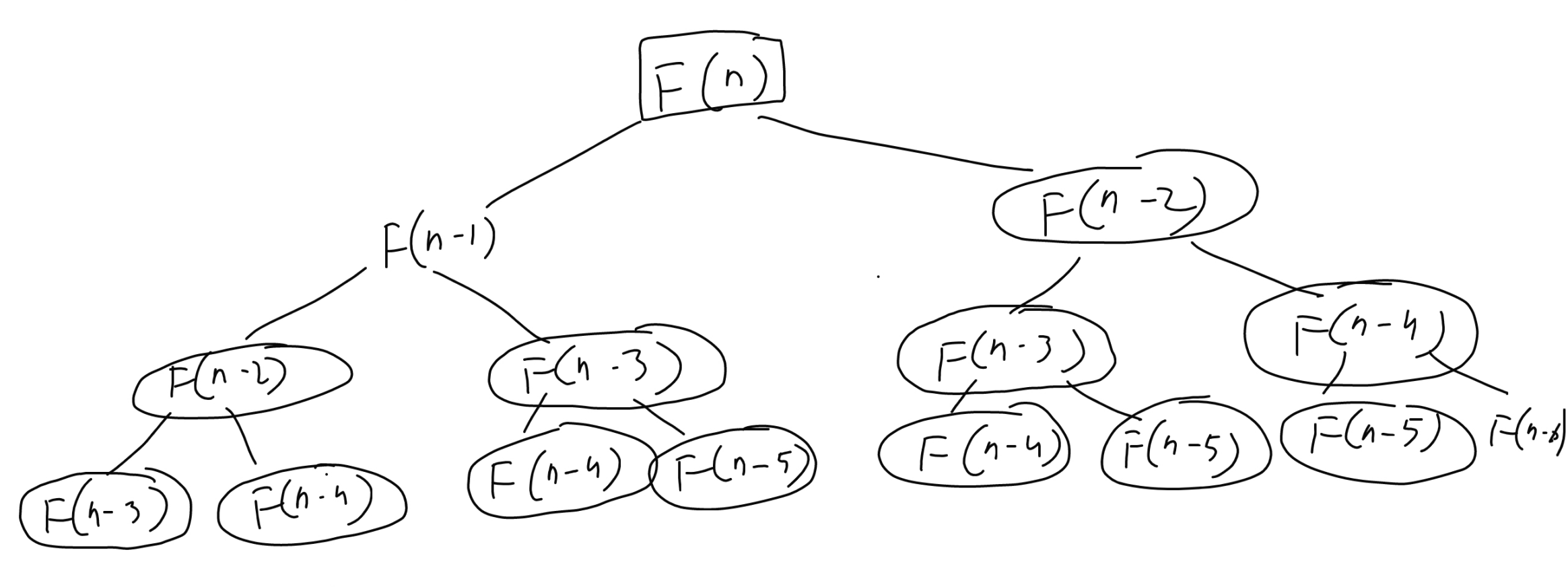
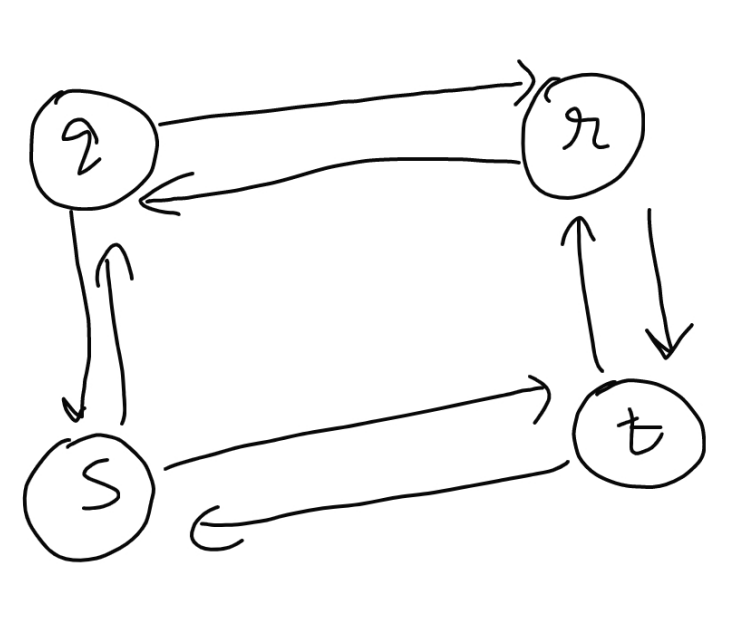
$\underline{Fib(n)} = \underbrace{F(n-1)}_{\substack{\uparrow \\ \text{subproblem-1}}} + \underbrace{F(n-2)}_{\substack{\uparrow \\ \text{subproblem-2}}}$


