Optimal Binary Search Tree

arr : [3, 5, 7]

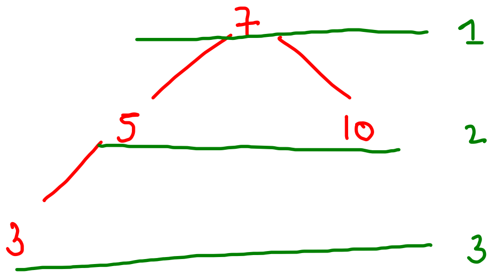
freq : [2, 3, 1]

3, 5, 7

		8	13
3	:	1 × 2	1 × 2
5	:	2 × 3	3 × 3
7	:	3 × 1	2 × 1

arr : [3, 5, 7, 10]

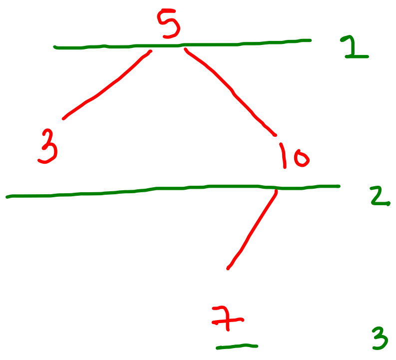
freq : [2, 3, 1, 8]



(7) (5) (10) (3)

cost = $1*1 + 2*3 + 2*8 + 3*2$

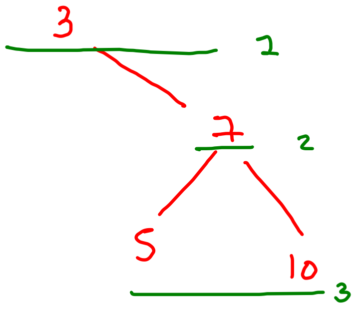
= 28



(5) (3) (10)

cost = $1*3 + 2*2 + 2*8 + 3*1$

= 26



(3) (7) (5) (10)

