Q) How to find time and space complexity of Recursive program?

iterative: for (i = 1 . . . . n)

{ a=b+c } → directly we say TC.

i, a, b, c (4 x 2) Bytes.

O (1) Complexity

```
A(n)
{ if n>0
    (A(n-1) + A(n-2))
  }
  else
      return 1
}
```

$\Longrightarrow$? TC of this prog

SC of this prog.

In order to find TC of recursive program, we use recurrence relations.

$A(n) \Rightarrow T(n)$

```
{
   if (n>0)
   {
      A(n-1) + A(n-2)
   }
   else
      return 1;
}
```

$A(n-1) \Rightarrow T(n-1)$, $A(n-2) \Rightarrow T(n-2)$

$$\boxed{T(n) = \begin{cases} T(n-1) + T(n-2), & n>0 \\ 1, & n \le 0 \end{cases}}$$

we have to solve this

methods to solve recurrence relations:-

- → Substitution method
- → recursive tree method
- → masters theorem.

$$T(n) = aT(n/b) + O(n) \rightarrow \text{masters} \rightarrow \text{fast.}$$

$$T(n) = \underbrace{T(n-1) + T(n-2) + T(n-3)}_{\text{more than one terms}} \rightarrow \text{Recursive tree}$$

$$T(n) = 2\underbrace{T(n-1)}_{\text{only one term}} + n \rightarrow \text{Substitution method}$$

In substitution method and recursion tree ~~meth~~ method, we can find space complexity also.

Substitution method:

$$T(n) = \begin{cases} 1 & \text{if } n=1 \\ T(n-1)+n & \text{if } n>1 \end{cases}$$

$n=1 \rightarrow$ base condition or Termination condition

$n=1$

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

$T(n-1) = T(n-2)+n-1 \rightarrow (2)$

$T(n-2) = T(n-3)+n-2 \rightarrow (3)$

Substitute (2) in (1)

$T(n) = (T(n-2)+(n-1))+n$

$$T(n) = \underline{T(n-2)} + (n-1) + n \rightarrow 4$$

3 in 4

$$T(n) = T(n-3) + (n-2) + (n-1) + n .$$

k=3, k-2, k-1

k times ✓

$$T(n) = \underbrace{T(n-k)}_{\downarrow \text{ stops @ } T(1)} + (n-(k-1)) + (n-(k-2)) + \cdots + (n-2) + (n-1) + n$$

stops@T(1) ✓

$$n-k = 1 \Rightarrow k = n-1$$

$$SC = O(n).$$

$$T(n) = T(\underbrace{n-k}_{1}) + T(n-(k-1)) + T(n-(k-2)) + \dots + (n-2) + (n-1) + n.$$

$$n-k=1$$

$$\Rightarrow k = n-1$$

$$= T(1) + (n-(n-1)-1) + (n-(n-1)-2) + \dots + n-2 + n-1 + n$$

$$= 1 + 2 + 3 + 4 + \dots + n-2 + n-1 + n.$$

$$= \frac{n(n+1)}{2} = O(n^2)$$

$$= \Omega(n^2)$$

$$= \Theta(n^2)$$

$\rightarrow$ termination

$T(n-k)$

$n-k=0$

$k=n$

$n!$

$O(n^n)$

$=$

$T(n) = T(n-1) * n \rightarrow (1)$

$T(n-1) = T(n-2) * (n-1) \rightarrow (2)$

$T(n-2) = T(n-3) * (n-2) \rightarrow (3)$

(2) in (1)

$T(n) = T(n-2) * n-1 * n \rightarrow (4)$

(3) in (4)

$T(n) = T(n-3) * (n-2) * n-1 * n$

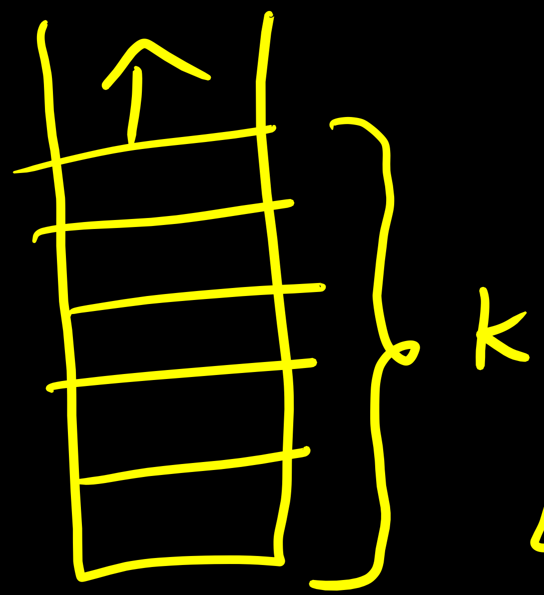
$$T(n) = T(\underbrace{n-3}_{k}) * (\underbrace{n-2}_{k-1}) * n-1 * n$$