F(n)
F(n-1)
F(n-2)
F(n-2)
F(n-3)
F(n-3)
F(n-4)
F(n-3)
F(n-4)
F(n-4)
F(n-5)
F(n-4)
F(n-5)
F(n-5)
F(n-6)

⇑
Overlapping subproblems



(LP) from
Longest path ~~btw~~ 'q' to 't'
$(q \to r \to t)$ or $(q \to s \to t)$

$\overset{(3)}{LP(q \text{ to } t)} \neq \overset{(4)}{LP(q \text{ to } r)} + \overset{(4)}{LP(r \text{ to } t)}$

$LP(q \text{ to } r) \Downarrow (q \to s \to t \to r)$

$LP(r \text{ to } t) \Downarrow (r \to q \to s \to t)$

Over here, the optimal solution to the subproblems did not result into the optimal solution of the actual problem.
There fore, this problem can not be solved using DP.

There are 2 essential conditions for a problem to be solved using Dynamic Programming

① It must have overlapping subproblems.
② Optimal solution to the subproblems must result into the optimal solution of the overall problem.

TOP- DOWN DP

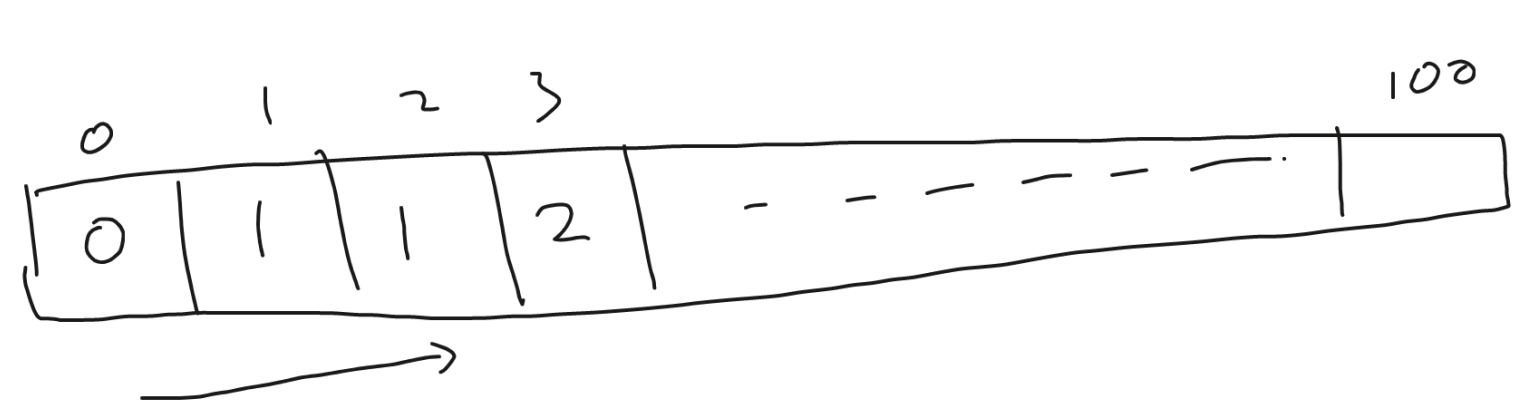
fib(100) = fib(99) + fib(98)

