Binary Search

0	1	2	3	4	5	6	7	8	9

arr[] = { ~~1~~, ~~2~~, ~~3~~, ~~4~~, ~~5~~, ~~6~~, ~~7~~, ~~8~~, 9, 10 }

si   si   ei

mid

target = 9

TC : O(logN) — length of array

SC : O(1)

1. define range when target can be find.
2. get the mid point
3. take a decision to eleminate half of the range
4. and move towards the part where your answer lies.

Binary Search over an infinite sorted array

target = 7

int[] arr = { 0, 1, 2, 3, 4, 5, 6, 7 .... ∞ }

NOTE: We don't know length of array, but all the ele. are sorted in inc. order

2 1 2 4 8

0 1 2 3 4 5 6 7 8 9 10 11

arr[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, ... ∞ }

si ei

target = 10

All the Ele from si to ei are sorted.

Range [arr[si], arr[ei]]

si = 0, ei = 1 → [1, 2]

si = 2, ei = 2 → [3,3]

[4,5] { si = ei + 1; ei = 2 * ei; }

si = 3, ei = 4

si = 5, ei = 8 | si = 9, ei = 16

$\{a, b, c, d, e, \ldots\ldots, x \ldots\ldots\}$

1st jump 2nd jump 3rd jump 4th jump

$$arr[\,] = \{ \overset{0}{1}, \overset{1}{2}, \overset{2}{3}, \overset{3}{4}, 5, 6 \ldots \ldots \ldots \underset{(p-1)^{th}}{\ldots}) \overset{p^{th}}{(x} \ldots \ldots \infty \}$$

$si = p$

$ei = 2 * (p-1)$

Number of Elements in this range $= ei - si + 1$

$= 2p - 2 - p + 1$

$= (p - 1) \rightsquigarrow$ Elements

$$\boxed{\text{TC} : O(\log_2 p) \text{ of BS.}}$$

p → index of target Element

ei	$1 = 2^0$	$2 = 2^1$	$4 = 2^2$	$8 = 2^3$	$16 = 2^4$	- - - - - - - -	$2(p-1) = 2^{K-1}$
jump	1	2	3	4	5		K

$$2(p-1) = 2^{K-1}$$

$$(p-1) = \frac{2^{K-1}}{2}$$

$$(p-1) = 2^{K-2}$$

take $\log_2$ Both Sides

$$\log_2 (p-1) = (K-2) \cancelto{1}{\log_2 2}$$

$$\boxed{\log_2 (p-1) + 2 = K}$$

TC for K Jumps

$= O(K)$

$\approx O(\log_2 P)$

TC : to find range = $O(\log_2 P)$

TC : to find in range = $O(\log_2 P)$

Total TC : = $2O(\log_2 P)$

$$\boxed{= O(\log_2 P)}$$

→ index of target element

```
int find ( int[] arr , int target )
{
    if (target < arr[0])
        return -1;

    int si=0;
    int ei = 1;

    while( arr[ei] < target)
    {
        si = ei+1;
        ei = 2*ei;
    }

    int indx = BinarySearch(arr, si, ei, target);

    return index;
}
```

