CS 3510

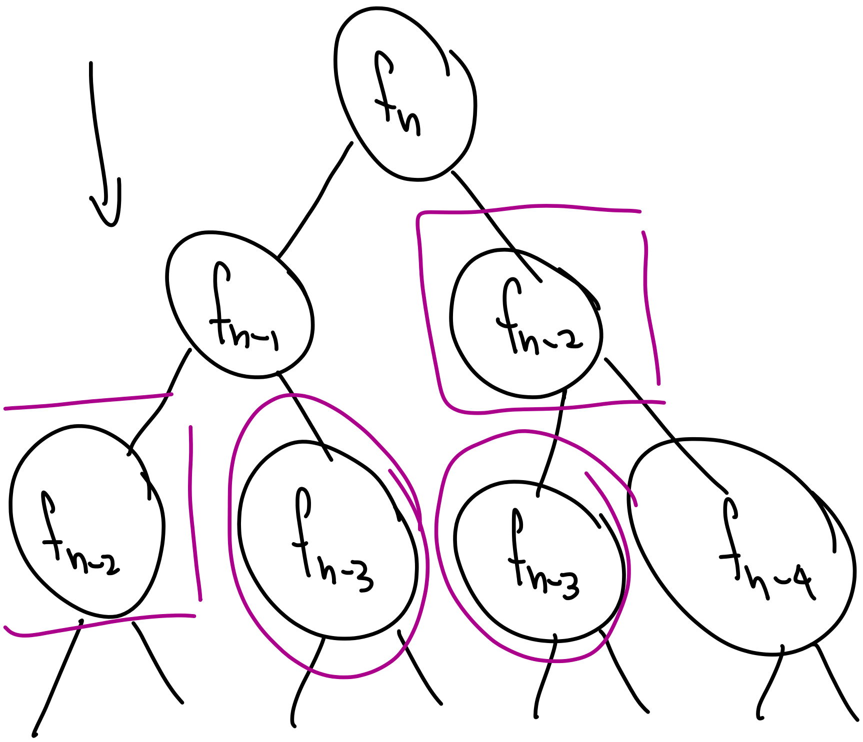
Tuesday, September 13, 2022 8:23 AM

* Dynamic Programming

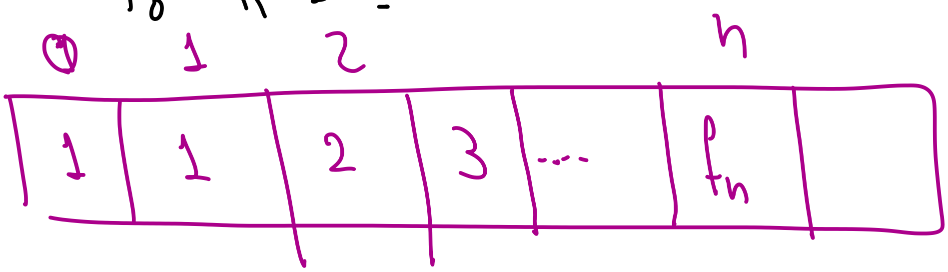
Warm up: Fibonacci

$f_0 = f_1 = 1$

$f_n = f_{n-1} + f_{n-2} \quad \forall n \geq 2$



$f_0 = f_1 = 1$!

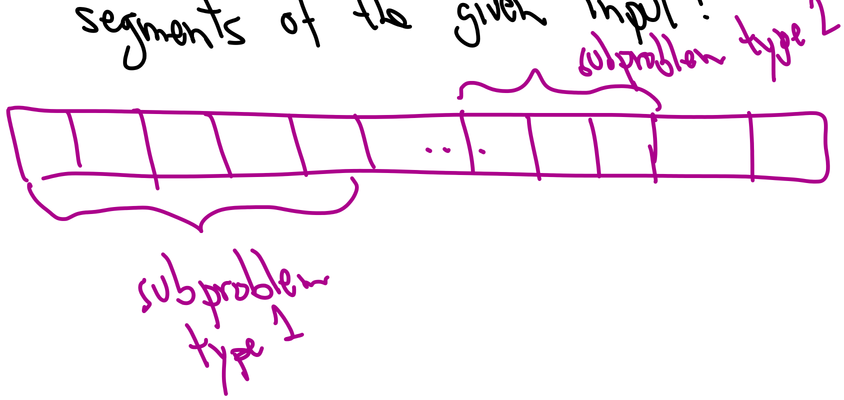


Sequencing DP

· Input is a sequence (or two)

- Subproblems / DP states :

  segments of the given input!



Longest Increasing Subsequence (LIS)

Input: $S = [s_1, s_2, \dots, s_n]$ $s_i \in \mathbb{R}$

Output: len of a LIS.

<u>Note</u>: a subsequence of a given sequence

$S = [s_1, s_2 \dots s_n]$ is

$$[s_{i_1}, s_{i_2}, s_{i_3}, \dots, s_{i_k}]$$

$$\le \quad \le \quad \le$$

such that:

- $s_{i_j} \in S$
- $i_1 < i_2 < i_3 < \dots < i_k$

Ex: [4, 1, 0, -2, 6, 5]

[4, 0] ✓

[1, 0, -2, 5] ✓

[6] ✓

[6, 0, 4] X

[4, 6] is Increasing!!!

Back to LIS

Input:

2	3	1	-4	1	2	6
1	2	2	2	2	3	4
1	2	1	1	2	3	4

T[k]

Step 1: T[k] is the len of a LIS of $[s_1, s_2, \dots, s_k]$, $1 \le k \le n$. $\boxed{\text{ending at } s_k}$.

Step 2: Write a recursive relation between subproblems

$$T[k] = \max\{\, 1 + T[j] \text{ if } S_k > S_j \quad j < k \,\}$$

Base case

$T[0] = 0 \qquad S_0 = -\infty$