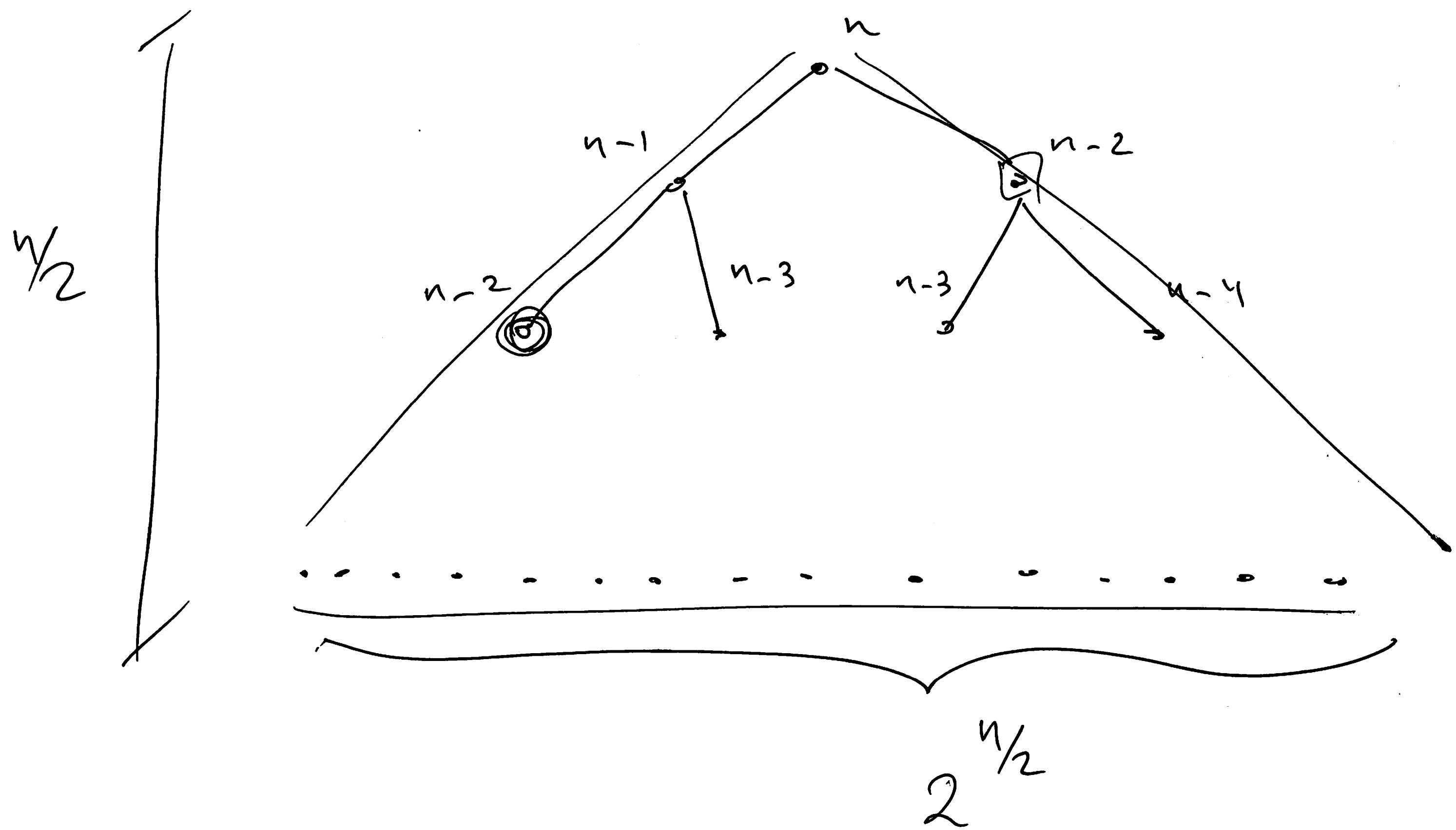
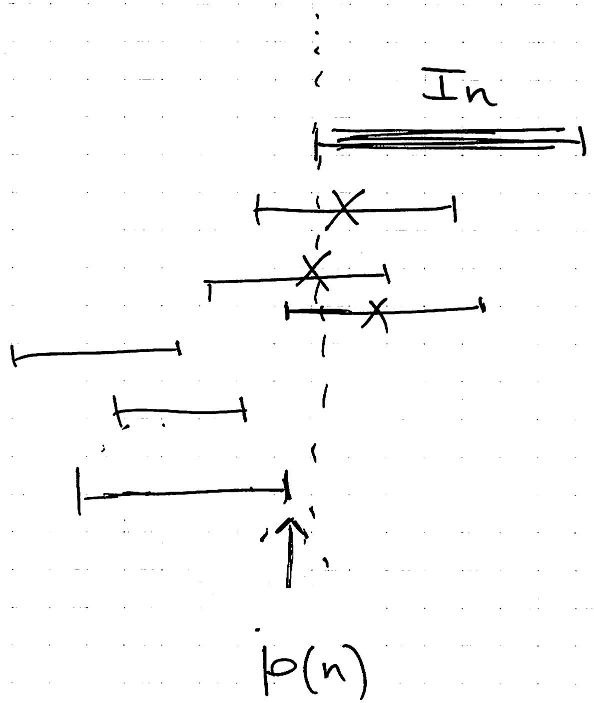
$T(n)$ = time complexity of fib(n)