$\boxed{TC: O(\log_2 f)}$

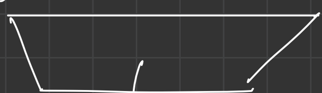
$\boxed{SC: O(1)}$

Capacity to Ship Packages within B days

$A[\,] = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ days = 5

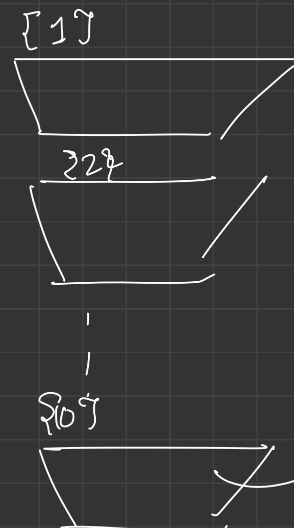
1 day

$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

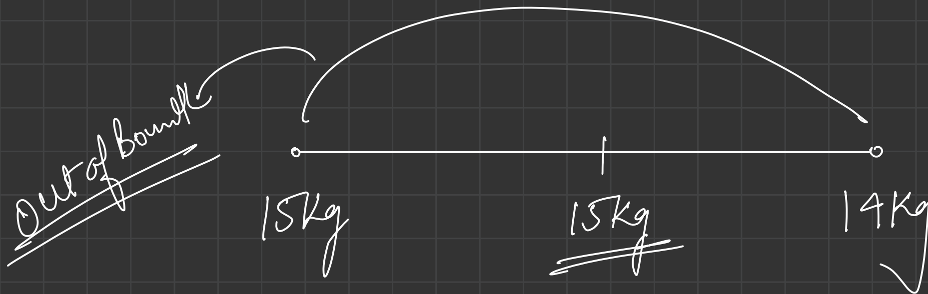


55kg

10days



10Kg



p ans = ~~32~~

~~20~~

~~17~~

15

A[ ] = { 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10 }   B = 5 days

capacity = 15 kg

day1 → 1 + 2 + 3 + 4 + 5

day2 → 6 + 7

day3 → 8

day4 → 9

day5 → 10

Minimum Limit of balls in a Bag

arr [] = { 2, 4, 8, 2 }   maxOpt = ~~4~~ ~~3~~ ~~2~~ 1

(1, 3) (2, 2)

{ 2, 1, 3, 8, 2 }

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

{ 2, 1, 3, 3, 5, 2 }

(2, 3) (1, 4)

{ 2, 1, 3, 3, 1, 4, 2 }

(1, 3) (2, 2)

{ 2, 1, 3, 3, 1, 2, 2, 2 } → max = 3

$arr[\,] = \{2, 4, 8, 2\}$ maxOpt = 4

maxOpt = ∞

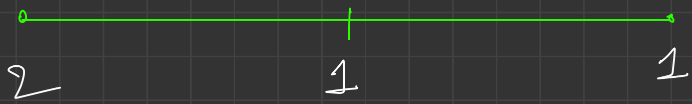
$\{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\}$

penality = 1

maxOpt = 0

penality = 8

pans = ~~1~~ 2



opt = 4

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

maxBalls = 1

(1,1) (2,2)

(1,1) (1,1)