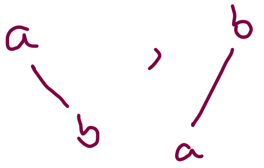
cost = $1*2 + 2*1 + 3*3 + 3*8$

= 37

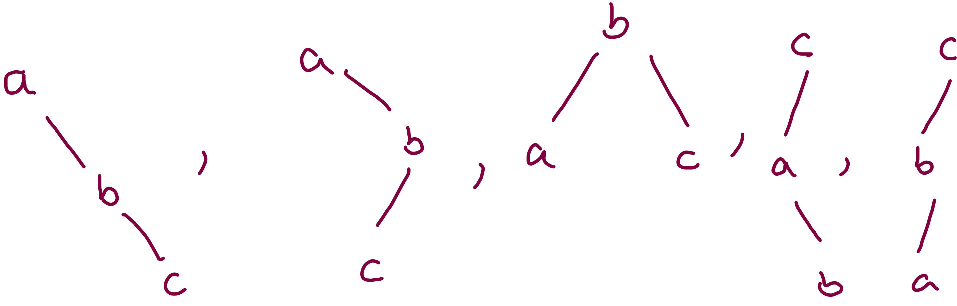
1 1 a

a, b, c, d

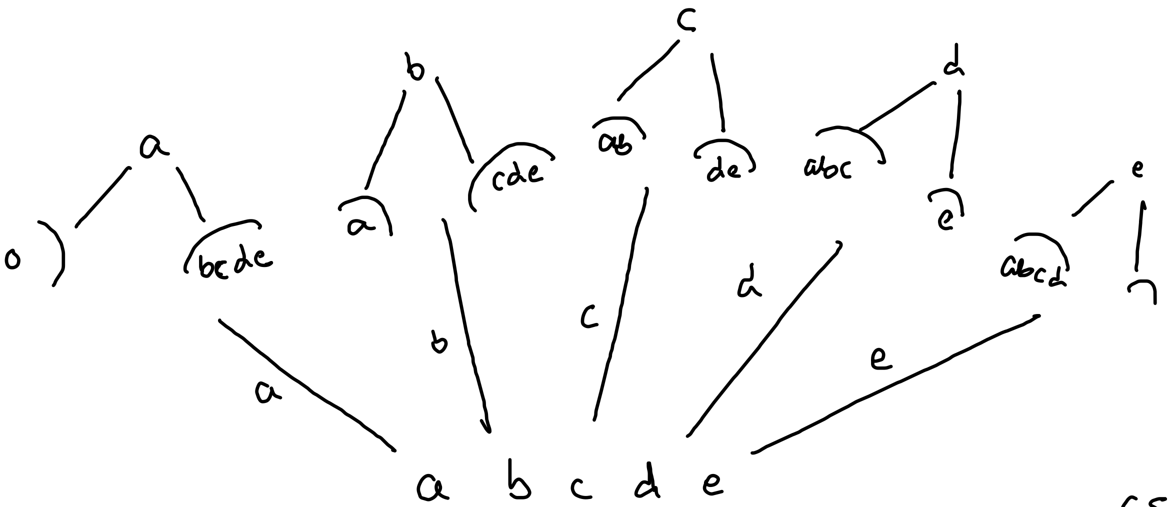
2 2



3 5



$$C_3 = \underbrace{C_0 \cdot C_2}_{2} + \underbrace{C_1 \cdot C_1}_{1} + \underbrace{C_2 \cdot C_0}_{2}$$

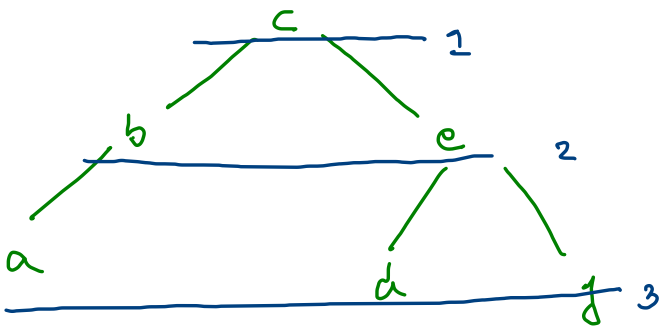


$a_1 = [a, b, c, d, e]$

$a_2 = [a', b', c', d', e')$

a' -> freq of searching
a

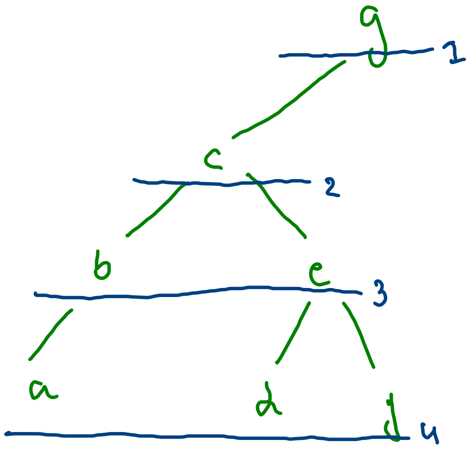
$C_5 = C_0C_4 + C_1C_3 + C_2C_2 + C_3C_1 + C_4C_0$



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

$+ 3d' + 3f'$

a, b, c, d, e, f



$c' + 2b' + 2e' + 3a' + 3d' + 3f' +$

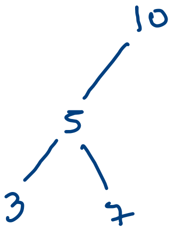
$g' + c' + b' + e' + a' + d' + f'$

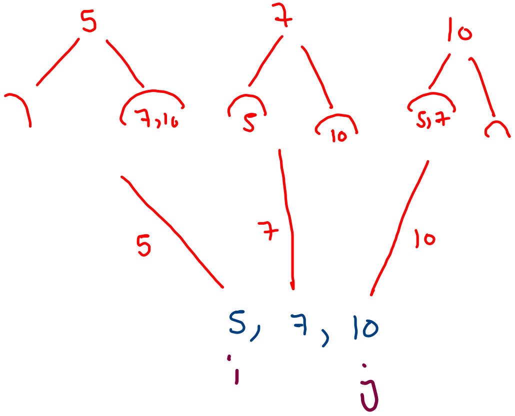
$g' + 2c' + 3b' + 3e' + 4a' + 4d' + 4f'$

