

Dynamic Programming

Fibonacci Sequence

$$F_1 = 1 ; F_2 = 1$$

$$F_n = F_{n-1} + F_{n-2}$$

(A) Recursive Solution

rec-fib(n)

EXPONENTIAL

if $n \leq 2$ then
return 1

return rec-fib($n-1$) + rec-fib($n-2$)

(B) Dynamic Programming

MEMOIZED mem-fib(n, memo)

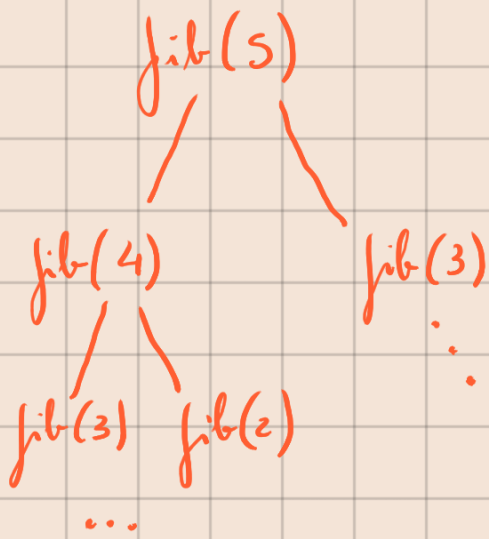
if memo[n] then
return memo[n]

if $n \leq 2$ then
return 1

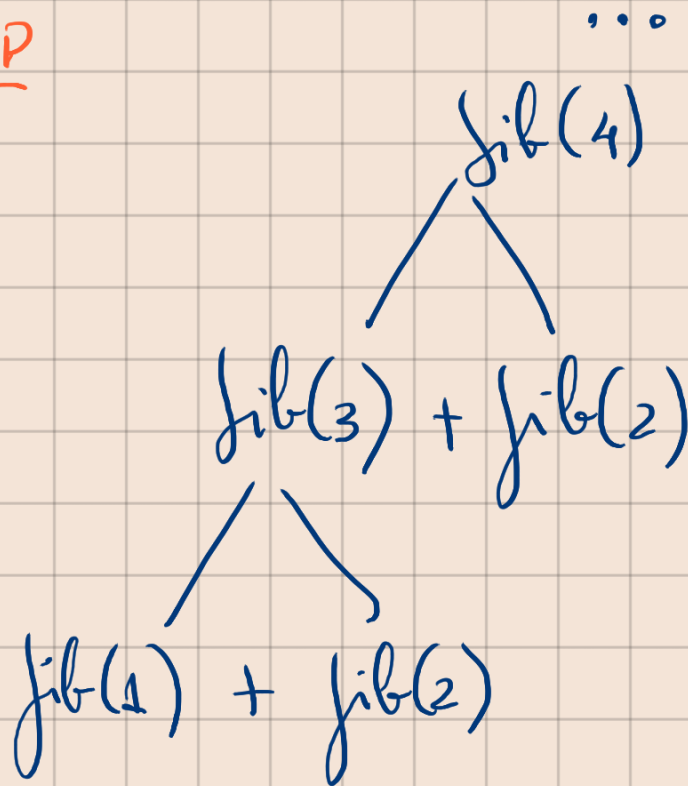
fib ← mem-fib($n-1$, memo) + mem-fib($n-2$, memo)

memo[n] ← fib

return fib



BOTTOM-UP



2.

a) Variables

m - value of change

v_1, \dots, v_n - values of the avail. coins

x_1, \dots, x_n - numbers of coins per value

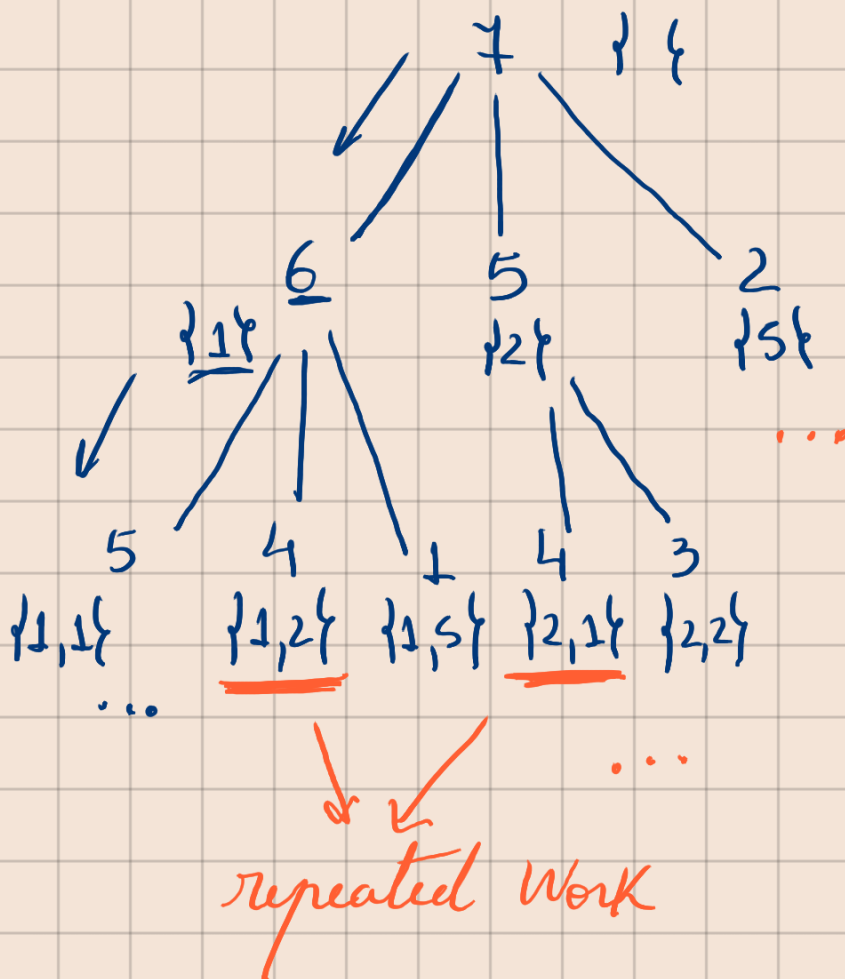
Objective

$$\min \sum_{i=1}^n x_i$$

Restrictions

$$\sum_{i=1}^n x_i v_i = m$$

TOP-DOWN : $m=7$, coins = $[1, 2, 5]$



BOTTOM-UP:

	$m=0$	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$
mincoins	0	1	1	2	2	1	2	2
lastcoin	0	1	2	2	2	5	5	5

i th coin denomination
 K change (amount)

$$\text{mincoins}(i, K) = \begin{cases} 0, & \text{if } K=0 \\ m+1, & \text{if } K>0 \wedge i=0 \\ \text{mincoins}(i, K-v_i)+1, & v_i \leq K \wedge \text{mincoins}(i, K-v_i)+1 < \text{mincoins}(i-1, K) \\ \text{mincoins}(i-1, K), & \text{else} \end{cases}$$

$$\text{lastcoin}(i, K) = \begin{cases} 0, & \text{if } K=0 \\ 0, & \text{if } K>0 \wedge i=0 \\ v_i, & \text{if } v_i \leq K \wedge \text{mincoins}(i, K-v_i)+1 < \text{mincoins}(i-1, K) \\ \text{lastcoins}(i-1, K), & \text{else} \end{cases}$$