SOE Time Complexity:

table : 2   table : 3   . . . . $\sqrt{n}$

$$\frac{n}{2} + \frac{n}{3} + \frac{n}{5} + \frac{n}{7}$$

$$n\underbrace{\left(\frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} \cdots\right)}_{\log\log n}$$

$$O(n \log\log n)$$

log n

$$\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots$$

log log n

$$\frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} \cdots$$

$$n = 10^{10^{10}}$$

$$\log n = 10^{10}$$

$$\log\log n = 10$$

Recursive Program Time Complexity

→ Recurrence Relation

→ Shortcut

→ Master Theorem

```
TOH(n, S, D, H)
{
   TOH(n-1, S, H, D)
   pf(move n from S to D)
   TOH(n-1, H, D, S)
}
```

Recurrence Relation:

$$T(n) = T(n-1) + 1 + T(n-1)$$

$$T(n) = 2\cancel{T(n-1)} + 1$$

$$2\,\cancel{T(n-1)} = 2^2\,\cancel{T(n-2)} + 1 \cdot 2$$

$$2^2\,\cancel{T(n-2)} = 2^3\,\cancel{T(n-3)} + 1 \cdot 2^2$$

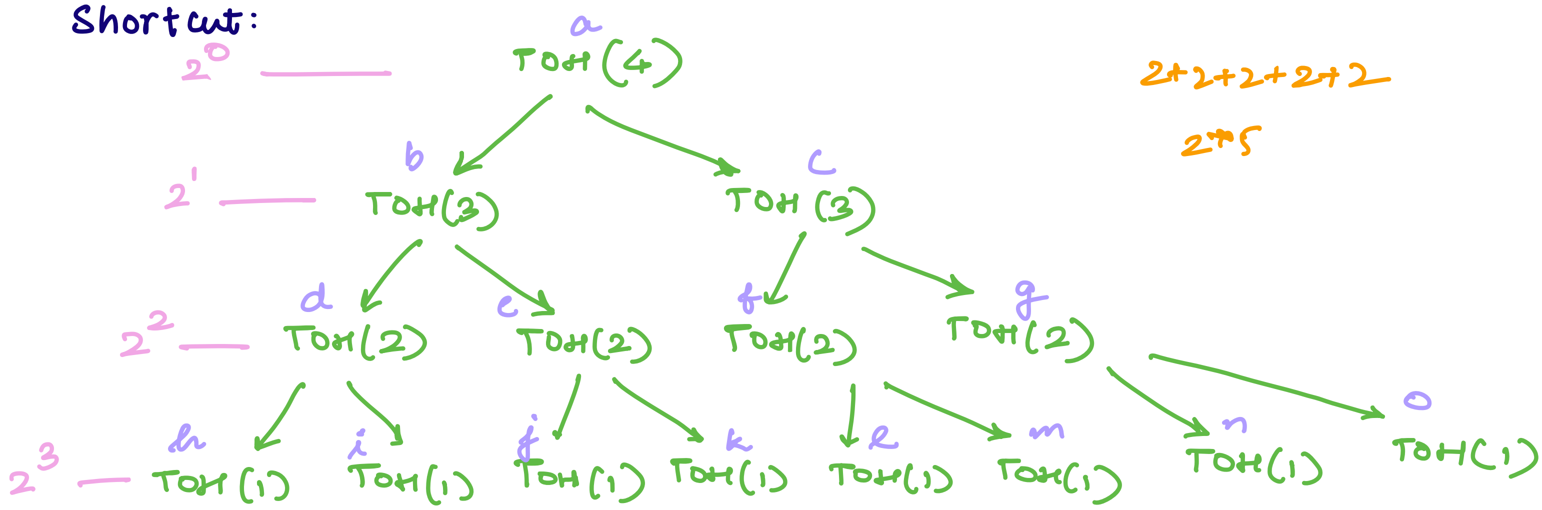
$$\vdots$$

$2^{n-1} T(n-(n-1)) = 1 \cdot 2^{n-1}$

$T(n) = 2^0 + 2^1 + 2^2 + \dots 2^{n-1}$ $\qquad a\left(\frac{r^n-1}{r-1}\right)$

$= 1\left(\frac{2^n-1}{2-1}\right) = \underbrace{2^n-1}_{f(n)} \le \underbrace{1\cdot 2^n}_{c\,g(n)} = O(2^n)$

Shortcut:



2+2+2+2+2

2*5

Time: $a+b+c+d+e+f+g+h+i+j+k+l+m+n+o$

If in every frame same amount of work then,

no. of fx^n frame * work

$2^0 + 2^1 + 2^2 + \text{——} 2^{n-1}$

$2^n - 1$

$TC = (2^n-1) * \overset{\text{print}}{1} = 2^n - 1 = O(2^n)$

<u>**Shortcut:**</u> if ≥ 2 Rec calls & same work in each fx^n frame

$TC = \underbrace{\text{no. of fx}^n\text{ frames}}_{\text{calls}} * \text{work}$

ht * work

Masters theorem

$$T(n) = aT\left(\frac{n}{b}\right) + n^k \log^p n$$

$a \geq 1$, $b > 1$, $k \geq 0$, p real no.

