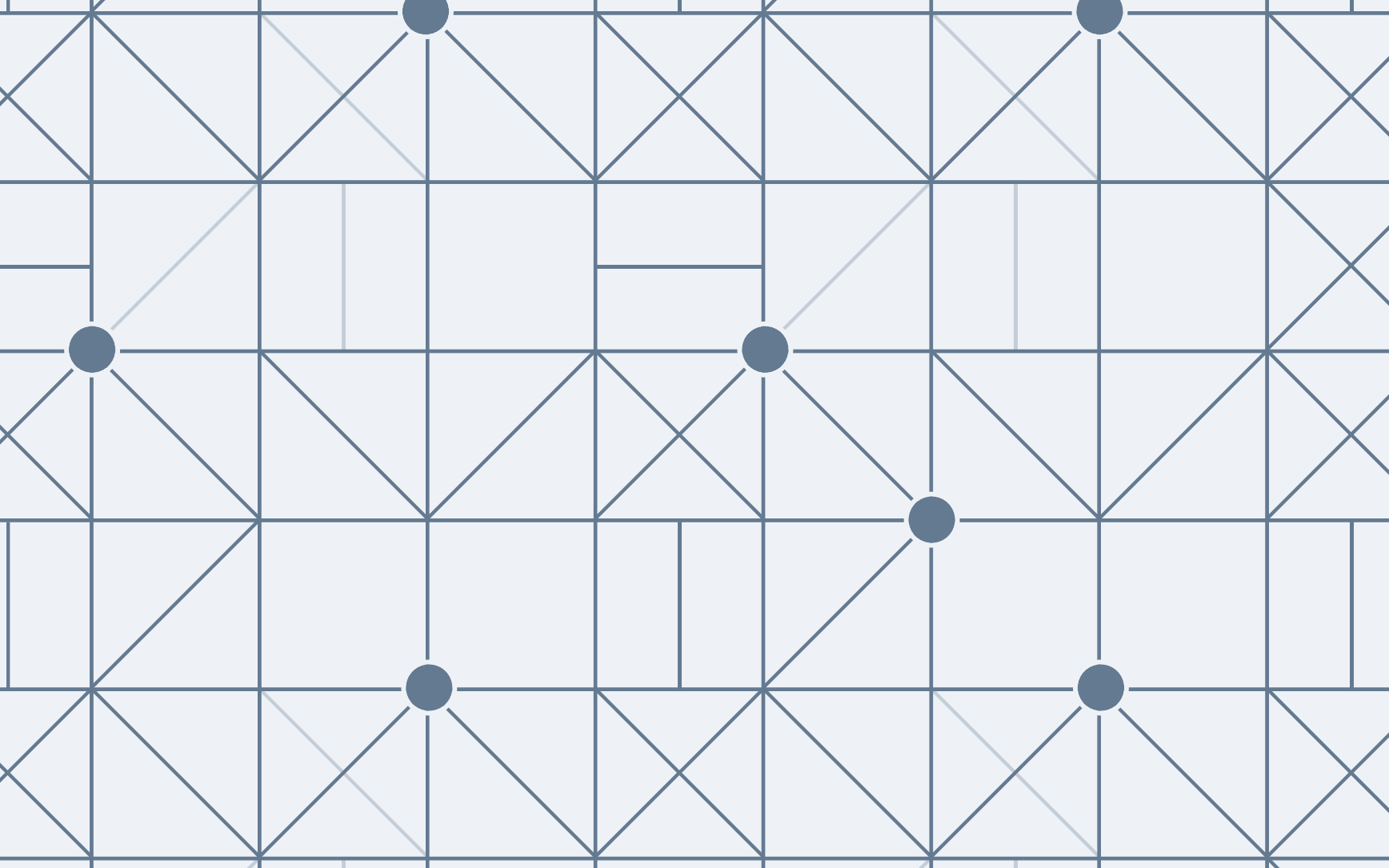


Aggressive cows.

stalls [] = { 0 1 2 3 4 5 : 0, 3, 7, 9, 10, 4 } { cows = 4 } → aggressive to each other

