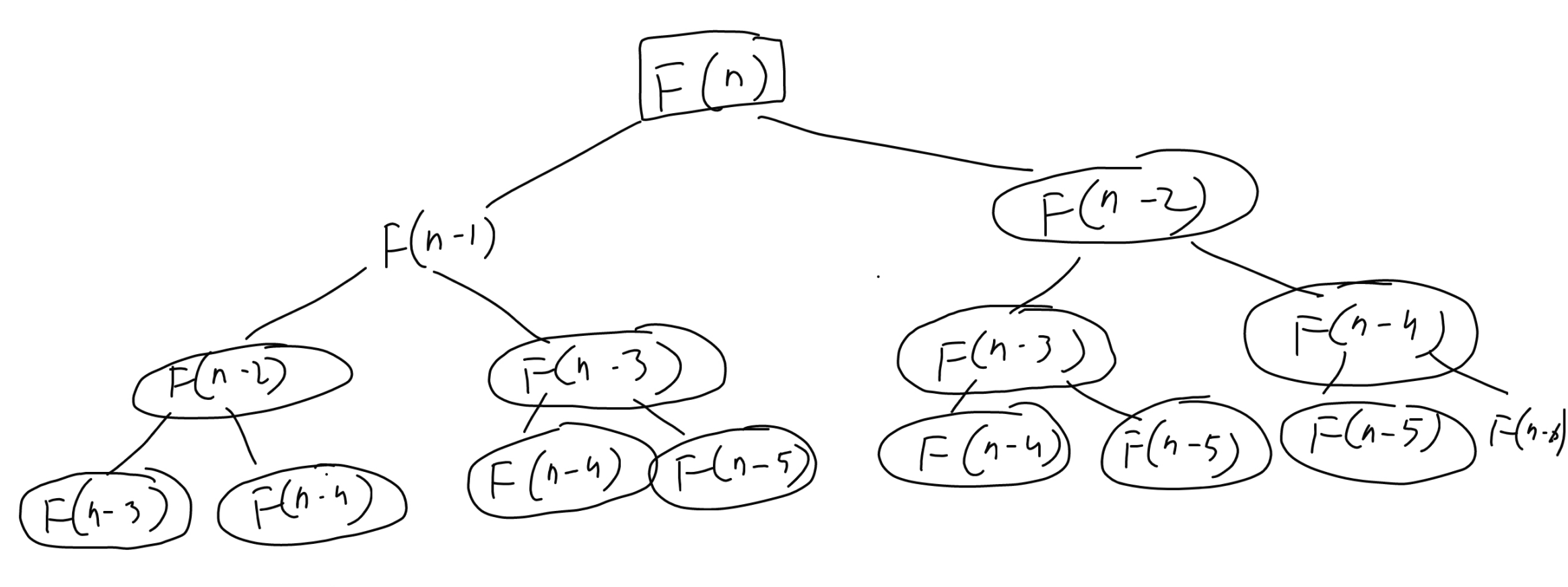
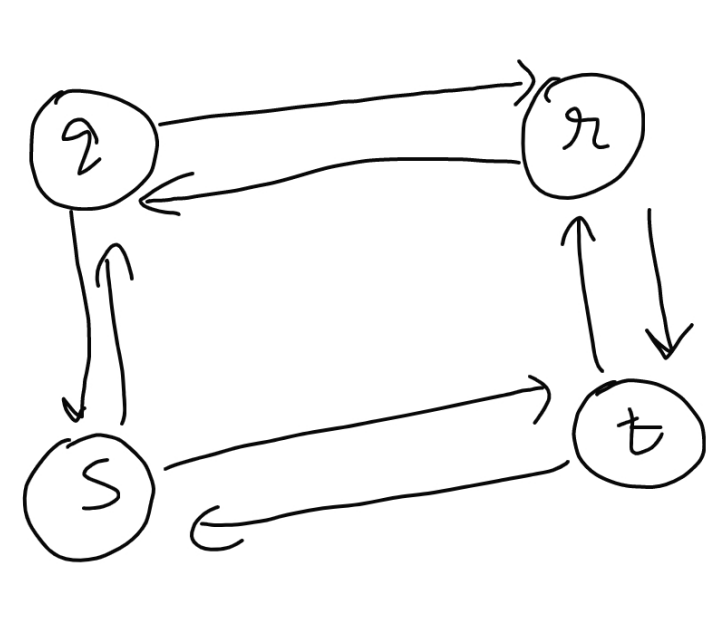
$\underline{Fib(n)} = \underbrace{F(n-1)}_{\uparrow\ \text{subproblem-1}} + \underbrace{F(n-2)}_{\uparrow\ \text{subproblem-2}}$


