(i) Burst ballons: Break ballon, add money with maximum possibility.

$a\ b\ c\ e\ f \xrightarrow[(ab)\ \text{cost}]{a} b\ c\ e\ f \xrightarrow[(bc)\ \text{cost}]{b} c\ e\ f \xrightarrow[(cef)\ \text{cost}]{e} c\ f \xrightarrow[(cf)\ \text{cost}]{c} f \xrightarrow[(f)]{f} \cdot$

d → a → b → e → c → f

$d \uparrow$ (cde) cost

$a_0 \quad b_1 \quad c_2 \quad d_3 \quad e_4 \quad f_5$ → cost
ballon index

$b \downarrow$ (abc) cost

a c d e f

$e \downarrow$ (def) cost

$a\ c\ d\ f \xrightarrow[(ac)\ \text{cost}]{a} c\ d\ f \xrightarrow[(df)\ \text{cost}]{f} c\ d \xrightarrow[(cd)]{c} d \xrightarrow[(d)]{d} \cdot$

b → e → a → f → c → d

a → d → c → b → e → f

⋮ All permutations

try all permutation of ballons to get max money.

From all permutation of order of ballons, maximize cost.

b a c → b * a * c

$2_0 \quad 3_1 \quad 5_2$

order of bursting of ballons

2 3 5	2 5 3	3 2 5	3 5 2	5 3 2	5 2 3
2 3 5	2 3 5	2 3 5	2 3 5	2 3 5	2 3 5
2 ↓ 6	2 ↓ 6	3 ↓ 30	3 ↓ 30	5 ↓ 15	5 ↓ 15
3 5	3 5	2 5	2 5	2 3	2 3
3 ↓ 15	5 ↓ 15	2 ↓ 10	5 ↓ 10	3 ↓ 6	2 ↓ 6
5	3	5	2	2	3
5 ↓ 5	3 ↓ 3	5 ↓ 5	2 ↓ 2	2 ↓ 2	3 ↓ 3
.	.	.	.	.	.
Cost 26	24	45	42	23	24

Maximize = 45

order → 3 → 2 → 5

left part cost

Right part cost

cost of bursting of i^{th} index

$0 + [3\,1\,5\,6] + 2$

$[2] + [1\,5\,6] + 3$

$[2\,3] + [5\,6] + 1$

$[2\,3\,1] + [6] + 5$

$[2\,3\,1\,5] + 0 + 6$

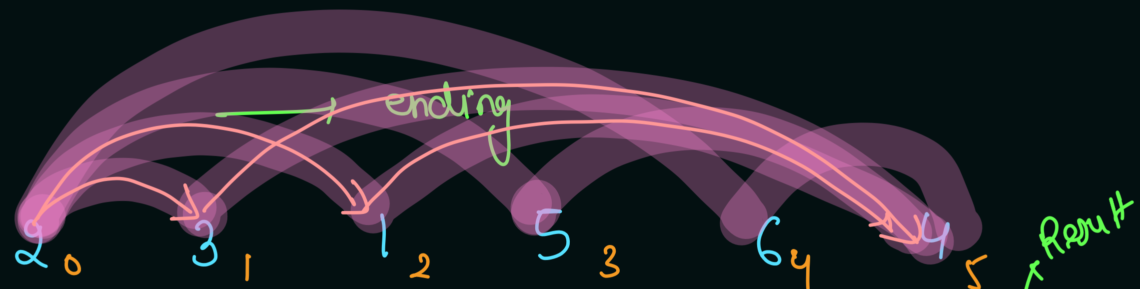
it is numerical '0' because no ballon present in Right

it is numerical '0' becse no ballon present in left

2 3 1 5 6

2 3 1 5 6

cost if last remaining ballon is 'i'



Start ↓

2_0	6 (2)	9 (23)	(231)	(2315)	(23156)	(231564)
3_1	x 0	6 (3)	45 (31) 3 × 1 6 8 1, 3 vs max, 1→4 45	(315)	i=1, j=4, 3156, max	(31564)
1_2	x 0	x 0	15 (1)	105 (15)	(156)	(1564)
5_3	x 0	x 0	x 0	30 (5)	140 (56)	(564)
6_4	x 0	x 0	x 0	x 0	120 (6)	140 (64)
4_5	x 0	x 0	x 0	x 0	x 0	24 (4)