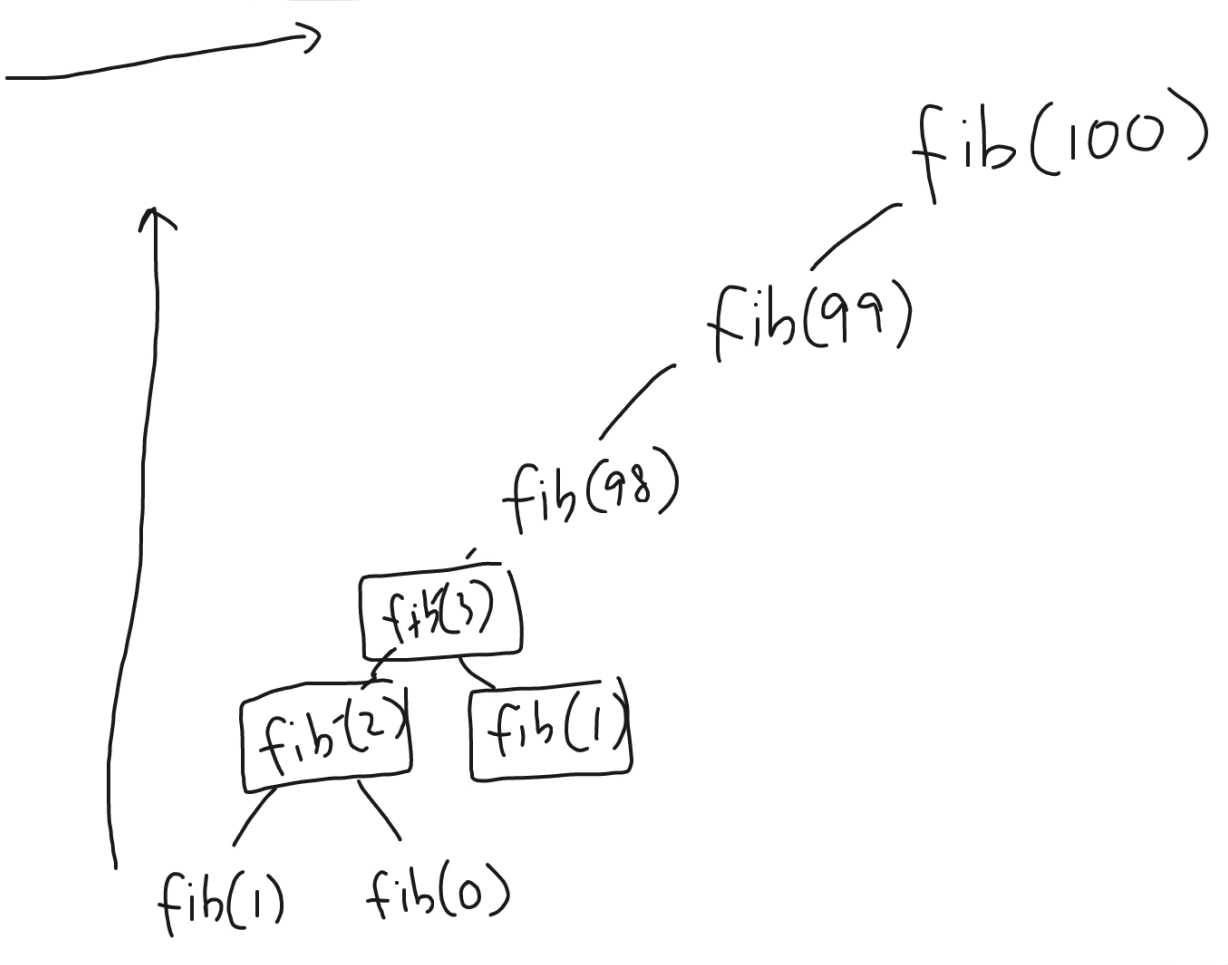
fib(99) → fib(98), fib(97)
fib(98) → fib(97), fib(96)

BOTTOM - UP DP

fib(100)

0 → 1 → 2 → 3 → 4 - - - - - - - - - → 100


0
1
2
3
100
0
1
1
2


fib(100)
fib(99)
fib(98)
fib(3)
fib(2)
fib(1)
fib(1)
fib(0)

$\overset{2\ 2\ 2\ 2\ 2\ 2 \to 2^6}{\text{"ABCDEF"}}$

For a string of length of n, there are 2^n subsequences.