$$T(n) = T(n-1) + T(n-2) + O(1)$$



$$T(n) = \Omega(2^{n/2})$$

$n \in OPT(n)$



First try (won't work)

① Define subproblems

$M[i]$ = length of L.I.S. of $A[1], \dots, A[i]$

② Derive recurrence

If $i \notin$ L.I.S. for $M[i]$

$$M[i] = M[i-1]$$

If $i \in$ L.I.S. for $M[i]$

$$M[i] = M[i-1] + 1$$

n	n-1	n-2	...	2	1

$M[n]$

X wrong!

$$\Longrightarrow \quad M[i] = \max\left(M[i-1],\ 1 + M[i-1]\right)$$

$$M[0] = 0$$

Second try (this on works!)

① Define subproblems

M[i] = the lenght of L.I.S. in A[1], ... A[i] that finishes with A[i]

② Find recurrence

condition on element before A[i] is L.I.S.

let A[j] be that element $\quad j < i$, $A[j] < A[i]$

$M[i] = 1 + M[j]$

who is j ?

$$\begin{cases} M[i] = 1 + \max\limits_{\substack{j<i \\ A[j] < A[i]}} M[j] \\ M[0] = 0 \end{cases}$$

#DP states = n+1

time per state = O(n)

total time = $O(n^2)$

< 1 2 3 4 ... n-1 0 >

M[n] = 1

M[n-2] = n-1

First try (wrong!)

① $M[i]$ = value of optimal solution among first i items

② If $i \notin$ OPT for $\{1, \dots, i\}$

$$M[i] = M[i-1]$$

If $i \in$ OPT for $\{1, \dots, i\}$

$$M[i] = v(i) + M[i-1]$$

$$\Rightarrow M[i] = \max\left(M[i-1], v(i) + M[i-1]\right)$$

X wrong!

Second try (this works!)

① $M[i, C]$ = value of OPT solution using knapsack of cap. C using items $\{1, .., i\}$

② If $i \notin OPT(i, C)$

$$M[i, C] = M[i-1, C]$$

If $i \in OPT(i, C)$

$$M[i, C] = v(i) + M[i-1, C - w(i)]$$

$$\Rightarrow M[i, C] = \begin{cases} \max(M[i-1, C], v(i) + M[i-1, C - w(i)]) & \text{if } w(i) \le C \\ M[i-1, C] & \text{if } w(i) > C \end{cases}$$

$M[0, C] = 0 \ \forall C$

③ Time complexity

$\#$DP states $= (n+1)(W+1)$

each takes $= O(1)$

total time $= O(nW)$

where is answer?

$$M[n, W]$$