2 [3 1 5 6] 4

→ considering all permut

Cost of bursting of [3 1 5 6]

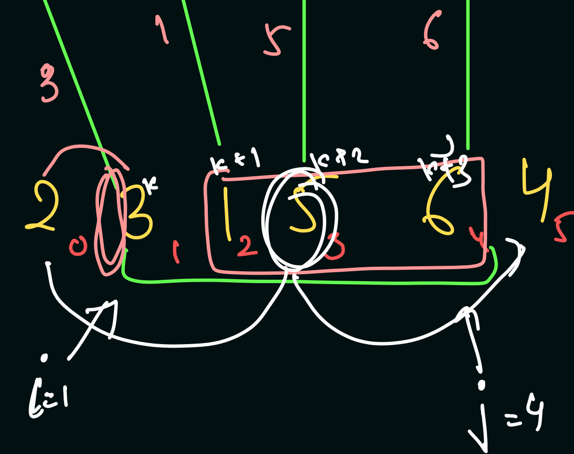
$[3]+[56]+2.1.4$

$[315]+0+2.6.4$

$0+[156]+2.3.4$

$[31]+[6]+2.5.4$

max



$a[i-1] * a[k] * a[k+1]$

$== 1$ if invalid index

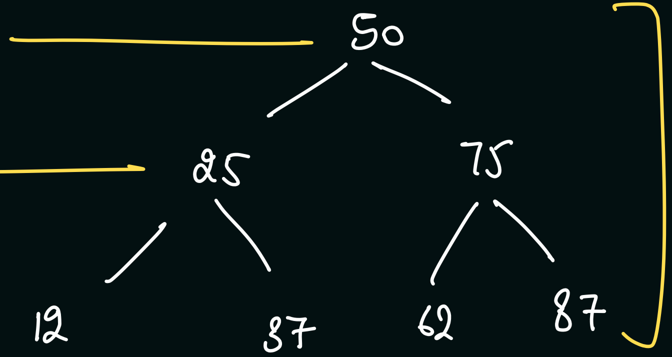
optimal BST :→

cost of searching of ith element = level of ith element * freq.

level 1

level 2

level 3



array →	12	25	37	50	62	75	87	
searching freq →	9	8	6	3	4	3	1	cost

BST

level		Cost
1	12	→ 9*1 = 9
2	25	→ 8*2 = 16
3	37	→ 6*3 = 18
4	50	→ 3*4 = 12
5	62	→ 4*5 = 20
6	75	3*6 = 18
7	87	7*1 = 7