"AE" ✓
"EA" ×
"ACD" ✓
"AEF"
"AA" ×

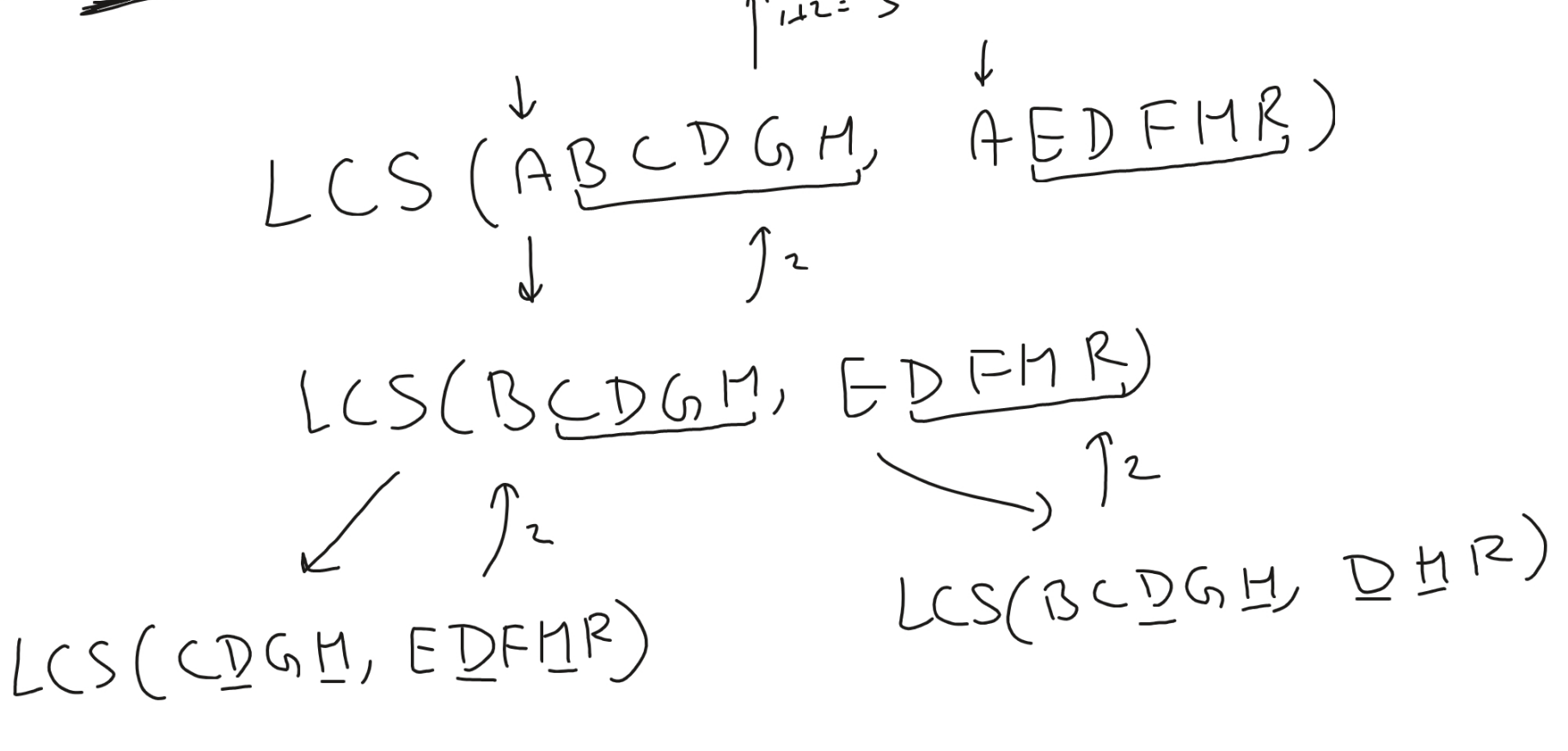
"ABC" → "" A, B, C, AB, AC, BC, ABC
1 2 3 4 5 6 7 8
$= 2^3 = 8$