opt - ~~1~~ ~~2~~ ~~3~~ 4

$\{2, 4, 8, 2\}$

$\downarrow$

$(2,2)$

$maxOpt = 4$

$\{2, 2, 2, 8, 2\}$

$\downarrow$

$(4,4)$

$\{2, 2, 2, 4, 4, 2\}$

$\downarrow$

$(2,2)$

$\{2,2,2,2,2,2,2,2\}$

penality = 2

$\{2, 2, 2, 2, 2, 4, 2\}$

$\downarrow$

$(2,2)$

$\{2, 4, 8, 2\}$

$ans = 4$

1 mid 3

$\cancel{4}$ 2

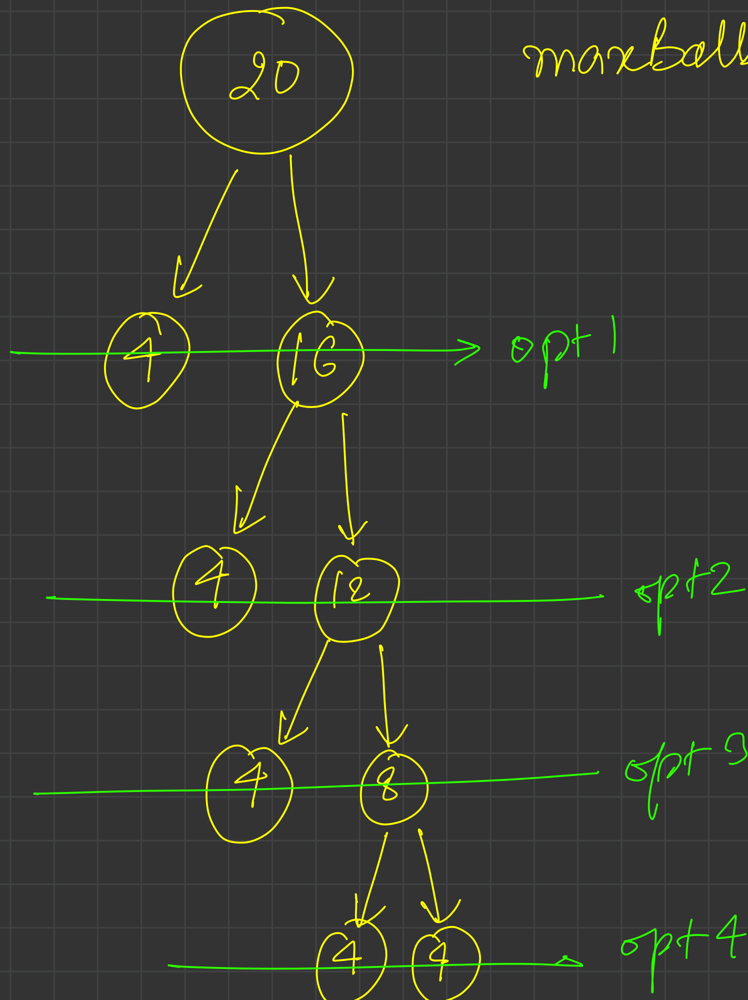
$\{2, 4, 8, 2\}$

$maxBalls = 4$

$opt = 1$

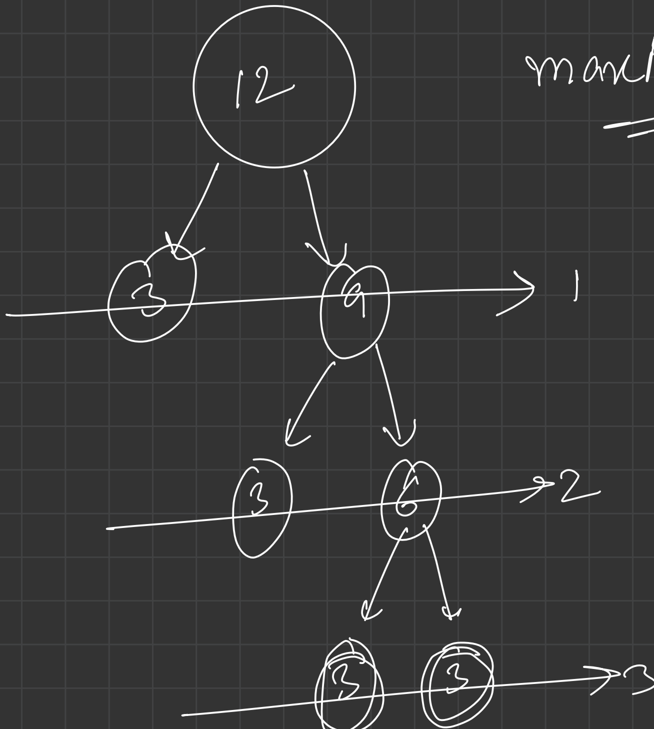
$$\left(\frac{8}{4} - 1\right) = \underline{\underline{1}}$$

maxBalls = 4



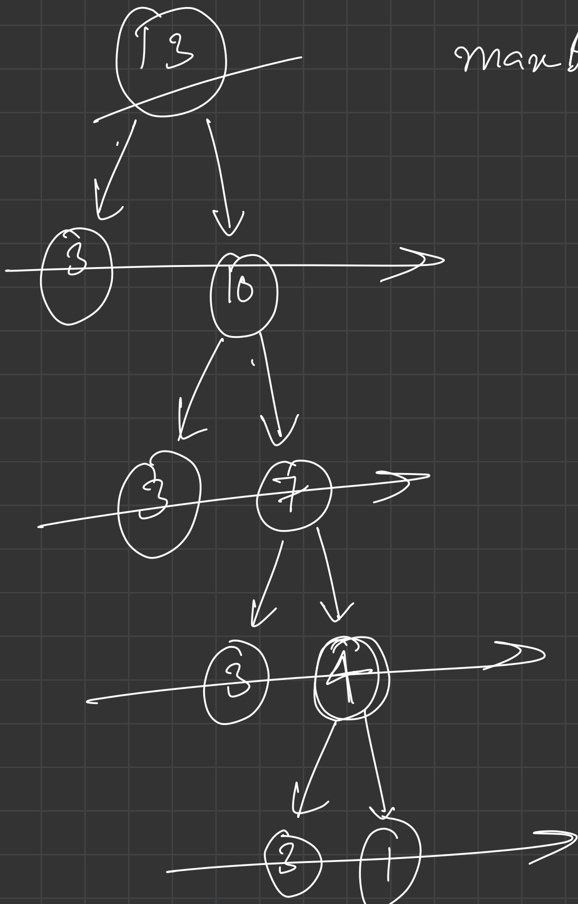
opt

$$\left(\frac{\text{Balls In Bag}}{\text{max Balls}} - 1\right)$$



maxballs = 3

$$\left(\frac{12}{3} - 1\right) \text{opt}$$



maxballs = 3

Number of opt

if (Balls % maxBalls == 0)

{ opt = $\frac{Balls}{maxballs} - 1$; }

else

{ opt = $\frac{Balls}{maxballs}$; }