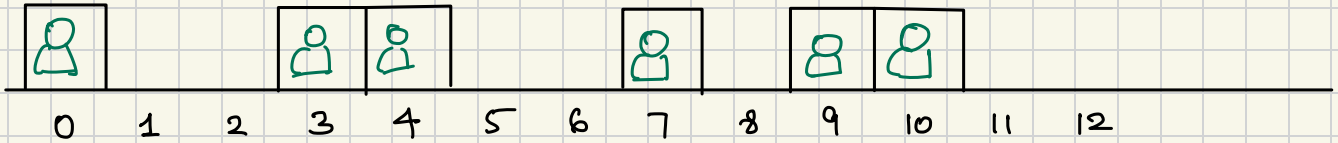
maximize minimum dist. b/w any two aggressive cows!

way1 min 1

way2 min 2

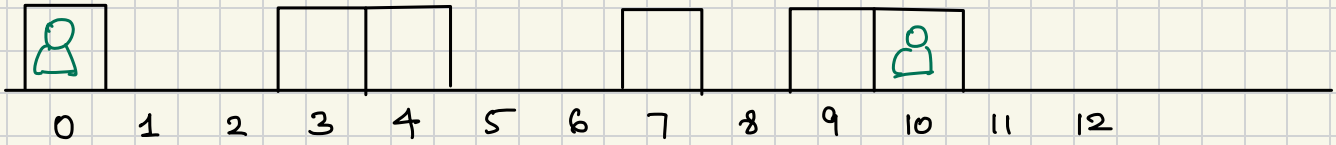
{ 6C4 ways }

Case 1 cows = N {No. of stalls}



min dist = 1 → max. = 1 ✓

Case 2 cows = 2

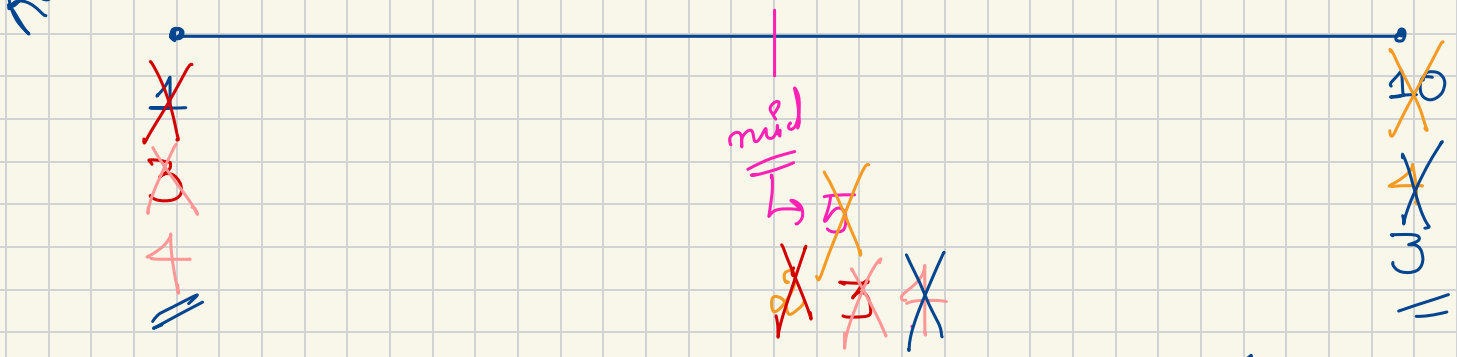


min dist = 10 ✓

Range of search!