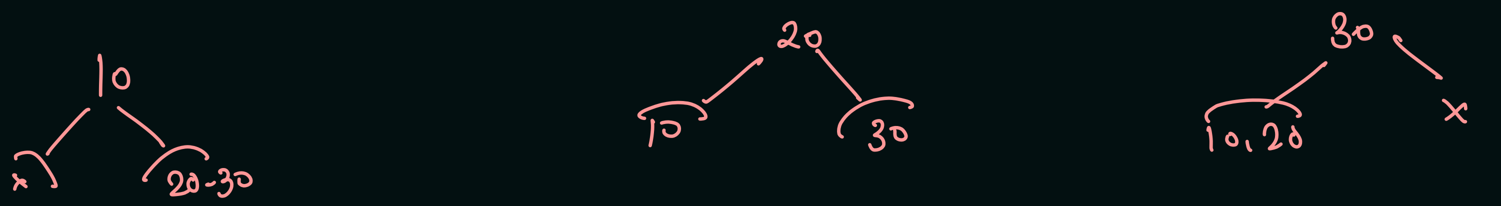
= 100

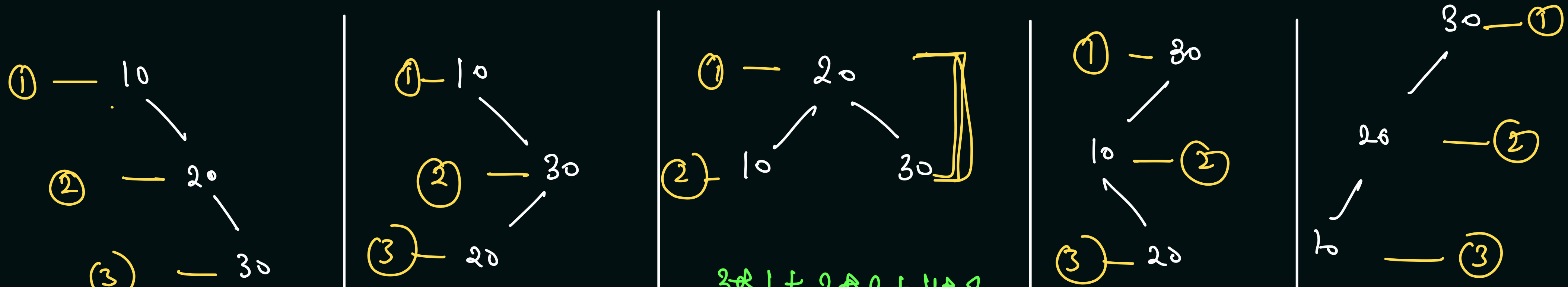
Cost ⇒ 3*9 + 2*8 + 3*6 + 1*3 + 3*4 + 2*3 + 3*1

27 + 16 + 18 + 3 + 12 + 6 + 3 = 85

Element →	10	20	30
freq →	2	3	4

Total possible BST → $C_3 = 5$

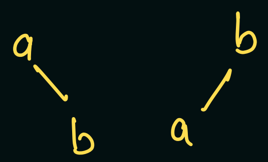
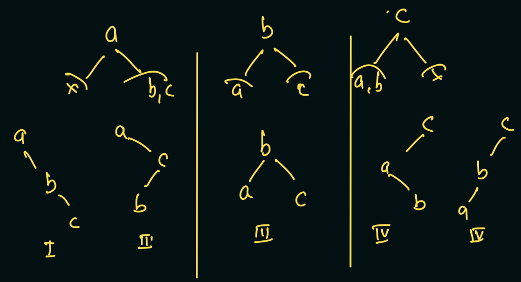




$1*2 + 2*3 + 4*3$	$1*2 + 4*2 + 3*3$	$3*1 + 2*2 + 4*2$	$4*1 + 2*2 + 3*3$	$4*1 + 3*2 + 3*2$
$2 + 6 + 12$	$2 + 8 + 9$	$3 + 4 + 8$	$4 + 4 + 9$	$4 + 6 + 6$
$= 20$	$= 19$	$= 15$	$= 17$	$= 16$

Optimal BST

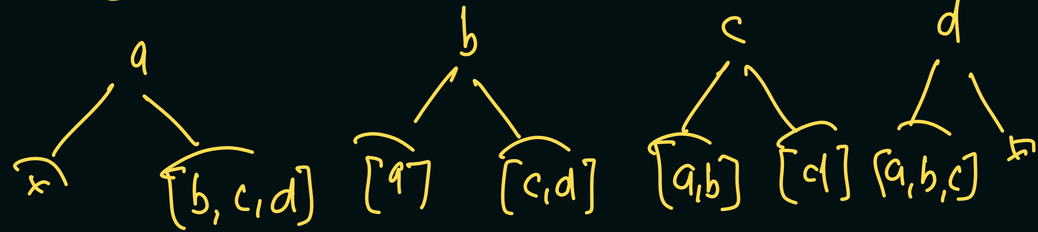
Min Cost for Searching

$\underline{\underline{n}}$	$\underline{\underline{C_n}}$ (catalan)
a $\longrightarrow$ 1	a
[a, b] $\longrightarrow$ 2	
[a, b, c] $\longrightarrow$ 5	
[a, b, c, d] $\longrightarrow$ 14	

Element → a b c d (sorted)

freq → a' b' c' d'

1 * 5 + 1 * 2 + 2 * 1 + 5 * 1



prefix sum arr: a' → $a'+b'$ → End $a'+b'+c'$ → $a'+b'+c'+d'$

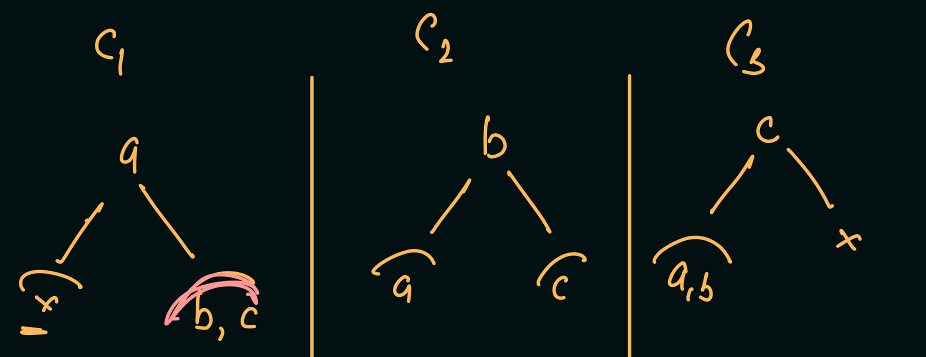
Element: a b c d

freq → a' b' c' d'