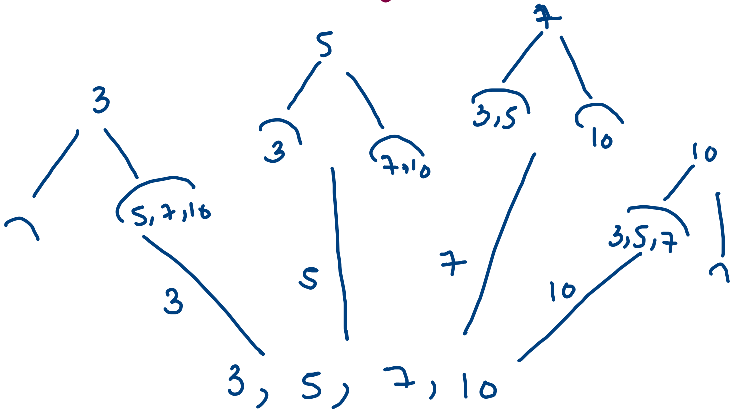
a, b, c, d, e, f, g

arr : [3, 5, 7, 10]

freq : [2, 3, 1, 8]







	3 - 2	5 - 3	7 - 1	10 - 8
3 - 2	2	7	9	23
5 - 3	α	3	5	17
7 - 1	α	α	1	10
10 - 8	α	α	α	8

312. Burst Balloons

```
Input: nums = [3,1,5,8]
Output: 167
Explanation:
nums = [3,1,5,8] --> [3,5,8] --> [3,8] --> [8] --> []
coins =  3*1*5    +   3*5*8   +  1*3*8  + 1*8*1 = 167
```

(i) bo: 5 8 3 1

coins: 52

(i)

3, 1, ~~5~~, 8

5 ↓ 40

3, 1, ~~8~~

8 ↓ 8

~~3~~, 1

3 ↓ 3

~~1~~

1 ↓ 1

(ii)

