types.

Integer M

Output: # of distinct ways to make a M-cents change, assuming unlimited supply of coins

Exmp: $c_1 = 1$, $c_2 = 2$, $c_3 = 4$, $M = 6$.

Then $M = 6 \times c_1 = 4 \times c_1 + c_2 = 2 \times c_1 + 2 \times c_2$

$= 3 \times c_2 = 2 \times c_1 + c_3 = c_2 + c_3$

$\therefore$ 6 ways in total.

▷ DP Alg:

- Sub problems?

  Use less coin types, make $M' (< M)$ -cents change.

$opt[i, j] :=$ # of distinct ways to make j-cents change using only coin types $c_1, \cdots, c_i$

$$opt[i,j] = \begin{cases} 0, & \text{if } i=0 \\ 1, & \text{if } j=0 \\ opt[i-1, j], & \text{if } j < c_i \\ opt[i-1, j] + opt[i, j-c_i], & \text{otherwise.} \end{cases}$$

$opt[n, M]$