(n) str1 = 'ABCDGH' 2^6
(m) str2 = "AEDFHR' 2^6
O/P = 3

"ABCHGD"
"AEDFHR"
O/P = 2

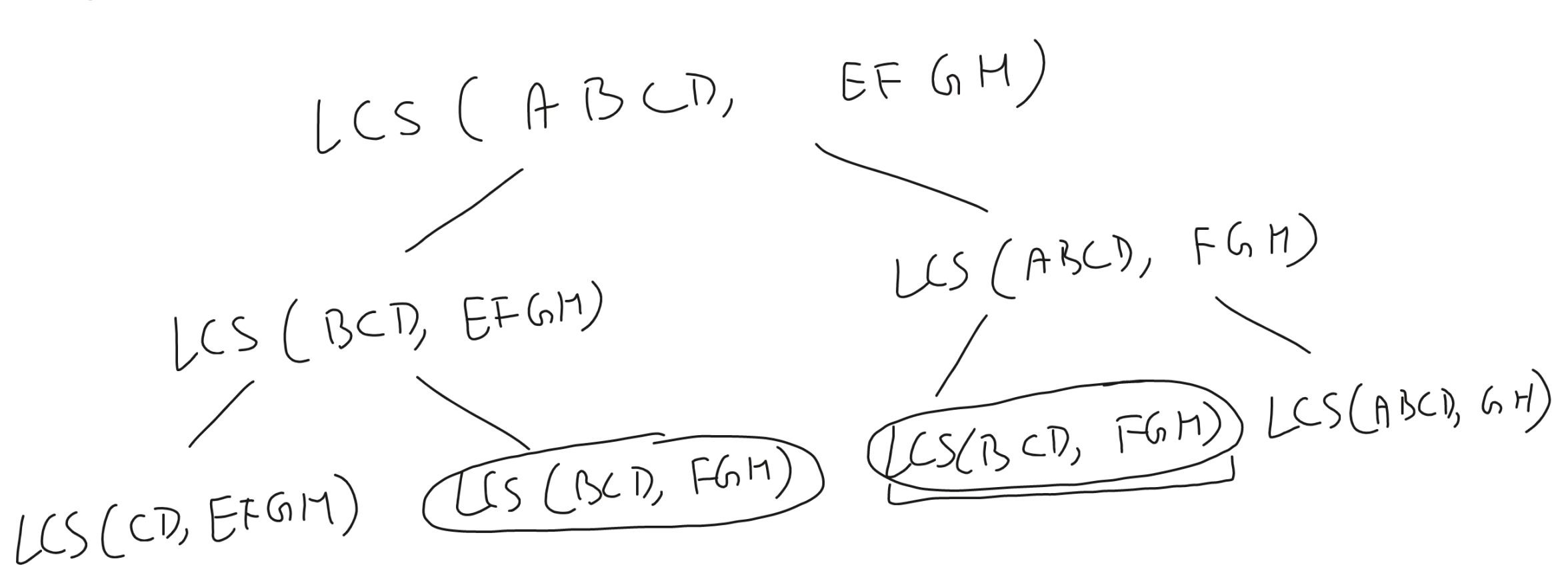
$O(n \cdot 2^n)$ $O(m \cdot 2^m)$

$O(n \cdot 2^n + m \cdot 2^m + \text{pick the common subseq.})$
③ ↑ 1+2 = 3


LCS(ABCDGH, AEDFHR)
LCS(BCDGH, EDFHR)
↑ 2
↑ 2
LCS(CDGH, EDFHR)
↑ 2
↑ 2
LCS(BCDGH, DHR)