3, ~~1~~, 5, 8

1 ↓ 15

3, ~~5~~, 8

5 ↓ 120

3, ~~8~~

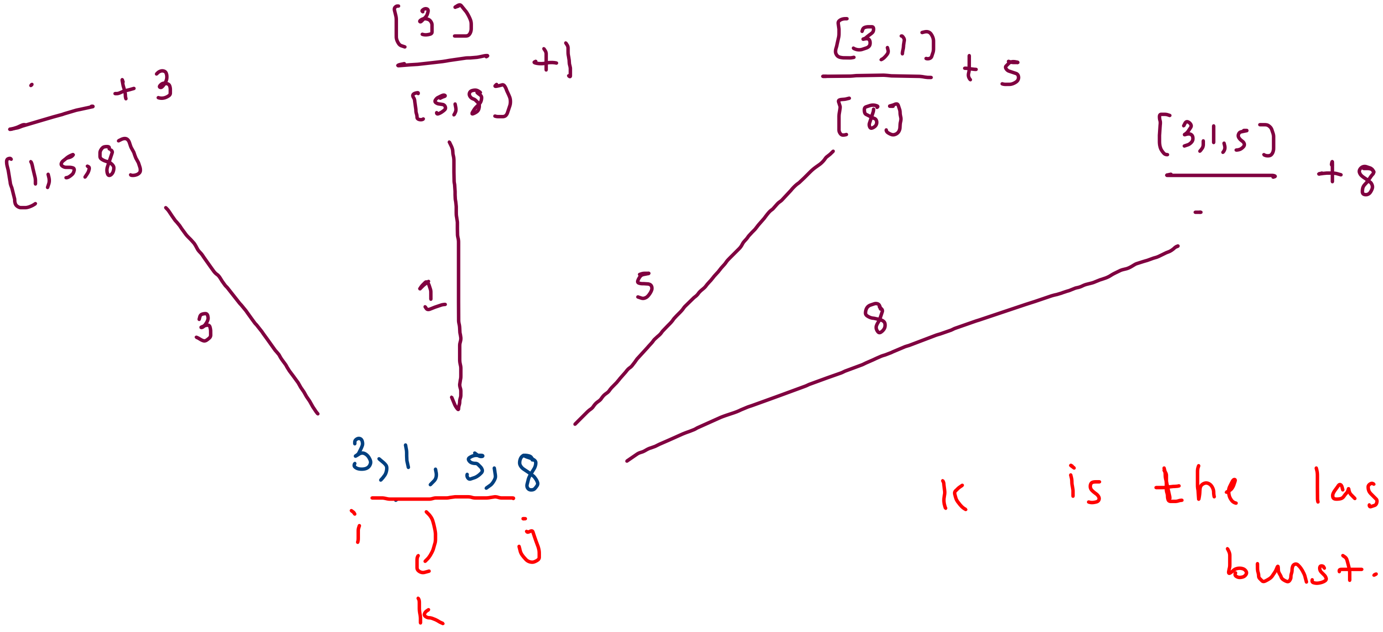
8 ↓ 24

~~3~~

3 ↓ 3

(ii) bo: 1 5 8 3

coins: 162



k is the last ballon to be burst.

arr : [3, 1, 5, 8]

	3	1	5	8
3	3	30		
1	α	15	135	
5	α	α	40	48
8	α	α	α	40

3 1 5 8

3: $\frac{[\cdot]}{[1]}$ + 15 → 0+15+15

1: $\frac{[3]}{[\cdot]}$ + 5 → 3+0+5

3 15 8

1: $\frac{[\cdot]}{[5]}$ + 1 × 3 × 8   0 + 40 + 24

5: $\frac{[1]}{[\cdot]}$ + 5 × 3 × 8   15 + 0 + 120

eg: 11 2 3 1 ~~5 8~~ 6 10

i = 3, k = 1, j = 8

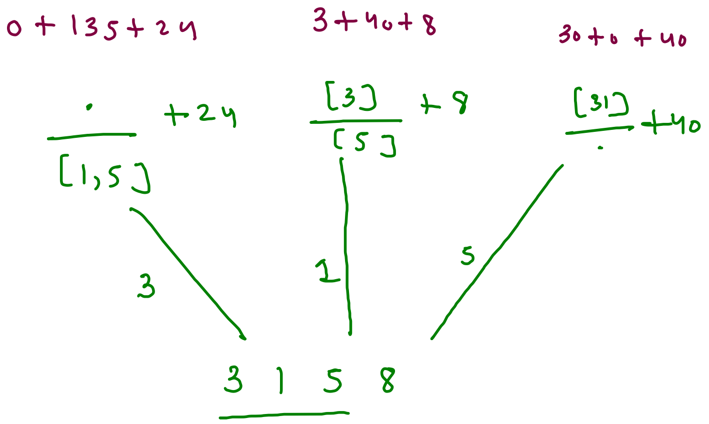
sc = v[k] × v[i-1] × v[j+1]

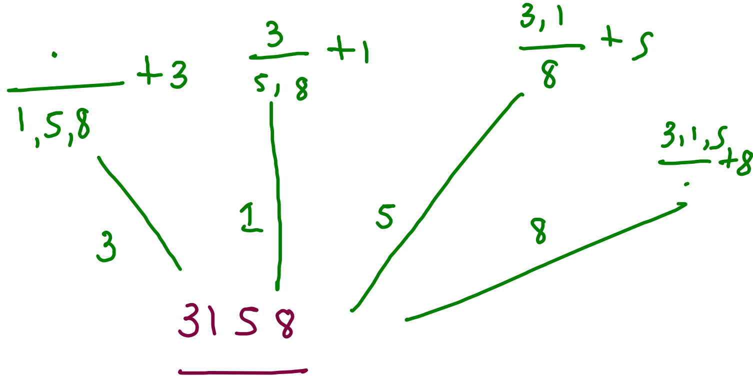
la = i, k-1

ra = k+1, j

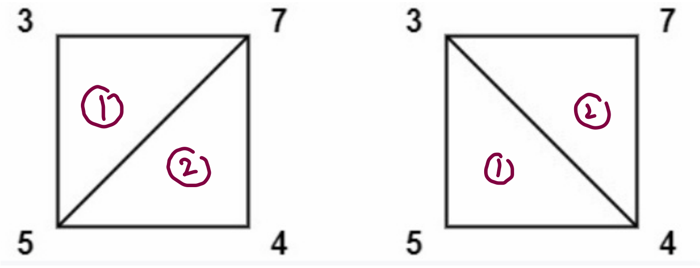
	3	1	5	8
3	3	30	159	167
1	α	15	135	143
5	α	α	40	48
8	α	α	α	40