a_0 b_1 c_2 d_3

Start ↓

	a_0	b_1	c_2	d_3
a_0	a: a'	ab: min[$a'+2b'$ OR $b'+2a'$]	abc: Min(C_1, C_2, C_3)	abcd
b_1	x	b: b'	bc: min[$b'+2c'$ OR $c'+2b'$]	bcd: $b'+c'+d'$
c_2	x	x	c: c'	cd: min[$c'+2d'$ OR $d'+2c'$]
d_3	x	x	x	d: d'



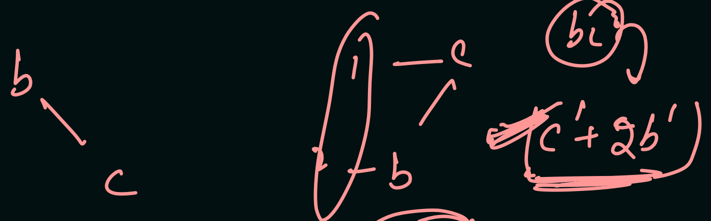
$(x)+[bc]+\text{cost}(a)$ | $[a]+[b]+\text{cost}(b)$ | $[a,b]+0+\text{cost}(C)$

C_1: $0+(bc)+a'+b'+c'$, with $bc = c'+b'$

C_2: $(a)+(b)+a'+b'+c'$

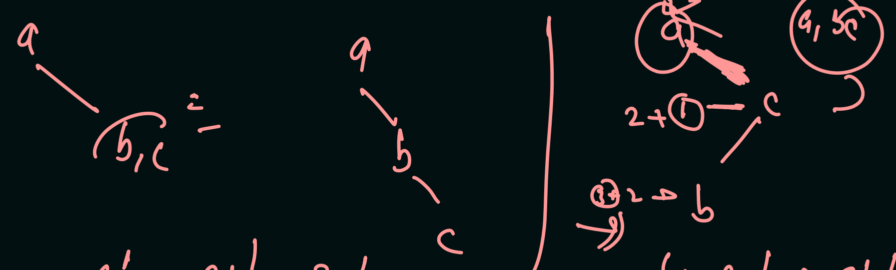
C_3: $(ab)+0+a'+b'+c'$

b, c



$b'+2c'$ OR $c'+2b'$

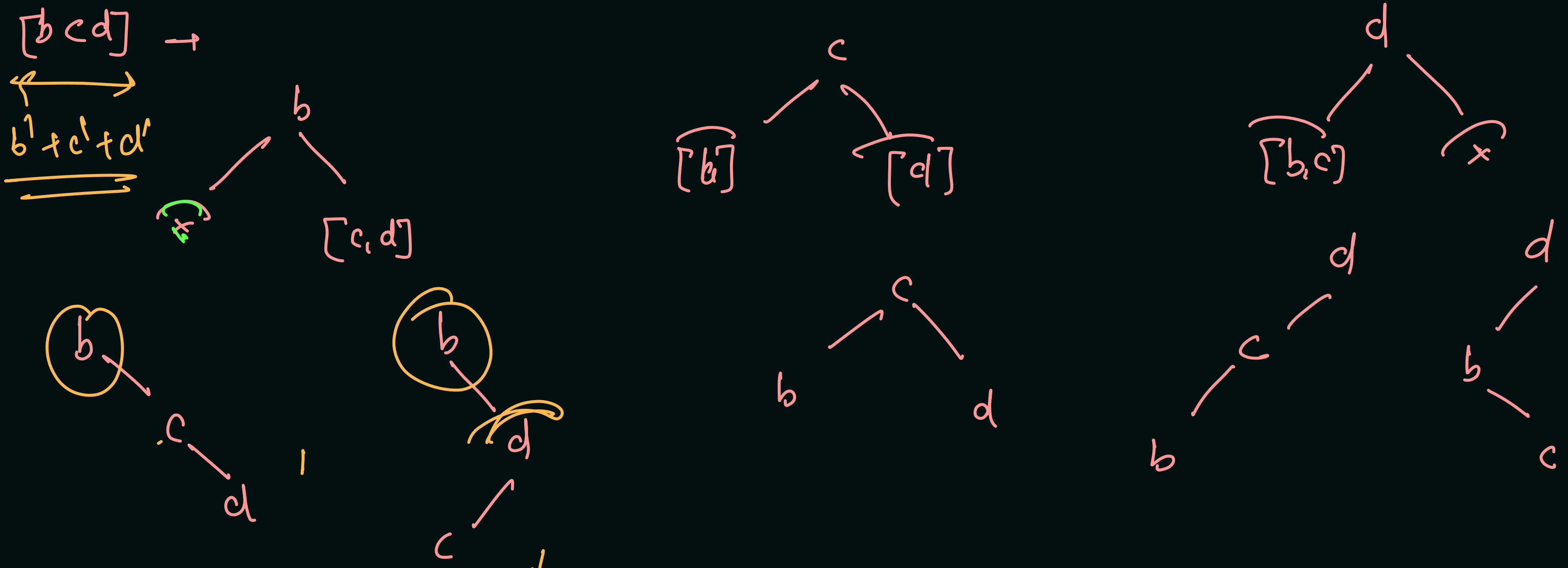
bc: $(c'+2b')$



$a'+2b'+3c'$

$a'+2c'+3b'$

$(a'+b'+c') + (c'+2b')$ bc