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

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

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

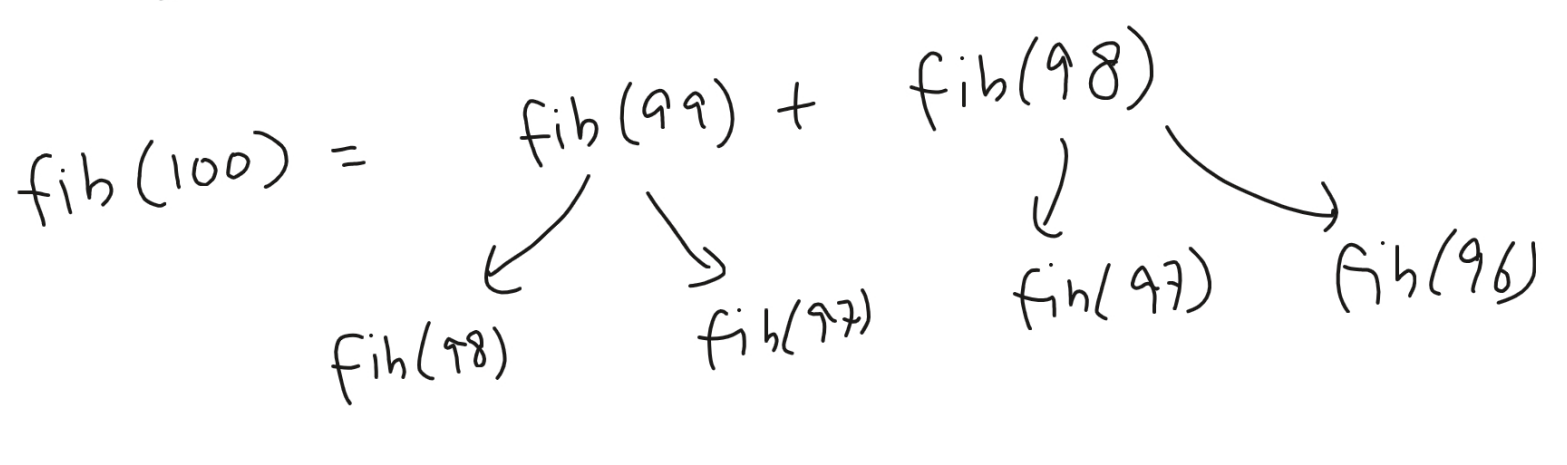
$LP(r \text{ to } t) \Downarrow (r \to q \to s \to t)$

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.

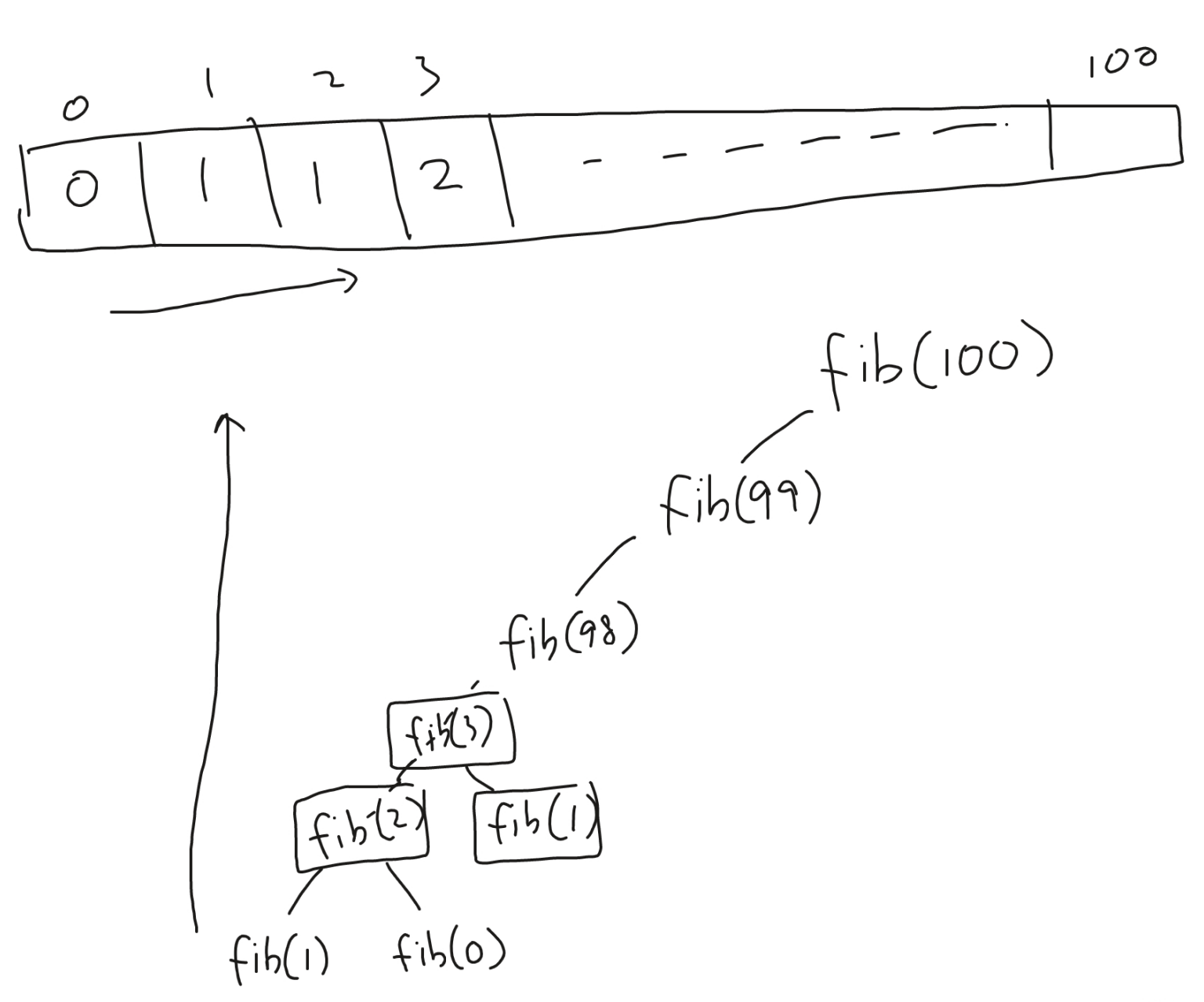
TOP- DOWN DP


fib(100) =
fib(99) + fib(98)
fib(98)
fib(97)
fib(97)
fib(96)