$2 \le cows \le N$

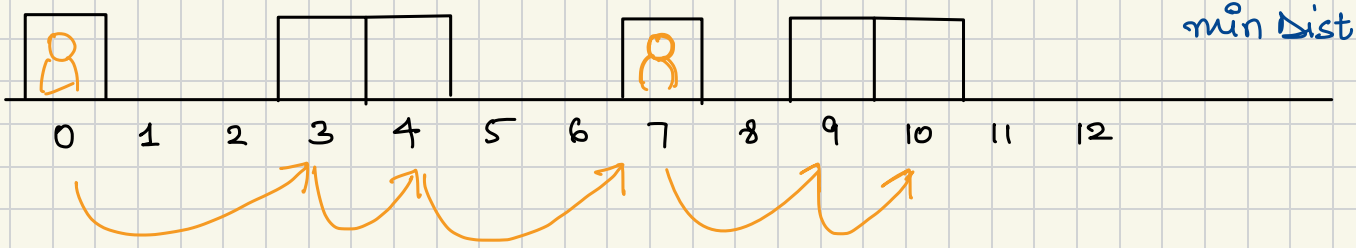
cows = 4



ans = ~~1~~ ~~2~~ 3

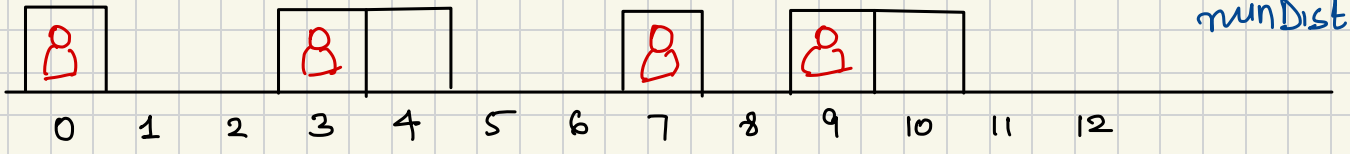
ans!

max { min Dist. bet^n any two cows }



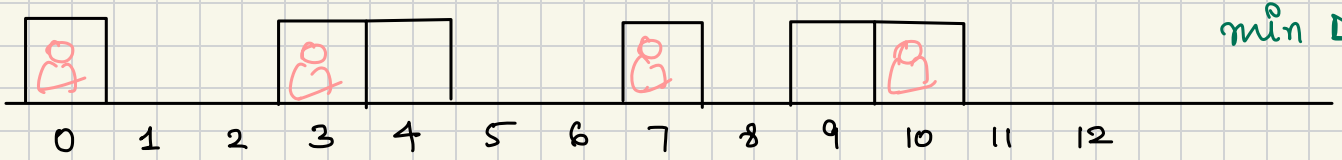
minDist = 5 units

cowsplaced = 2



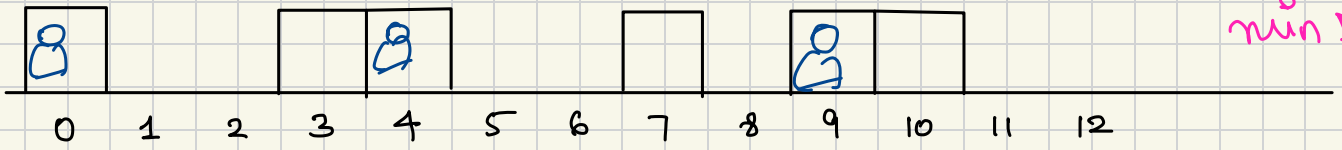
minDist = 2 units

cowsplaced = 4



min Dist = 3 units

cocos placed = 4

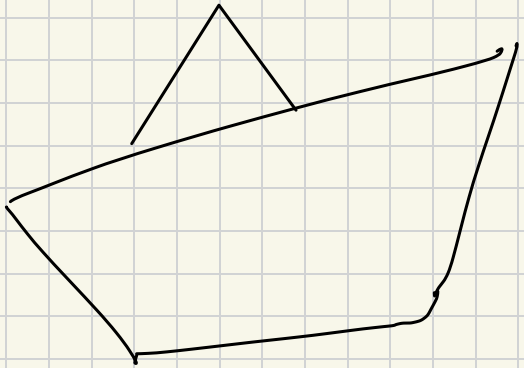


min Dist = 4 units

cocos placed = 3

Capacity of ship packages within D days

int[] A = { 3 (0), 2 (1), 2 (2), 4 (3), 1 (4), 4 (5) } day = 3



min^m { max^m capacity of ship to deliver from one port to another within D days }

