DP problem from Practice Midterm

Notation:

$\overset{n}{\vdash\!\!\!\text{———}\!\!\!\dashv}$: rope of length n

$\overset{n-1}{\vdash\!\!\!\text{——}\!\!\!\dashv}$: rope of length $n-1$

$\vdots$

$\overset{1}{\vdash\!\!\dashv}$: rope of length 1

$\overset{0}{\bullet}$: rope of length 0

- First thing to observe is that:

$$
\begin{aligned}
\text{optcut}\left(\overset{n}{\vdash\!\!\!\text{———}\!\!\!\dashv}\right)
&= \max\Big[\text{price}\left(\overset{n}{\vdash\!\!\!\text{——}\!\!\!\dashv}\right) + \text{optcut}\left(\overset{0}{\bullet}\right),\\
&\qquad \text{price}\left(\overset{1}{\vdash\!\!\dashv}\right) + \text{optcut}\left(\overset{n-1}{\vdash\!\!\!\text{——}\!\!\!\dashv}\right),\\
&\qquad \dots\\
&\qquad \text{price}\left(\overset{n-1}{\vdash\!\!\!\text{——}\!\!\!\dashv}\right) + \text{optcut}\left(\overset{1}{\vdash\!\!\dashv}\right)\Big]\\
&= \max\Big[\text{price}\left(\overset{1}{\vdash\!\!\dashv}\right) + \text{optcut}\left(\overset{n-1}{\vdash\!\!\!\text{——}\!\!\!\dashv}\right),\\
&\qquad \dots\\
&\qquad \text{price}\left(\overset{n-1}{\vdash\!\!\!\text{——}\!\!\!\dashv}\right) + \text{optcut}\left(\overset{1}{\vdash\!\!\dashv}\right),\\
&\qquad \text{price}\left(\overset{n}{\vdash\!\!\!\text{——}\!\!\!\dashv}\right) + \text{optcut}\left(\overset{0}{\bullet}\right)\Big]
\end{aligned}
$$

just moved the first sum to the end

with a base case of $\text{optcut}\left(\overset{0}{\bullet}\right) = 0$

- Example of one of the recursive calls:

$$\text{optcut}\left(\overset{n-1}{\vdash\!\!\!\longrightarrow\!\!\!\dashv}\right)$$

$$= \max\Big[\text{price}\left(\overset{1}{\vdash\!\!\!\longrightarrow\!\!\!\dashv}\right) + \text{optcut}\left(\overset{\overset{= n-2}{n-1-1}}{\vdash\!\!\!\longrightarrow\!\!\!\dashv}\right),$$