1 5 3 8



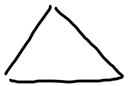
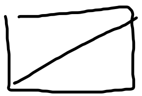
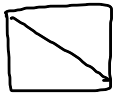
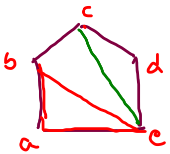
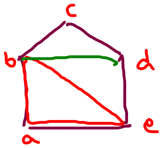
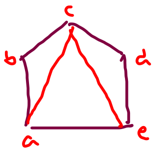
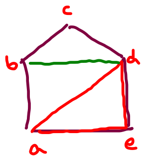
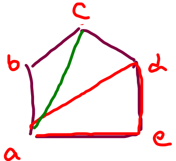


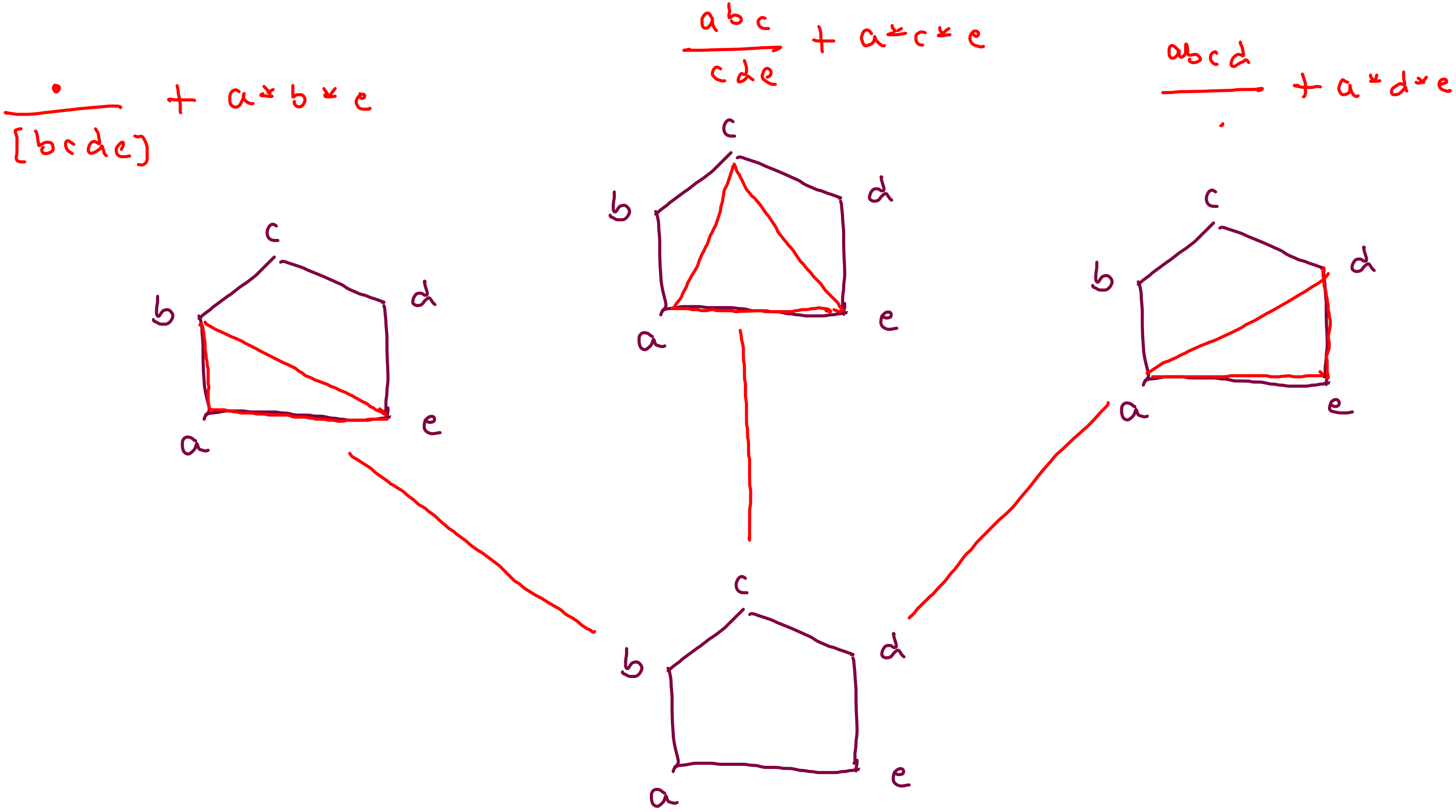
1039. Minimum Score Triangulation of Polygon