Case 1

int[] A = { 3, 2, 2, 4, 1, 4 } (0 1 2 3 4 5) days = 1

[3, 2, 2, 4, 1, 4] → 16 kg → min capacity = 16 kg

Case 2

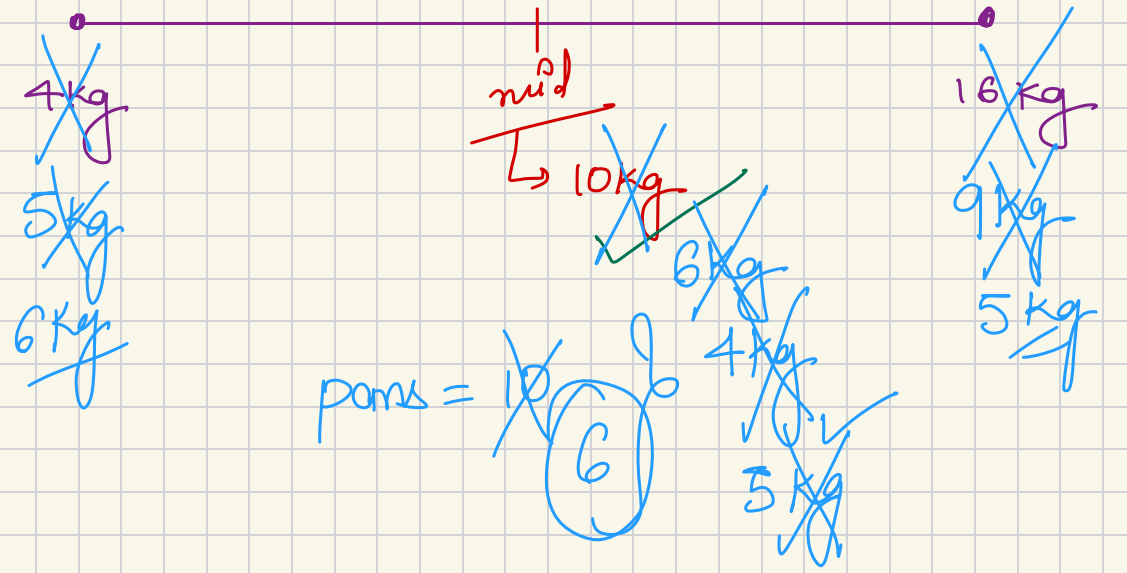
int[] A = { 3, 2, 2, 4, 1, 4 } (0 1 2 3 4 5) days = 6

[3]
[2]
[2]
[4]
[1]
[4]

→ 4 kg → min capacity = 4 kg

$1 < B < N$

day = 3.



int[] A = { 3, 2, 2, 4, 1, 4 } (0 1 2 3 4 5) maxCap = 10

d1 → 3 + 2 + 2
d2 → 4 + 1 + 4

int[] A = { 3, 2, 2, 4, 1, 4 } (0 1 2 3 4 5) maxCap = 6

d1 → 3 + 2
d2 → 2 + 4
d3 → 1 + 4

Binary Search

Define Range
remove half

$N \rightarrow \frac{N}{2} \rightarrow \frac{N}{4} \rightarrow \cdots \rightarrow 1$