BOTTOM - UP DP

fib(100)

$0 \to 1 \to 2 \to 3 \to 4 \ \text{-------} \to 100$


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)

"ABCDEF" $\quad 2\ 2\ 2\ 2\ 2\ 2 \to 2^6$

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

"AE" ✓ "EA" ×
"ACD" ✓ "AEF" ✓ "AA" ×

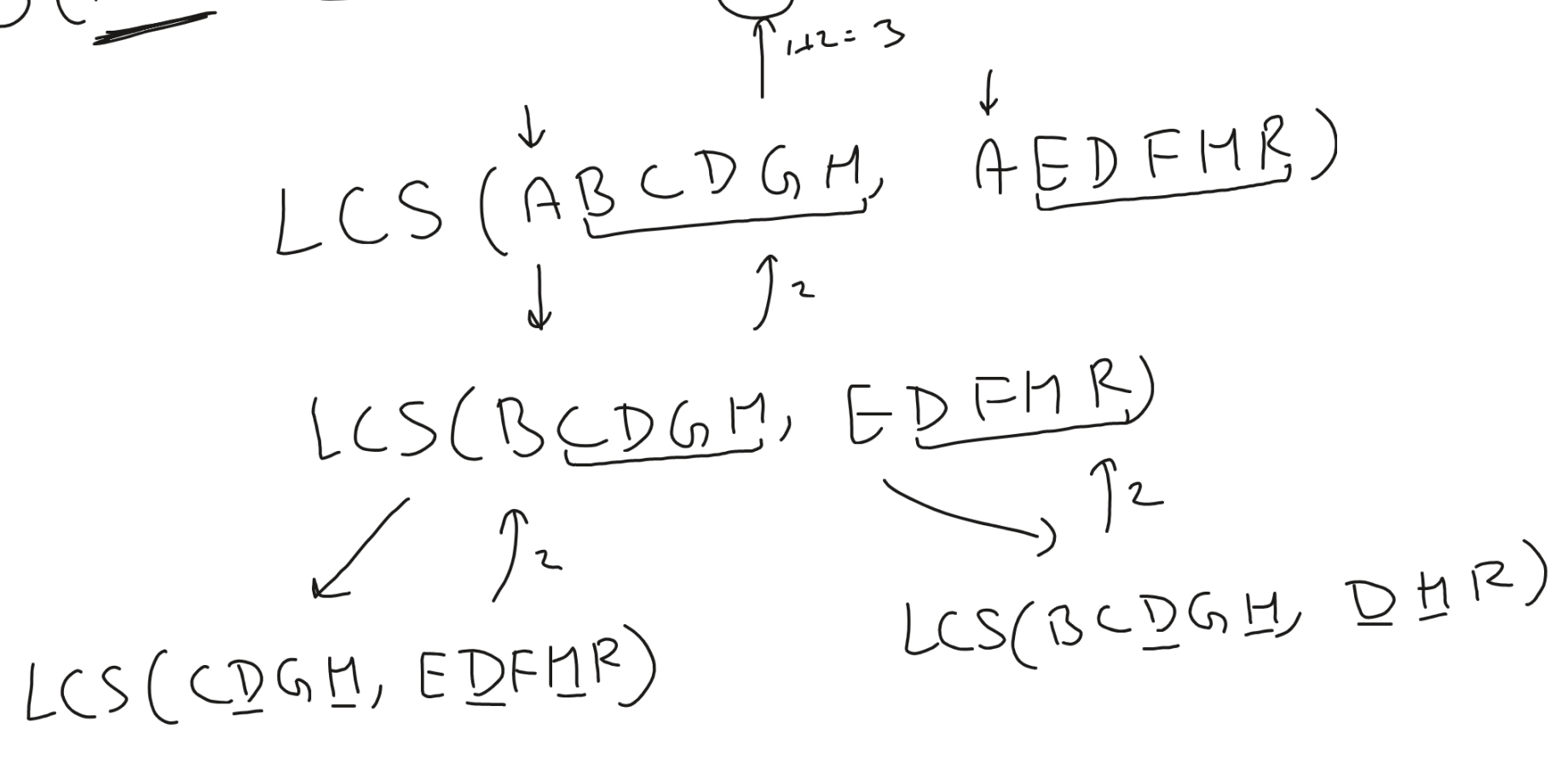
"ABC" → "", A, B, C, AB, AC, BC, ABC
1 2 3 4 5 6 7 8
$= 2^3 = 8$