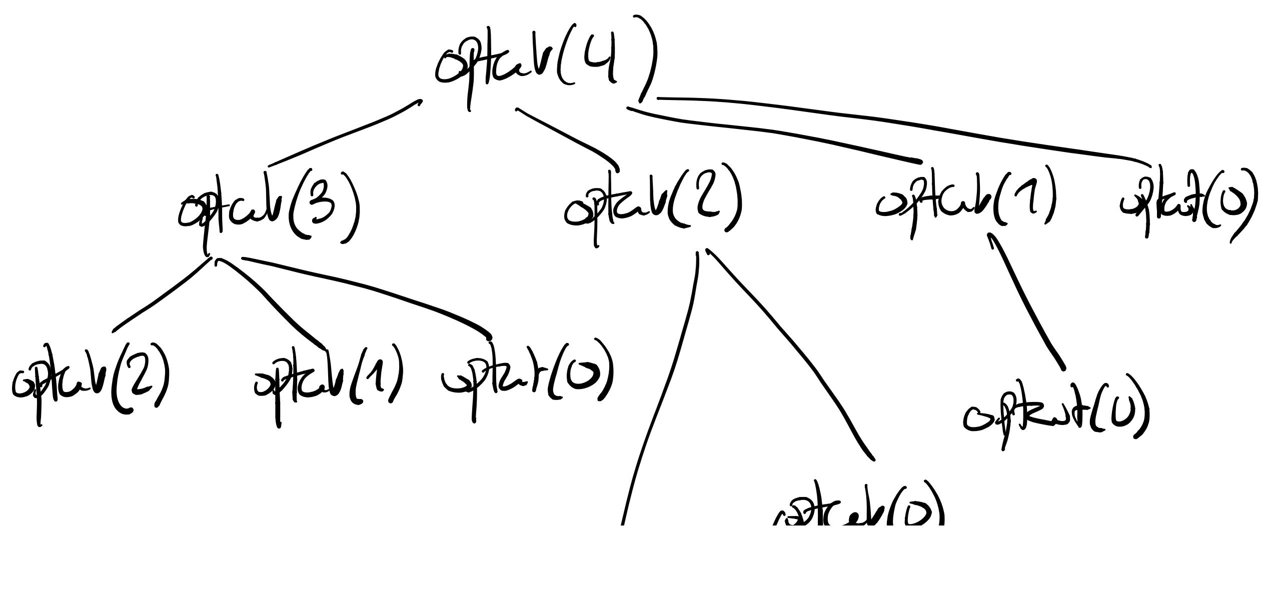
$$\ldots$$

$$\text{price}\left(\overset{\overset{= n-2}{n-1-1}}{\vdash\!\!\!\longrightarrow\!\!\!\dashv}\right) + \text{optcut}\left(\overset{1}{\vdash\!\!\!\longrightarrow\!\!\!\dashv}\right),$$

$$\text{price}\left(\overset{n-1}{\vdash\!\!\!\longrightarrow\!\!\!\dashv}\right) + \text{optcut}\left(\overset{0}{\bullet}\right)\Big]$$

- Let's look at the recursion tree for $n = 4$ to get some intuition:

To save space on the page, we'll write optcut(n) for optcut$\left(\overset{n}{\vdash\!\!\!\longrightarrow\!\!\!\dashv}\right)$ etc...



optcut(1)

|

optcut(0)

- What is the (topological) order with which we need to solve the problems?

  We'll have to start with optcut(0), then optcut(1), then optcut(2), ...

$$\begin{array}{ccccccc} \text{optcut}(0) & \longrightarrow & \text{optcut}(1) & \longrightarrow & \text{optcut}(2) & \longrightarrow & \text{optcut}(3) \\ & & & & & & \downarrow \\ & & & & & & \text{optcut}(4) \end{array}$$

- Let's design an <u>iterative</u> algorithm that will solve the sub-problems in topological order and store the store the solutions in an array A so that we can reuse them

- High-level idea:

. we'll solve optcut(0) and store it in A[0]
. " " optcut(1) " " " " A[1]
. " " optcut(2) " " " " A[2]

⋮

. " " optcut(n) " " " " A[n+1]

• The array A will look like

A = | optcut(0) | optcut(1) | optcut(2) | optcut(3) | optcut(4) | . . .

$i=0$

0

$i=1$

$\max[\text{price}(\overset{j}{1})+\text{optcut}(\overset{i-j}{1-1})]$

$i=2$

$\max[\text{price}(\overset{j}{1})+\text{optcut}(\overset{i-j}{2-1}),$
$\text{price}(2)+\text{optcut}(2-2)]$

$i=3$

$\max[\text{price}(\overset{j}{1})+\text{optcut}(\overset{i-j}{3-1}),$
$\text{price}(2)+\text{optcut}(3-2),$
$\text{price}(3)+\text{optcut}(3-3)]$

$i=4$

$\max[\text{price}(\overset{j}{1})+\text{optcut}(\overset{i-j}{4-1}),$

price(2) + optcut(4-2),

price(3) + optcut(4-3),

price(4) + optcut(4-4)]

- Now let's write this up in pseudo-code

<u>optcut(n)</u>

```
. A is array of size n+1
. for i = 0 to n
    . A[i] = 0
. for i = 1 to n
    . for j = 1 to i
        . A[i] = max[A[i], price(j) + A[i-j]]
. return A[n]
```

Here we are computing the max "incrementally". The max for optcut(i) is computed "incrementally" over the "i loop".