$b' + c' + d' + c' + 2d'$ $b' + c' + d' + d' + 2c'$

$0 + cd + b' + c' + d'$ $b + d + b' + c' + d'$ $bc + 0 + b' + c' + d'$

Egg drops

E - Eggs , F - floors, Minimum number of attemp to find critical floor with 'e' eggs?

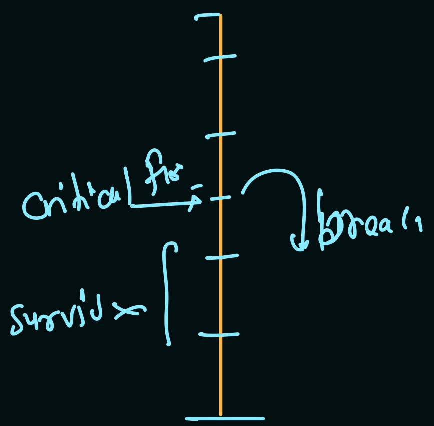
critical floor. [NOTE: critical floor may be ~~exist~~ or not in all floors]

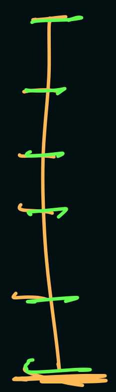
floor $f >= c$] breaks.

floor $< c$] survive

C → critical floor.

Egg = 1 , Floor = k





Problem is not about to find critical floor, but it is about to find min. number of attempts such that we can find critical floor certainly. (Guaranteed)