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

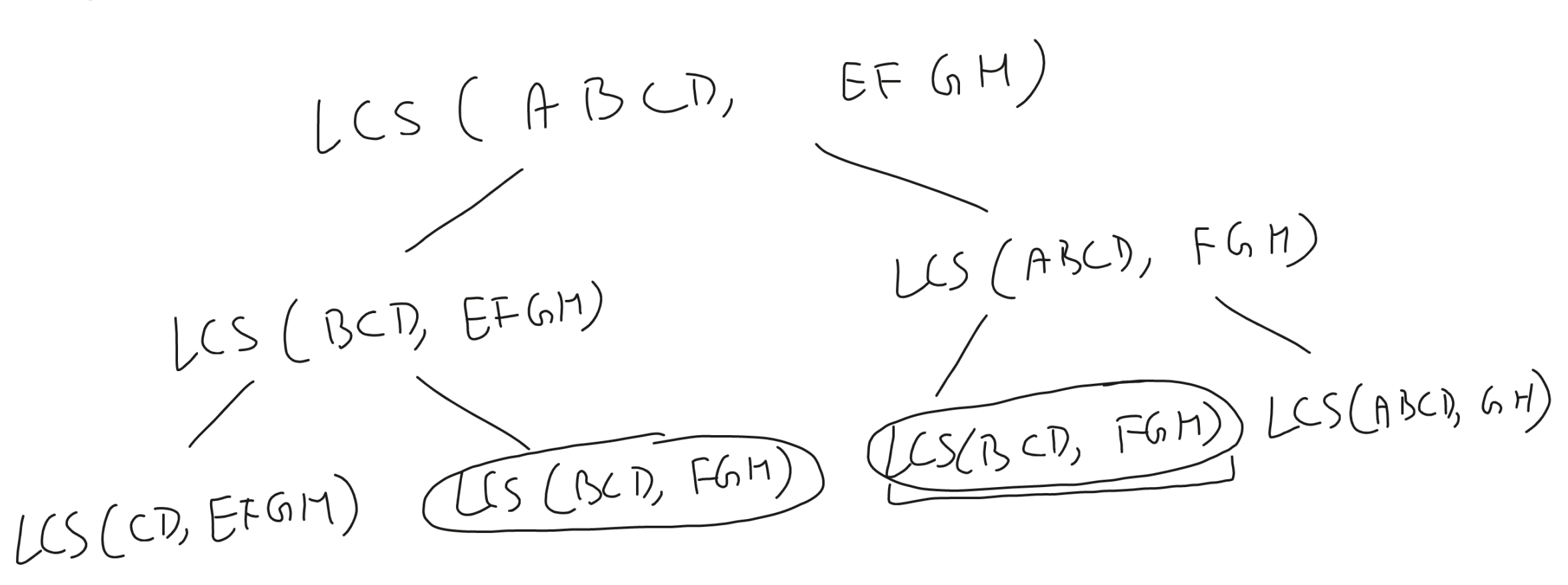
"ABCMGD"
"AEDFMR"
O/P = 2

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

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

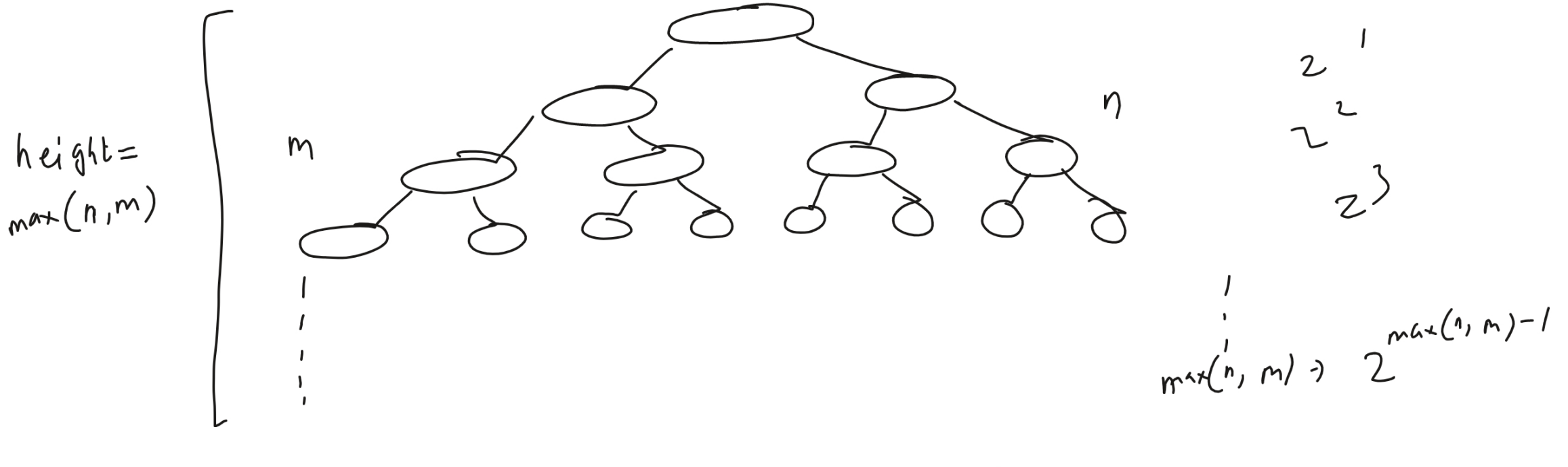

LCS(ABCDGM, AEDFMR)
2
LCS(BCDGM, EDFMR)
2
2
LCS(CDGM, EDFMR)
LCS(BCDGM, DMR)

$$LCS(s1, s2) = \begin{cases} 1 + LCS(s1[1 \dots n-1],\ s2[1 \dots m-1]) & [\text{if } (s1[0] == s2[0])] \\ = \max(LCS(s1,\ s2[1 \dots m-1]),\ LCS(s1[1 \dots n-1],\ s2)) & [\text{otherwise}] \end{cases}$$