$$5 * 3 * 7 + 5 * 4 * 7 = 245$$

$$4 * 5 * 3 + 4 * 7 * 3 = 144$$

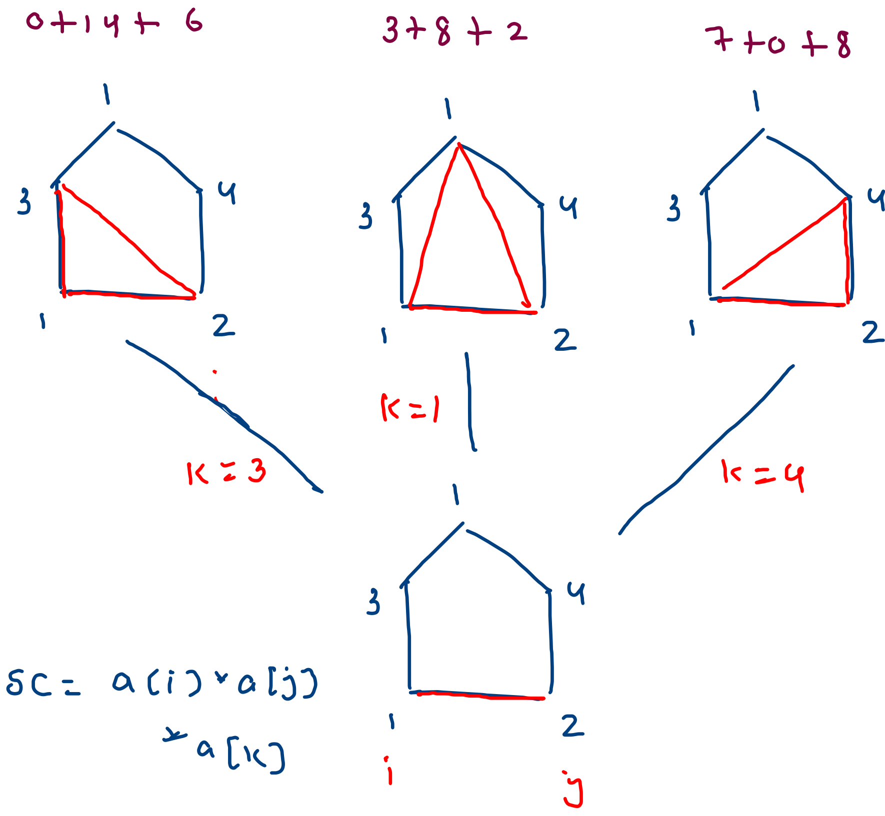
v	△ ways	Catalan	diagram
1	X	X	
2	X	X	
3	1	C_1	
4	2	C_2	 
5	5	C_3	    



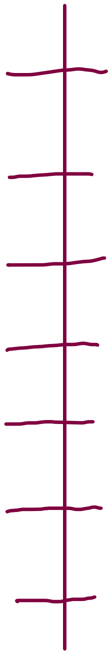
abc
cde
+ a*c*e
•
[bcde]
+ a*b*e
abcd
+ a*d*e
c
b
d
a
e

[1,3,1,4,2,5]

	1	3	1	4	2	5
1	0	0	3	7	13	
3	α	0	0	12	14	
1	α	α	0	0	8	18
4	α	α	α	0	0	40
2	α	α	α	α	0	0
5	α	α	α	α	α	0