TC: $O(\log_2 N)$

SC: $O(1)$

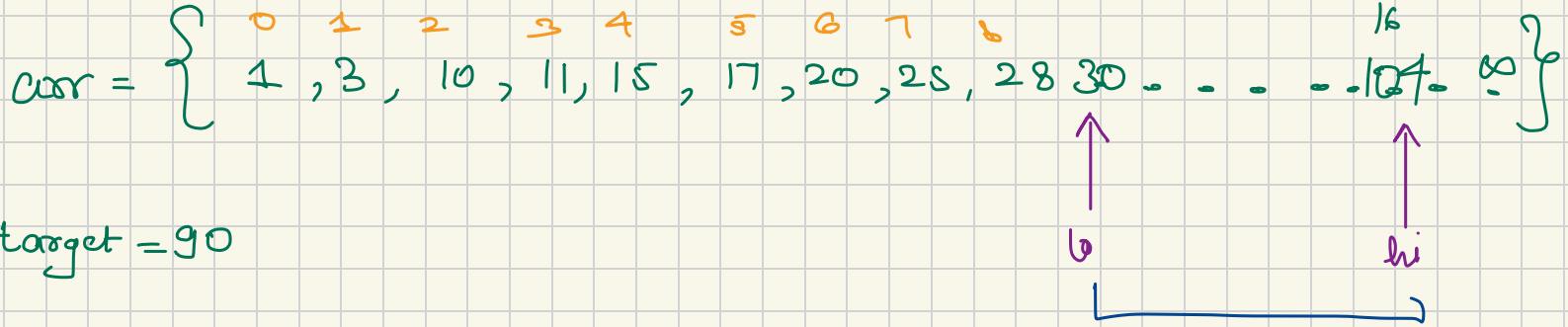
Infinite Sorted Array

arr = { 1, 3, 10, 11, 15, 17, 20, 25, 28 ..... ∞ }

target = 1014

Brute Force

↳ Linear search { TC : O(M) } ✓



target = 90

Range = [0, 1] ⇝ 2 X

1 Range = [2, 2] ⇝ 1 ⇝ 2^0

2 Range = [3, 4] ⇝ 2 ⇝ 2^1

3 Range = [5, 8] ⇝ 4 ⇝ 2^2

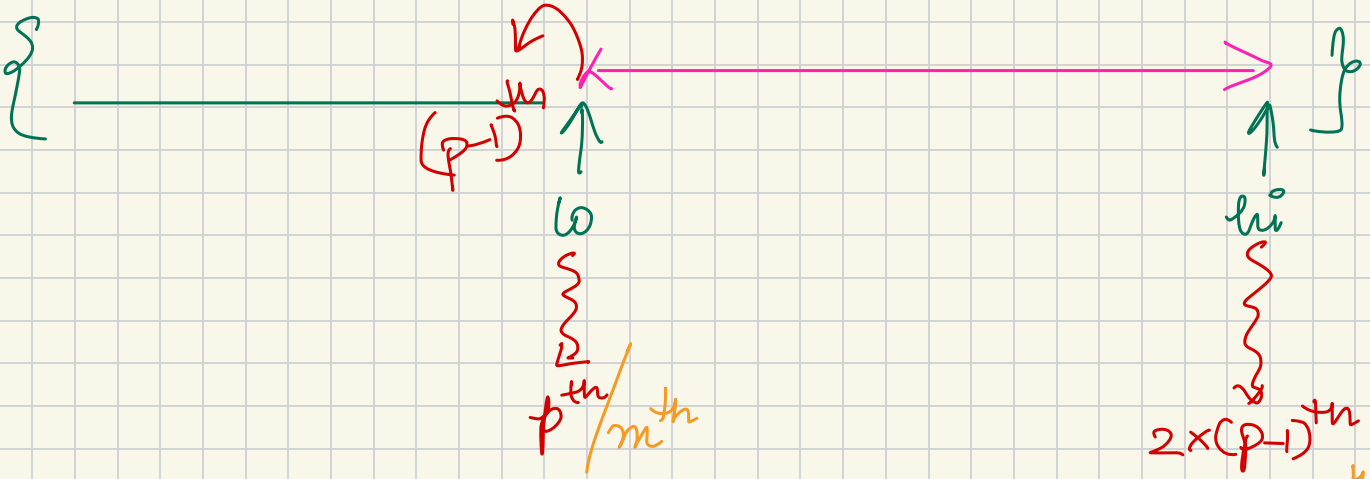
4 Range = [9, 16] ⇝ 8 ⇝ 2^3

⋮

K ⇝ 2^{K-1}

$lo' = hi + 1$

$hi' = hi * 2$



$lo = m^{th}$

$hi = 2 \times (m-1)^{th}$

No. of Ele $= hi - lo + 1 = 2m - 2 - m + 1$

$= \underline{\underline{m - 1}}$

TC : $O(\log_2(m))$ ✓

$$m-1 = 2^{k-1}$$

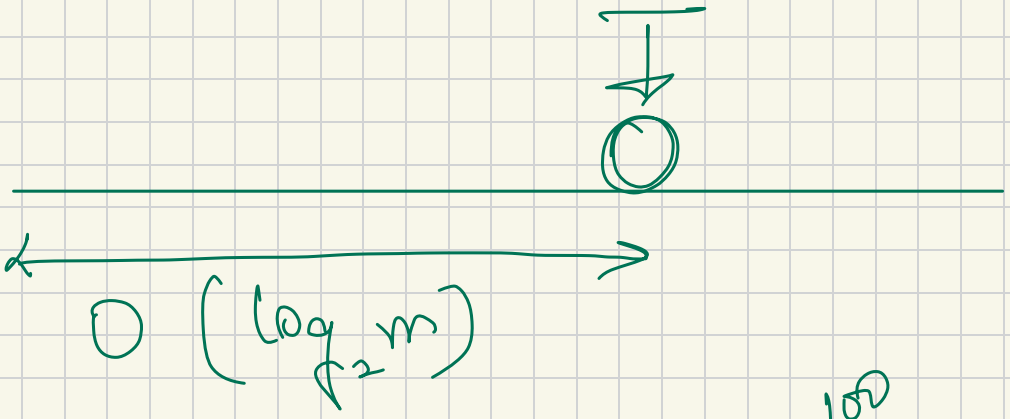
$$\log_2(m-1) = k-1$$

i.e. $$k = \log_2(m-1) + 1$$

TC : $O(\log_2 m)$

Total TC : $\log_2 m + \log_2 m = 2\log_2 m$

$= O(\log_2 m)$



m < n

optimized BS → TC: O(log m) → 100

normal BS → TC: O(log n) → 2 log2 10 → 10000000

10^8

2 log2 10

Min^m limit of Balls in a Bag.