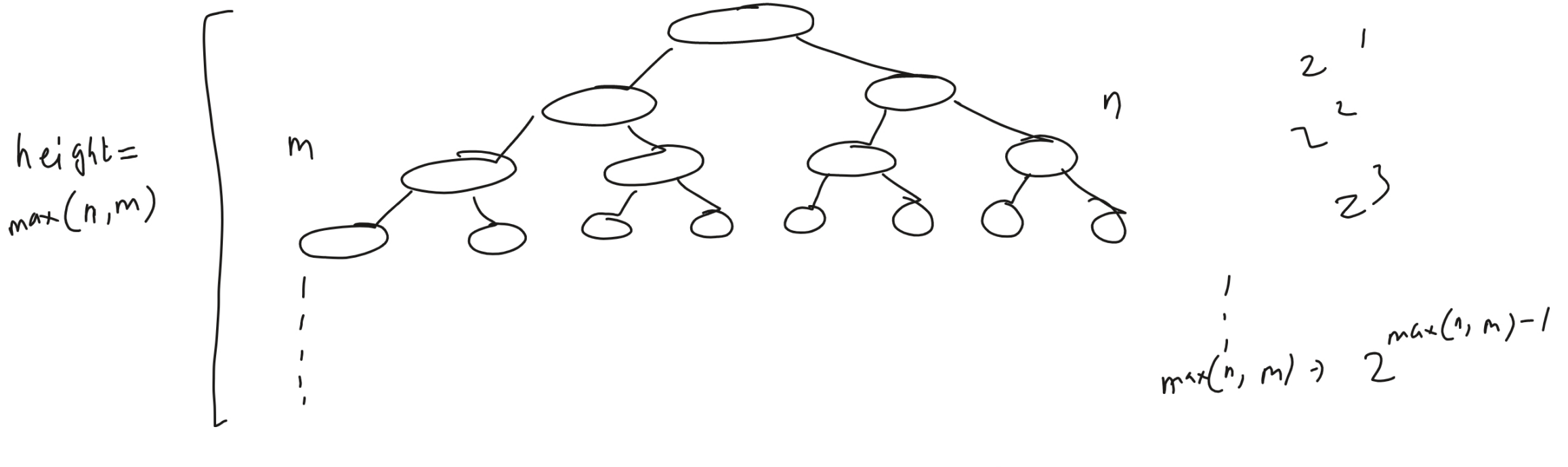
$$\underline{LCS(S1, S2)} = 1 + LCS(S1[1 \ldots n-1], S2[1 \ldots m-1]) \quad [\text{if } (S1[0] == S2[0])]$$

$$= \max(LCS(S1, S2[1 \ldots m-1]), LCS(S1[1 \ldots n-1], S2)) \quad [\text{otherwise}]$$