→ optimise the bad luck

→ Best of the worst

CEO → Manager → team leads
↓
employee

P - Project

Guaranteed best time

$T_{11} = \max(t_1, t_1')$

$T_{22} = \boxed{\max}(t_2, t_2')$

$T_{33} = \boxed{\max}(t_3, t_3')$

$T_{44} = \boxed{\max}(t_4, t_4')$

$t_1 \quad t_1'$ — x_1

$t_2 \quad t_2'$ — x_2

$t_3 \quad t_3'$ — x_3

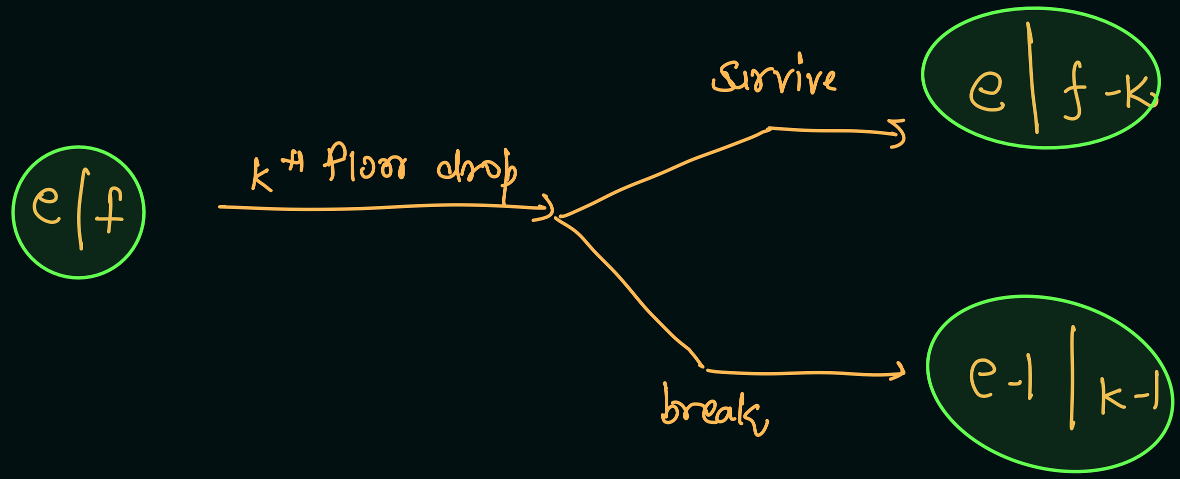
$t_4 \quad t_4'$ — x_4

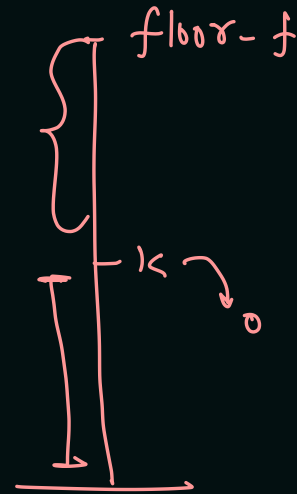
bad luck

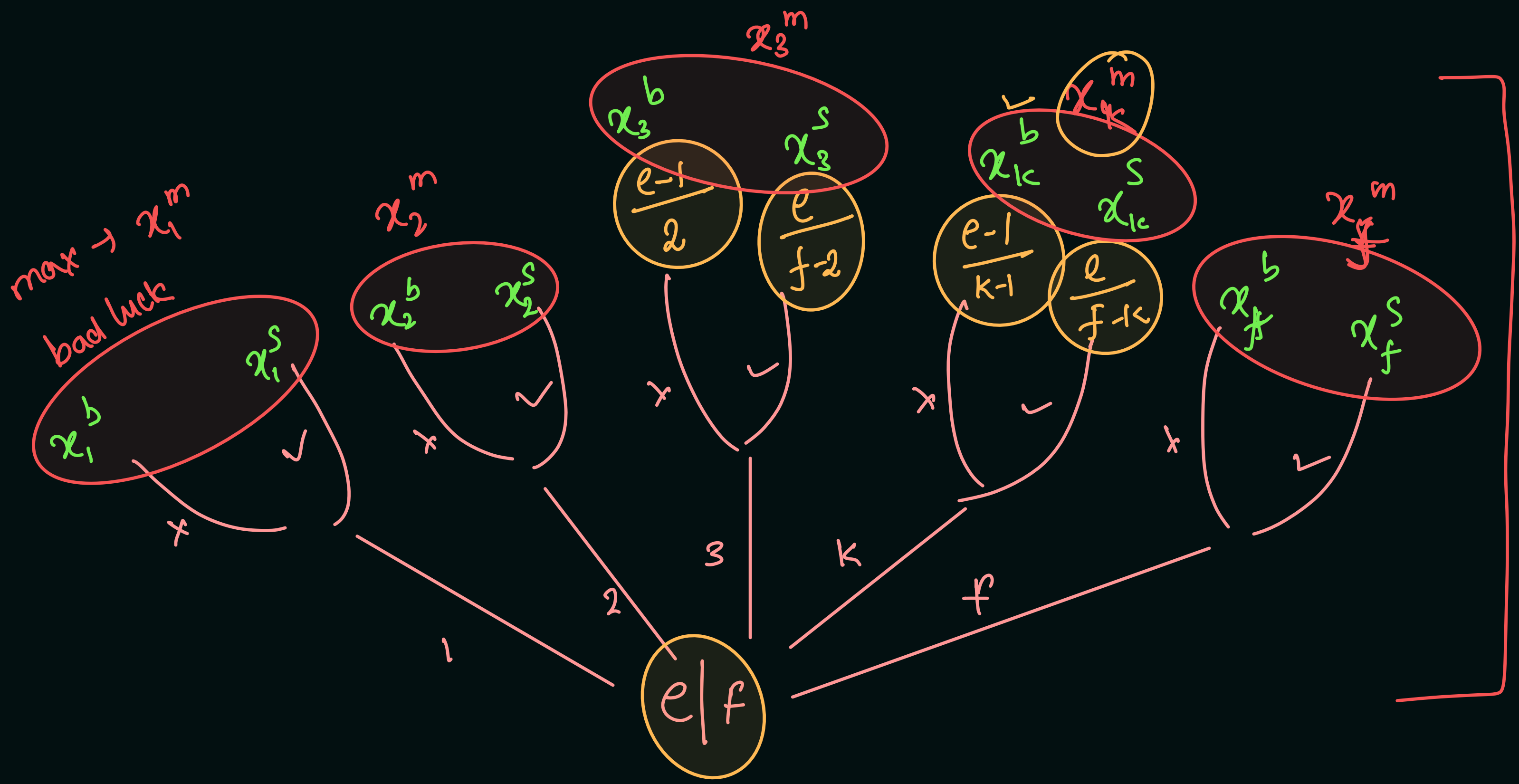
Manager

choice → best ⇒ $\min(T_{11}, T_{22}, T_{33}, T_{44})$

Egg drops:







$$\mathrm{Min}\left(x_1^m, x_2^m, x_3^m, x_k^m, x_f^m\right) + 1$$