Egg Drop



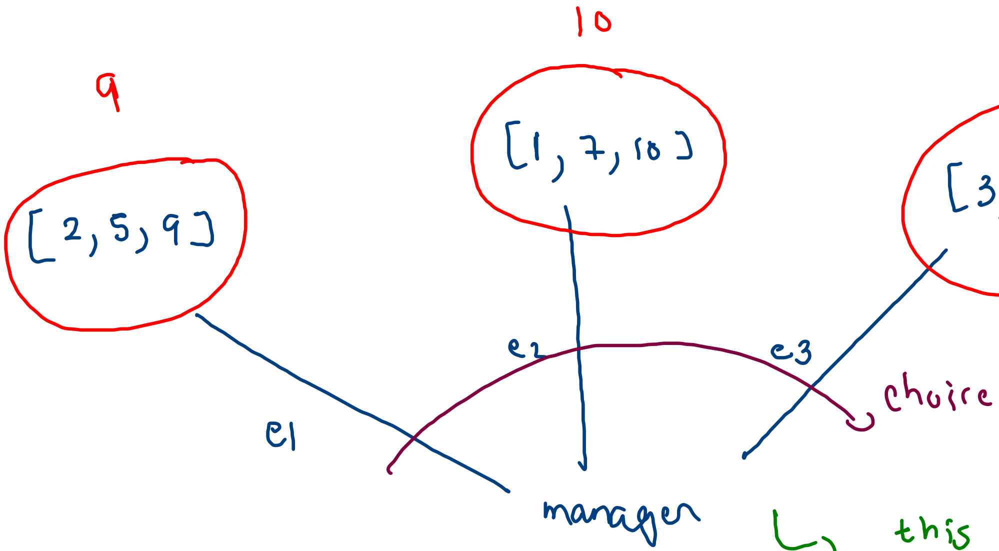
critical floor : the lowest floor from which egg will break.

e, f

e : eggs

f : floors

min no. of attempts such that we can the critical floor no matter where it lies.



↳ this job can be done in 6 hours for sure.

guaranteed time

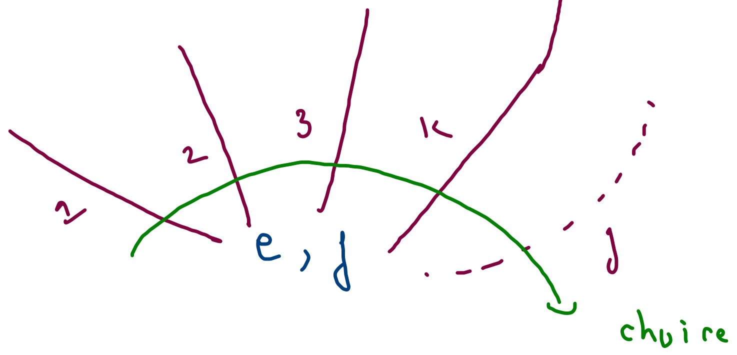
(i) things to happen : worst

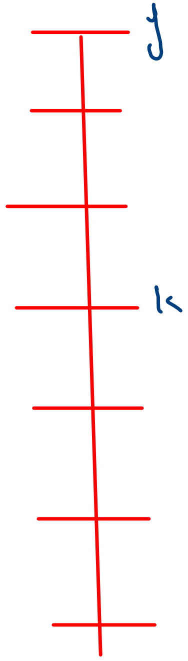
(ii) choice, you do things (luck 0%) : best egg drop

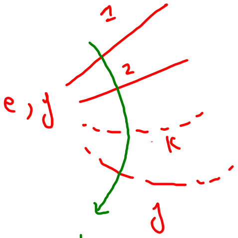
$$\max(a_{kb}\ ,\ a_{ks})$$

$$(e-1, k-1)\ |\ (e, j-k)$$

egg break | egg saved







choice (min)

egg break : $e-1,\ k-1$

egg survive : $e,\ j-k$

max

$e = 1$, ans : j

$j = 1$, ans : 1

base cases

$e = 3$

$j = 7$

→ floors

eggs ↓

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	1	2	2	3	3		
3	0	1						

1,0 eb · 2,1 es · 1,1 eb · 2,0 es

2, 2
e j

2: (1,0) eb, (2,2) es

1: (1,1) eb, (2,1) es

2: (1,2) eb, (2,0) es

2, 3
e j

3: (1,0) eb, (2,4) es

2: (1,1) eb, (2,3) es

2: (1,2) eb, (2,2) es

3: (1,3) eb, (2,1) es

4: (1,4) eb, (2,0) es

2, 5

```
for(int k = 1; k <= j;k++) {
    int eb = dp[i-1][k-1];
    int es = dp[i][j-k];
    int max = Math.max(eb,es); //out of control

    min = Math.min(min,max);
}
```