arr[] = { 2 , 4 , 8 , 2 }
(0) (1) (2) (3)

maxOpt = 4

(1,3) (2,2)

min { max^m no. of Balls in a bag. }

arr[] = { 2, 4, 8, 2 } opt = 4

(3,1) (2,2)

Opt 1.

{ 2, 3, 1, 8, 2 }

(1,7) (2,6) (3,5) (4,4)

opt 2.

{ 2, 3, 1, 2, 6, 2 }

(1,5) (2,4) (3,3)

opt 3

{ 2, 3, 1, 2, 3, 3, 2 }

(1,2)

opt 4

penality (3)

{ 2, 3, 1, 2, 3, 1, 2, 2 }

Case 1 . opt = ∞

{ 2, 4, 8, 2 }

↓

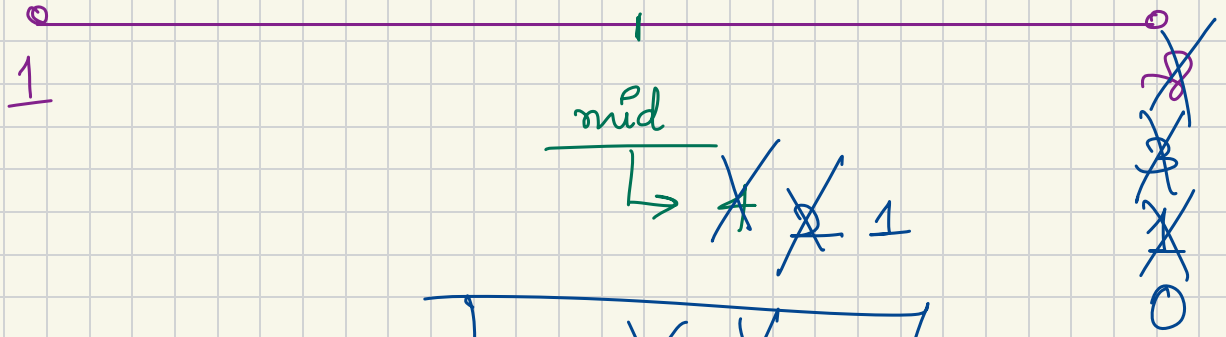
{ 1, 1, 1 . . . . . 1 } Penality = 1

Case 2 opt = 0

{ 2, 4, 8, 2 } Penality = 8

$0 \leq opt < \infty$

$opt = 4$



ans = ~~1~~ ~~2~~ 2

arr = {2, 4, 8, 2}

(4, 4)

penalty = 1

opt = 1

arr = {2, 4, 8, 2}

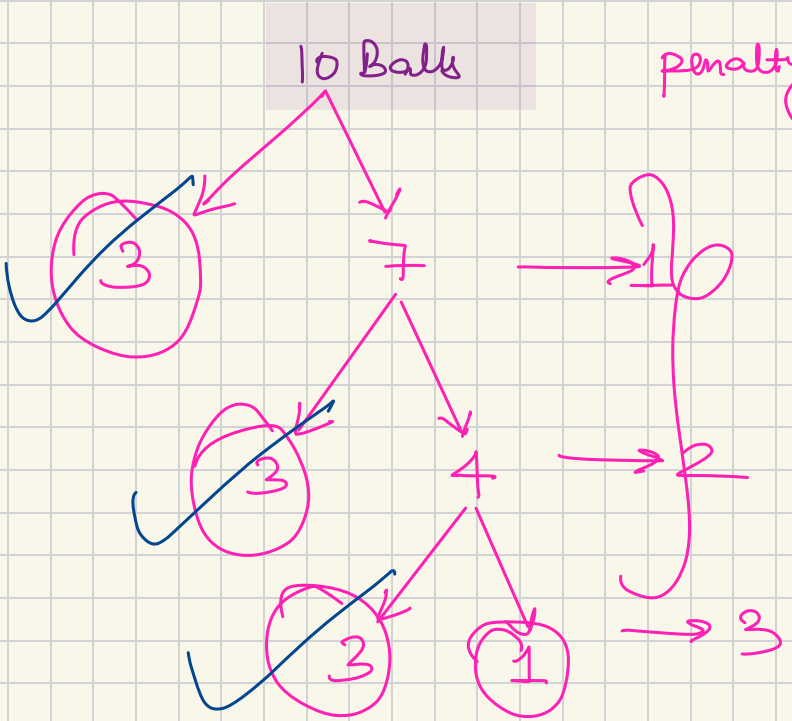
(2, 2) (2, 6)