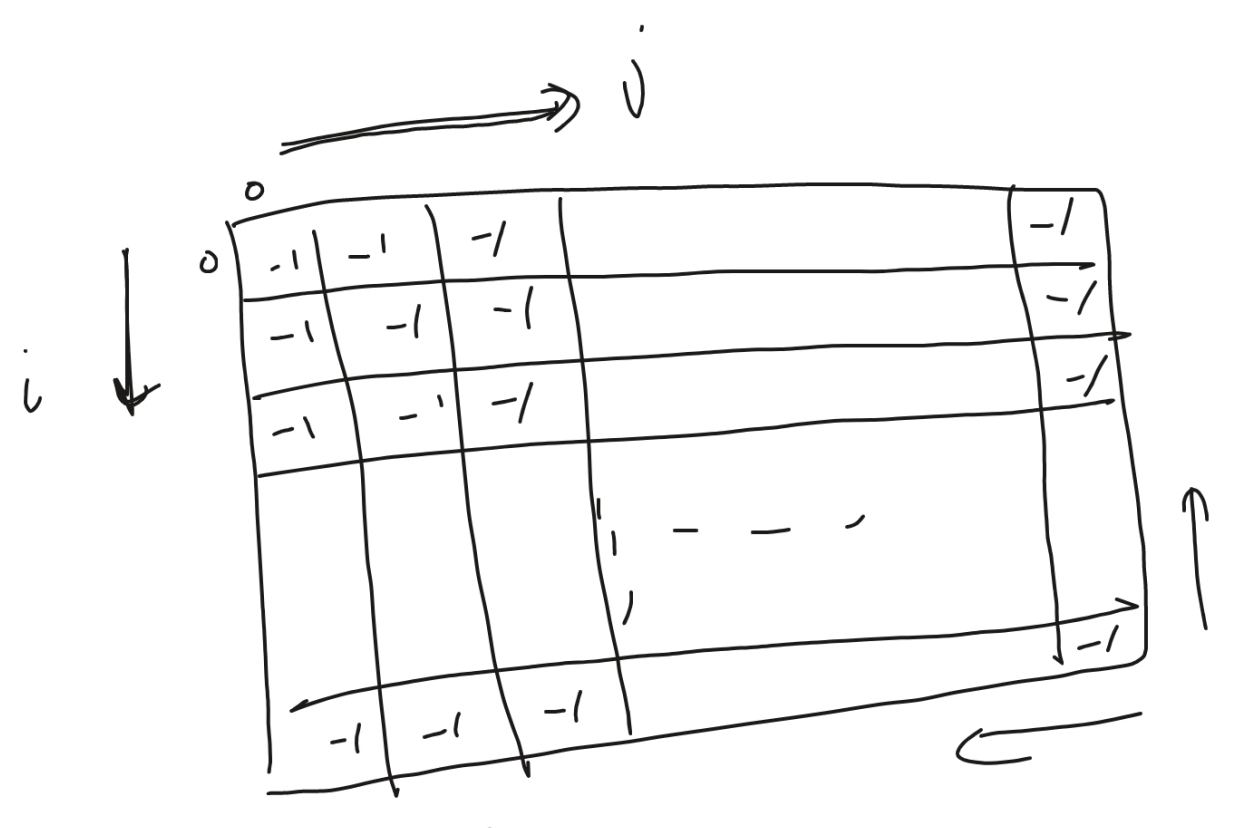

LCS (ABCD, EFGH)
LCS (BCD, EFGH)
LCS (ABCD, FGH)
LCS (CD, EFGH)
LCS (BCD, FGH)
LCS(BCD, FGH)
LCS(ABCD, GH)