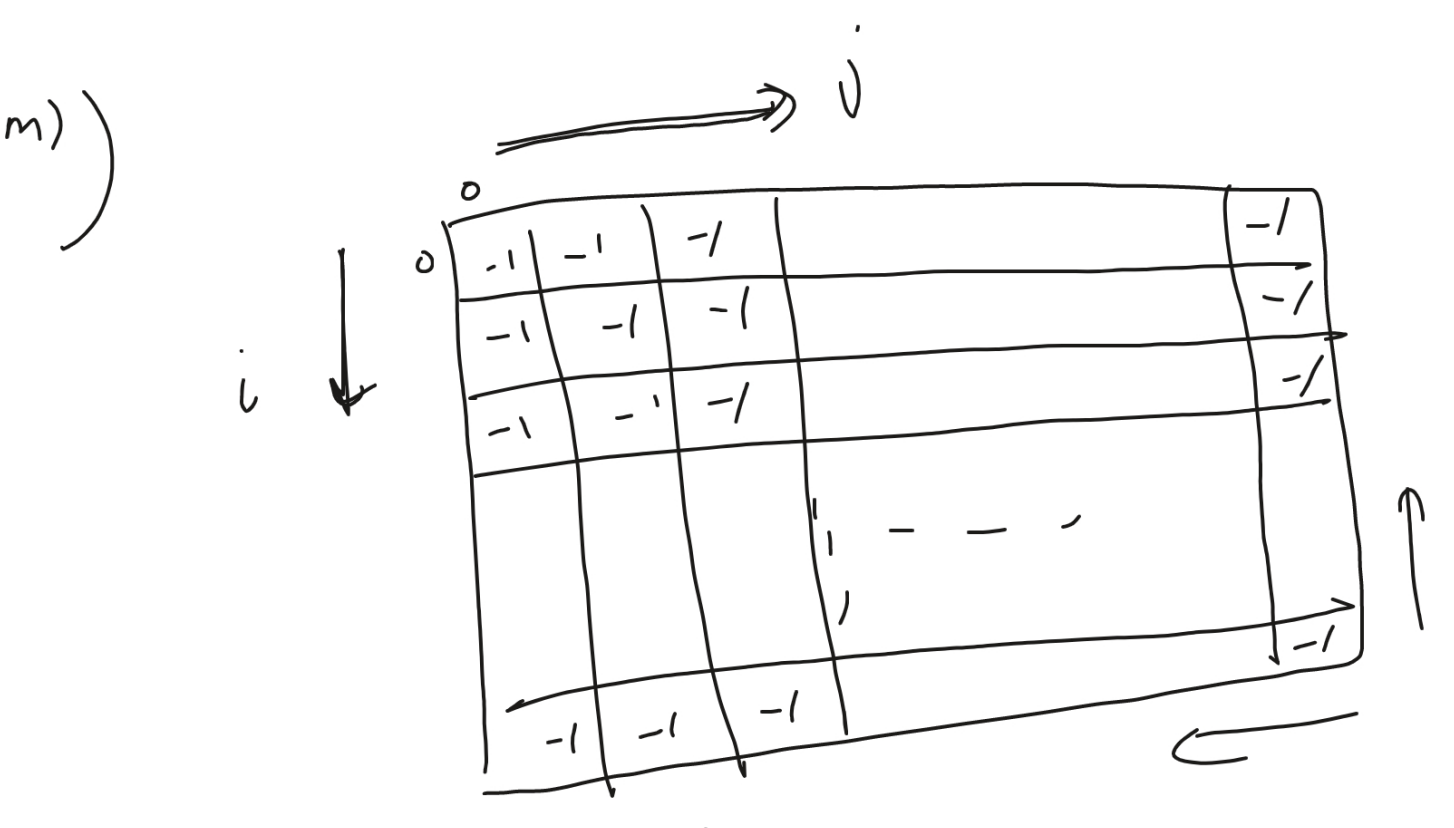

LCS(ABCD, EFGH)
LCS(BCD, EFGH)
LCS(ABCD, FGH)
LCS(CD, EFGH)
LCS(BCD, FGH)
LCS(BCD, FGH)
LCS(ABCD, GH)

Overlapping subproblems.
⓪

LCS("", "ABC") = 0
LCS("ABC", "") = 0


height= max(n,m)
m
n
2^0
2^1
2^2
2^3
$\max(n,m) \to 2^{\max(n,m)-1}$

$TC = O(2^{\max(n,m)})$


j
i
0
0
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1

		0	1	2	3	4	5	6
			A	E	D	F	H	R
0		0	0	0	0	0	0	0
1	A	0	1	1	1	1	1	1
2	B	0	1	1	1	1	1	1
3	C	0	1	①	1	1	1	1
4	D	0	1	1	2	2	2	2
5	G	0	1	1	2	②	2	2
6	H	0	1	1	2	2	3	$\boxed{3}$

$s1[i-1] \mathrel{!=} s2[j-1]$
$\Rightarrow dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$s1[i-1] == s2[j-1]$
$\Rightarrow dp[i][j] = 1 + dp[i-1][j-1]$

return
$\boxed{\Rightarrow dp[n][m]}$