(2, 4)

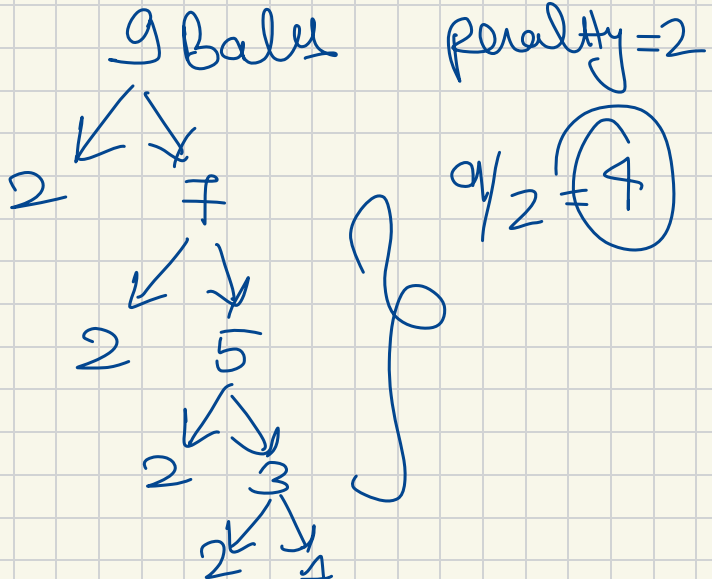
(2, 2)

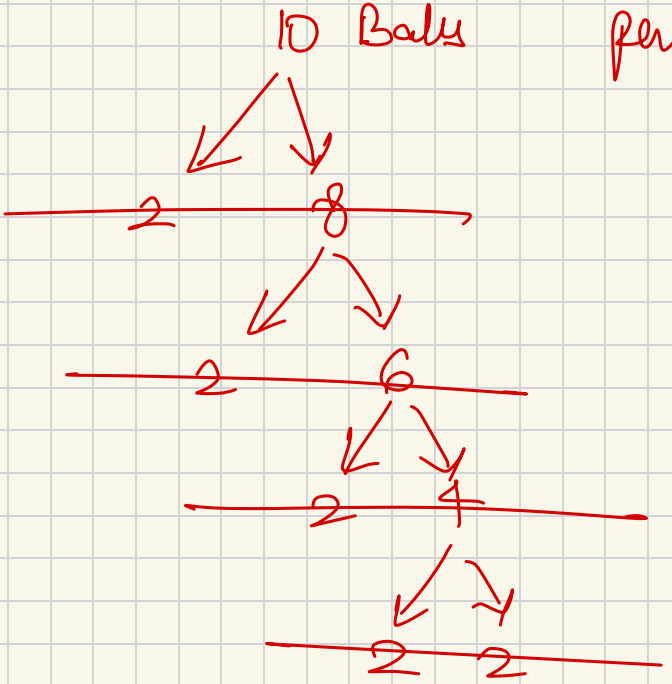
penalty = 2

opt = ~~0~~ ~~1~~ 1


10 Balls
penalty = 3
3
7
10
3
4
2
3
1
3

$10/3 = 3$ opt


9 Balls
penalty = 2
9/2 = 4
2
7
2
5
2
3
2
1



penalty = 2

$10/2 = 5 \times 4$

$$\text{minOp} = \begin{cases} \text{Balls/penalty} & ; \text{Not divisible} \\ \text{Balls/penalty} - 1 & ; \text{Ceil} - 1 \end{cases}$$