Floor ⇒ 7

Eggs ⇒ 3

Base case →

$f=0 \longrightarrow 0$

$e=0 \longrightarrow x$

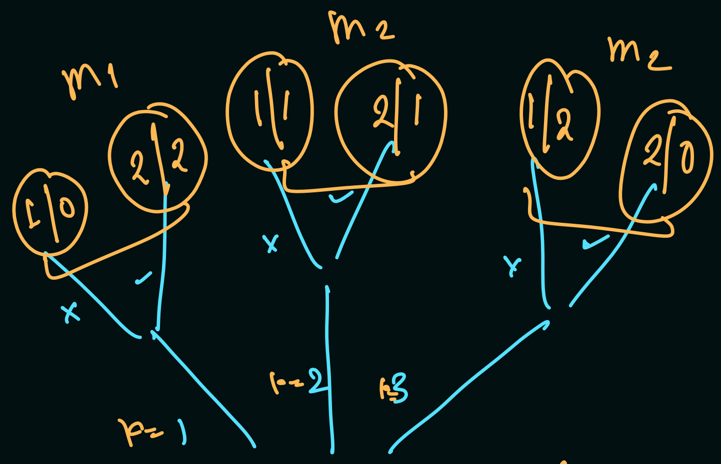
$e=1 \longrightarrow f$

$f=1 \longrightarrow 1$

floor ⟶

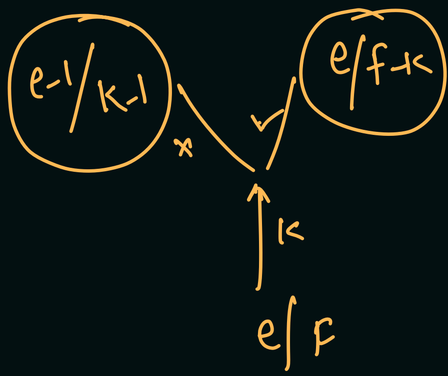
eggs ↓

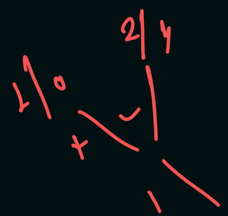
	0	1	2	3	4	5	6	7
0	x	x	x	x	x	x	x	x
1	0 a →	1 b →	2 c →	3 d →	4 e	5	6	7
2	0 e' ←	1 d' ←	2 c' ←	2 b' ←	3 a'	3	3	4
3	0	1	2	2	3	3	3	3



$e=2 \mid f=3$

$\min(m_1, m_2, m_3) + 1$





$$\begin{bmatrix} \text{Max}(a, a') \\ \text{"} \ (b, b') \\ \text{"} \ (c, c') \\ \text{"} \ (d, d') \\ \text{"} \ (e, e') \end{bmatrix} \min + 1$$