Optimal Strategy For A Game

20	30	2	2	2	10
0	1	2	3	4	5

(i) arr ; even length

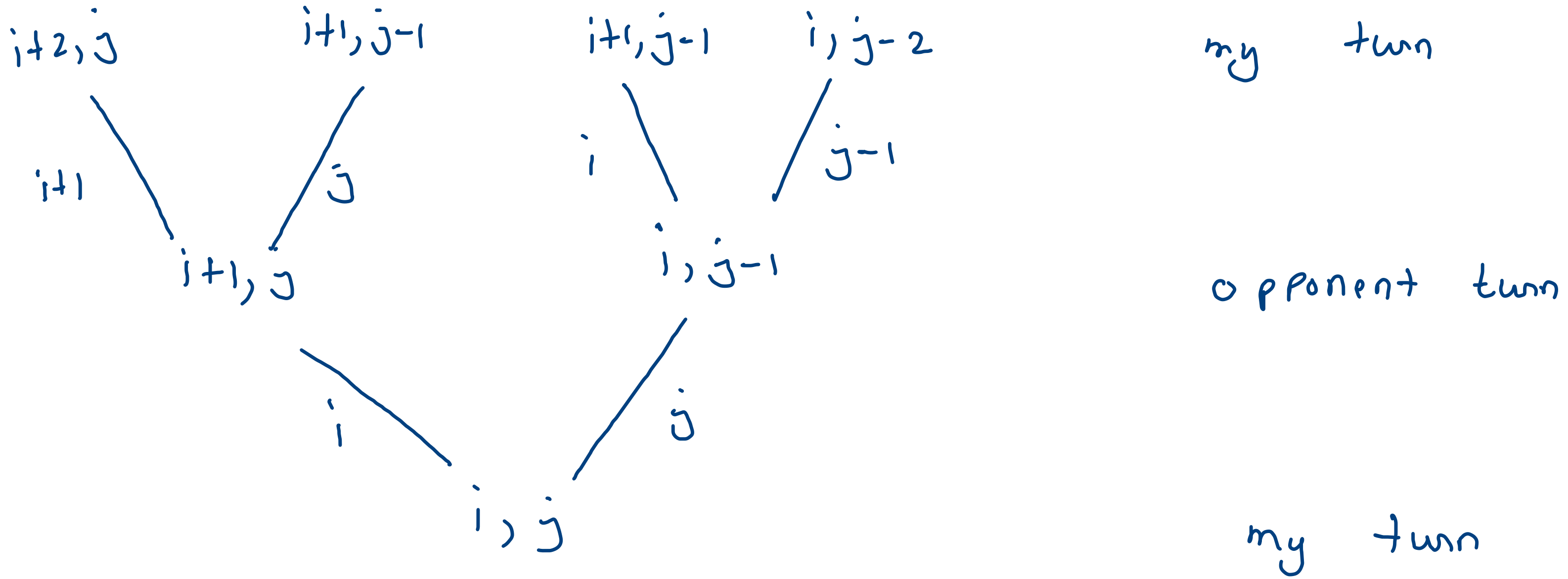
(ii) alternatively play

(iii) you take first turn

(iv) max collected coin on winning.

If only win is considered $\left[\begin{array}{l}\text{odd sum} > \\ \text{even sum}\end{array}\right.$

(select odd always)



$$dp[i][j] = \max \begin{bmatrix} val[i] + \min \begin{pmatrix} dp[i+2][j] \\ dp[i+1][j-1] \end{pmatrix} \\ \\ val[j] + \min \begin{pmatrix} dp[i+1][j-1] \\ dp[i][j-2] \end{pmatrix} \end{bmatrix}$$

$$dp[i][j] = \max \begin{bmatrix} val[i] + \min \begin{pmatrix} dp[i+2][j] \\ dp[i+1][j-1] \end{pmatrix} \\ \\ val[j] + \min \begin{pmatrix} dp[i+1][j-1] \\ dp[i][j-2] \end{pmatrix} \end{bmatrix}$$

	5	3	7	10
5	5	5	10	15
3	α	3	7	13
7	α	α	7	10
10	α	α	α	10

— : i

$5 + \min(7, 10)$

$= 12$

— : j

$10 + \min(7, 5)$

$= 15$