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) → 2^{max(n,m)-1}

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



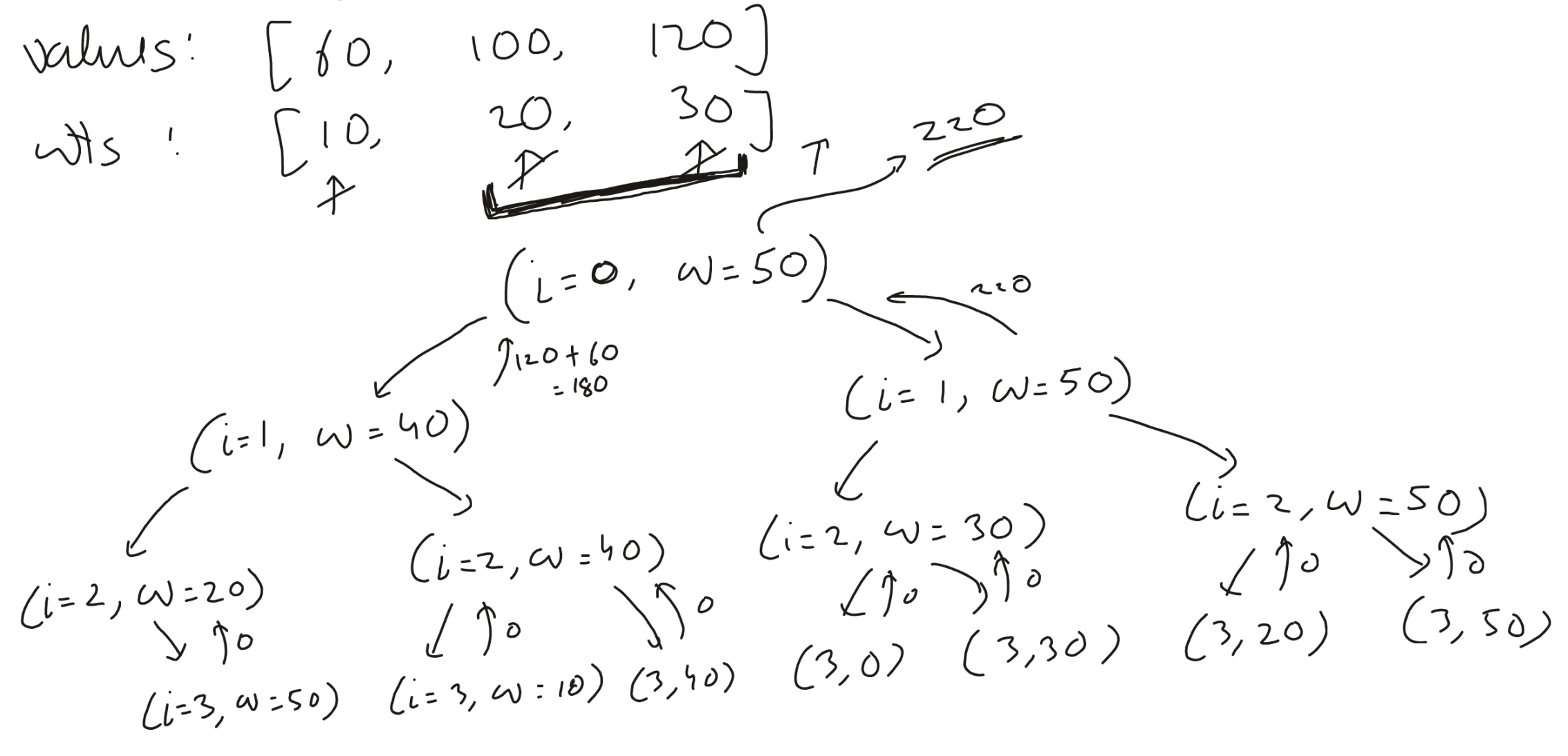
		0	1	2	3	4	5	6
			A	E	D	F	M	R
0		0	0	0	0	0	0	0
1	A	0	1	1	1	1	1	1
2	B	0	1	1	1	1	1	1
3	C	0	1	①	1	1	1	1
4	D	0	1	1	2	2	2	2
5	G	0	1	1	2	②	2	2
6	M	0	1	1	2	2	3	$\boxed{3}$

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

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

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

N = 3, W = 50
values: [60, 100, 120]
wts: [10, 20, 30]
(indices 0, 1, 2)

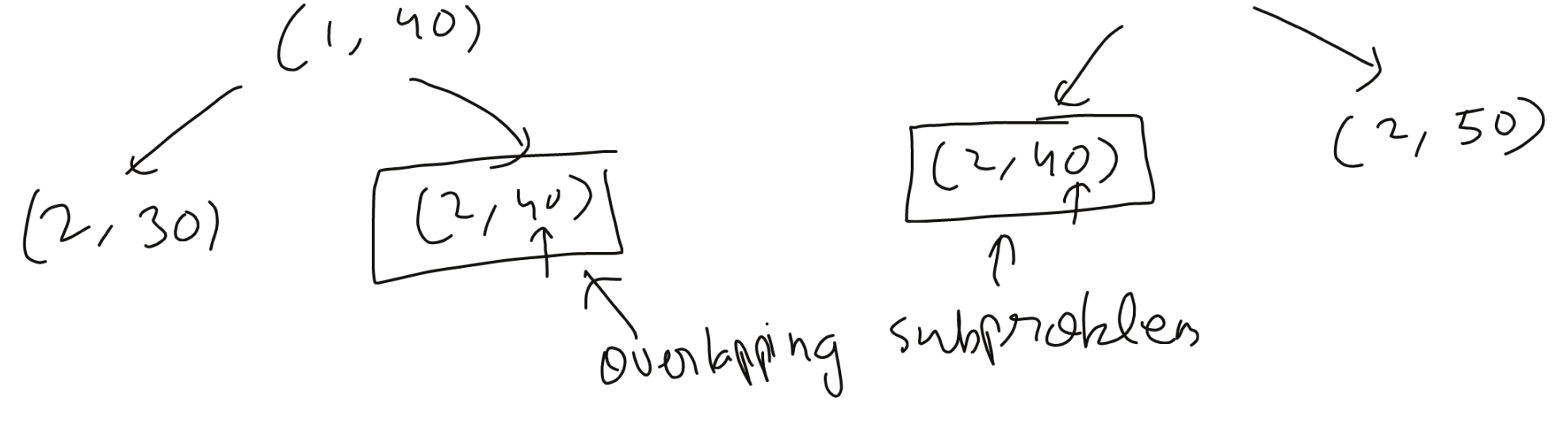

220
(i=0, w=50)
220
120+60 = 180
(i=1, w=40)
(i=1, w=50)
(i=2, w=20)
(i=2, w=40)
(i=2, w=30)
(i=2, w=50)
0
(i=3, w=50)
(i=3, w=10)
(3,40)
(3,0)
(3,30)
(3,20)
(3,50)