1) if $a > b^k$ then $T(n) = \Theta\left(n^{\log_b a}\right)$

2) if $a = b^k$

   a) if $p > -1$ then $T(n) = \Theta\left(n^{\log_b a} \log^{p+1} n\right)$

   b) if $p = -1$ then $T(n) = \Theta\left(n^{\log_b a} \log \log n\right)$

   c) if $p < -1$ then $T(n) = \Theta\left(n^{\log_b a}\right)$

3) if $a < b^k$

   a) if $p \geq 0$ then $T(n) = \Theta\left(n^k \log^p n\right)$

   b) if $p < 0$ then $T(n) = O\left(n^k\right)$

$$T(n) = 2T\left(\frac{n}{4}\right) + n^2$$

$a = 2$
$b = 4$
$k = 2$
$p = 0$

$a < b^k$
$2 < 4^2$

$$n^k \log^p n = n^2 (\log n)^0 = n^2$$

Binary Search:

sorted

item: 30 ?

0	1	2	3	4	5	6	7	8
5	10	15	20	25	30	35	40	45

~~X~~ lo mid ~~X~~ hi

lo = 0
hi = 8
mid = $\frac{0+8}{2}$ = 4

mid 25 < item 30

sorted

lo = mid + 1
= 5

lo = 5
hi = 8

mid = 6

mid 35 > item 30

hi = mid - 1
= 5

lo = 5
hi = 5
mid = 5

mid 30 == item 30

5 index found

mid = lo + $\frac{(hi - lo)}{2}$

lo = INT_MAX - 50
hi = INT_MAX - 20

4
4

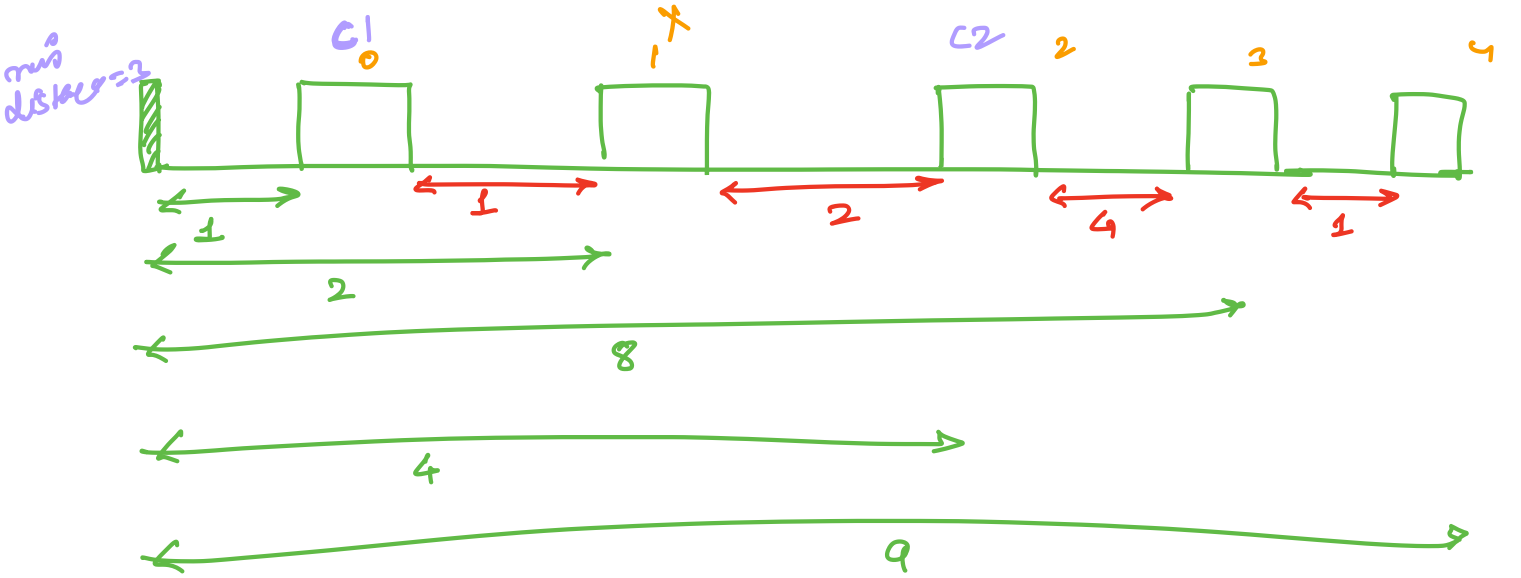
15236

Aggressive Cow

\# cows: 3
\# stalls: 5

pos: ~~f~~ 4

maximize the min distance b/w cows.



min distance = 1

C1 C2 C3

min distance = 2 — not — C1 C2 C3

min distance = 3 — C1 C2 C3

min distance = 4 — ✗ — C1 C2

not able to place all cows

min distance	0	1	2	3	4	5	6	7	8
		✓			✗	✗	✗	✗	✗

lo = 0
hi = 8
mid = 4
min distance 4 place the cows?
NO
LHS
hi = mid-1
= 3

lo = 0
hi = 3
mid = 1
min distance 1 place cows?
Yes
RHS
lo = mid+1
= 2

lo = 2
hi = 3
mid = 2
Is it possible?
Yes
RHS
lo = mid+1
= 3

lo = 3
hi = 3
mid = 3
Is it possible?
Yes
RHS
lo = 4

lo = 4
hi = 3