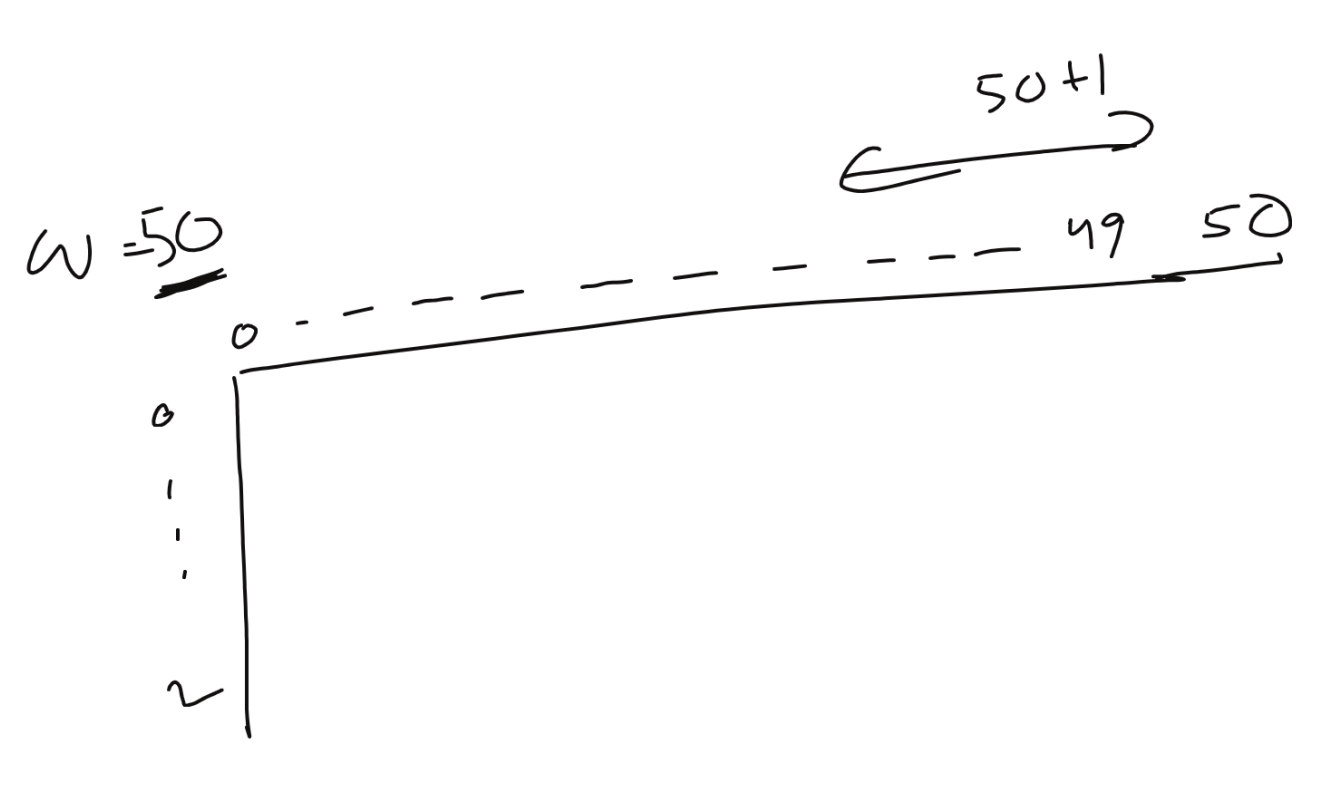
trying to add the current item is the knapsack

max.value
↑
$mv(i, w) = \max(\,[mv(i+1,\ w - \text{wt. of current item}) + \text{value of current item}],$

$\underline{mv(i+1, w)})$
↳ ignoring the current item

vals: [60, 100, 120] W= 50
wts: [10, 10, 20]


(i=0, w = 50)
(1, 40)
(1, 50)
(2, 30)
(2, 40)
(2, 40)
(2, 50)
overlapping subproblem


50+1
W = 50
0
49
50
0
2

Time Complexity of memoized code =

$O\left[(\text{Number of unique recursive states}) * (\text{time needed to evaluate one state})\right]$

$O(n*w) \qquad O(1)$

$= O(n*w)$

wt	val		0	1	2	……	10	11	--20	30	----	50
-	-	0	0	0	0	--	0	0	..	0	--	0
10	60	1	0	0	0	---	60	60	..	60	--	60
20	100	2	0	0	0	...	60	60	100	160	--	160
30	120	3	0	0	0	---	0	0	--	120	..	220

if $wt[i-1] > j$
$\Rightarrow dp[i][j] = dp[i-1][j]$

otherwise:
$dp[i][j] = \max(\underline{dp[i-1][j - wts[i-1]]} + \underline{vals[i-1]},\ \underline{dp[i-1][j]})$

60 100 220

$${}^nC_r = \frac{(n!)}{r!\,(n-r)!}$$

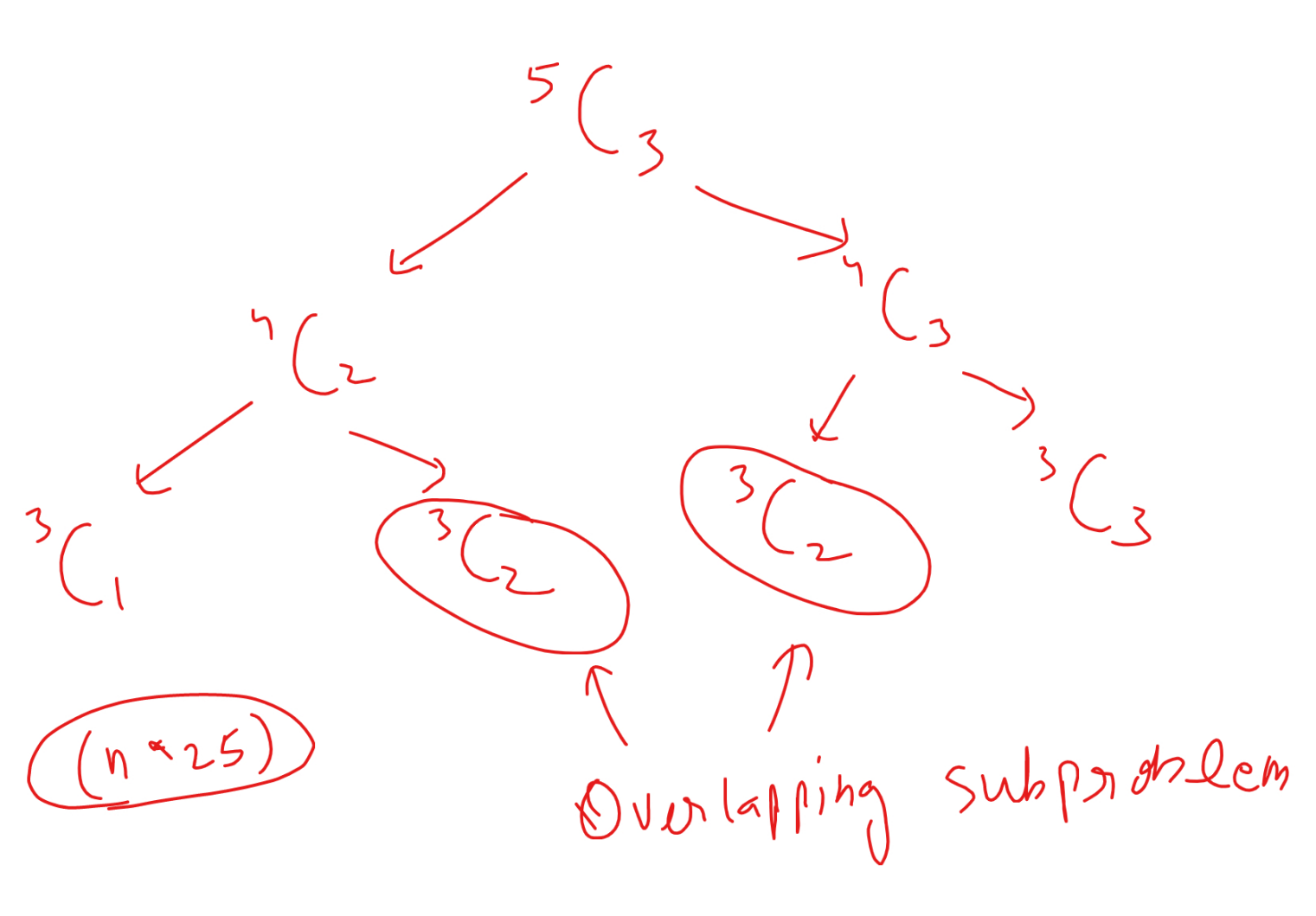
Find ${}^nC_r$ mod P

$(a+b)\,\%\,P = a\,\%\,P + b\,\%\,P$
$(a*b)\,\%\,P = (a\,\%\,P) * (b\,\%\,P)$

BUT $\left[\left(\frac{a}{b}\right)\%\,P \neq \frac{a\,\%\,P}{b\,\%\,P}\right]$

$\left[{}^nC_r = {}^{n-1}C_{r-1} + {}^{n-1}C_r\right]$

$\Rightarrow {}^nC_r\,\%\,P = ({}^{n-1}C_{r-1}\,\%\,P + {}^{n-1}C_r\,\%\,P)\,\%\,P$


5C3
4C2
4C3
3C1
3C2
3C2
3C3
Overlapping subproblem

~~a~~ kbcd
d
25

(n*25)

0 1 2 3
0 → abcd
1 efgh
→ 2 aijkl
→ 3 → abce
4 → ajcd

abce
fbcd

[a d
b b
bc]

1+4 = 5 ×
2+2 = 4
2+3 = 5

a: (0,0) (2,0) (3,0) (4,0)
j: (1,1) (2,1) (4,1)

$\sum i$ * ㉖

$\Rightarrow (23^0 \times 1) + (23^1 \times 2) + (23^2 \times 3) + (23^3 \times 4)$
$\Rightarrow (23^0 \times 1) + (23^1 \times 10) + (23^2 \times 3) + (23^3 \times 4)$

$O(n^2)$

$23^1 \times (8) + 23^2 (4)$
$23(8 + 23^1(4))$

㉖