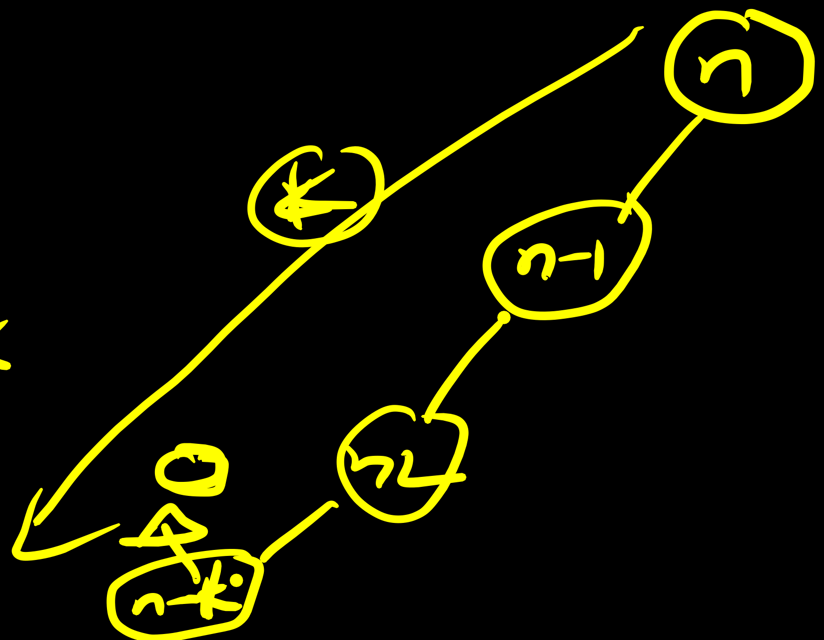
k time

$$T(n) = T(n-k) * (n-(k-1)) * (n-(k-2)) * \ldots\ldots * (n-1) * n.$$

let $n-k=0 \Rightarrow \underline{\underline{k=n}}$

what ever the value of k is, it is space complexit





$SC = O(n)$

$$T(n) = T(n-k) * (n-(k-1)) * (n-(k-2)) * \cdots\cdots * (n-1) * n.$$

$$n-k=0 \quad k=n$$

$$= T(0) * (n-(n-1)) * (n-(n-2)) * \cdots\cdots (n-1)(n)$$

$$= 1 * 1 * 2 * 3 \cdots\cdots (n-1)(n)$$

$$= n!$$

$$= O(n!) \qquad O(n^n)$$

$$= \Omega(n!) \qquad O(n^n)$$

$$= \theta(n!)$$

$$T(n) = \begin{cases} 0 & \text{if } (n=0) \\ T(n-2) + n^2 & \text{if } n>0 \end{cases}$$

$(n=0) \longrightarrow T(n-k)$ ✓

let assume $n-k=0$

$T(0)=0$

$$\left.\begin{array}{c} T(n) \\ T(n-2) \\ T(n-4) \\ \vdots \\ T(n-k) \end{array}\right\} O(K) = O(n)$$

$n-k=0$

$K=(n)$

$$\left.\begin{array}{|c|} \hline T(n-k) \\ \hline \vdots \\ \hline T(n-4) \\ \hline T(n-2) \\ \hline T(n) \\ \hline \end{array}\right\} O(K) = O(n)$$

$$T(n) = T(n-2) + n^2 \rightarrow ①$$

3 equations

$$T(n-2) = T(n-4) + (n-2)^2 \rightarrow ②$$

$$T(n-4) = T(n-6) + (n-4)^2 \rightarrow ③$$

② in ①

$$T(n) = T(n-4) + (n-2)^2 + n^2 \rightarrow ④$$

③ in ④

$$T(n) = T(n-6) + \underbrace{(n-4)^2}_{2k-2} + (n-2)^2 + n^2$$

$\underbrace{}_{2k}$ } k times

$$T(n) = T(n-2k) + (n-(2k-2))^2 + \cdots\cdots + (n-2)^2 + n^2$$

let $(n-2k) = 0$

$n = 2k$; $k = n/2$; $SC = O(n)$.

$$T(n) = T(0) + 2^2 + 4^2 + 6^2 + 8^2 + \cdots\cdots + n^2$$

$$= \cancel{T(0)} \; (2\times1)^2 + (2\times2)^2 + (2\times3)^2 + \cdots\cdots (2\times\tfrac{n}{2})^2$$

$$= 2^2(1^2 + 2^2 + 3^2 + \cdots\cdots + n/2)^2$$

$$= 2^2\left(1^2+2^2+3^2+\cdots+\left(\frac{n}{2}\right)^2\right)$$

$$= 2^2\left(\frac{\frac{n}{2}\left(\frac{n}{2}+1\right)\left(2\frac{n}{2}+1\right)}{6}\right)$$

$$1^2+2^2+3^2+\cdots n^2 = \frac{n(n+1)(2n+1)}{6}$$

$$\frac{n}{2} \qquad \frac{\left(\frac{n}{2}\right)\left(\frac{n}{2}+1\right)\left(2\frac{n}{2}+1\right)}{6}$$

$$TC \begin{cases} = O(n^3) \\ = \Omega(n^3) \\ = \Theta(n^3) \end{cases} \qquad SC = O(n)$$

$$T(n) = \begin{cases} 0 & \text{if } n = 0 \\ T(n-2) + \log_2 n \end{cases}$$

$\rightarrow T(n-2k)$

$\rightarrow n - 2k = 0$

$n = 2k$ & $k = n/2$

$SC = O(n)$

$n-4 \qquad n-4 \qquad n-4$

$T(n) = T(n-2) + \log_2 n \rightarrow (1)$

$T(n-2) = T(n-4) + \log_2 (n-2) \rightarrow (2)$

$T(n-4) = T(n-6) + \log_2 (n-4) \rightarrow (3)$

(2) in (1)

$$T(n) = T(n-4) + \log_2 (n-2) + \log_2 n \rightarrow 4$$

3 in 4

$$= T(n-6) + \log_2 (n-4) + \log_2 (n-2) + \log_2 n$$

{ K times

$$= \cancel{T(n-2K)} + \log_2 (n-(2K-2)) + \log_2 (n-(2K-4)) + \dots\dots + \log_2 (n-2) + \log_2 n$$

n
$n-2$
$n-4$
$n-6$
$\vdots$
$n-2K$

$$n - 2K = 0$$

$$= \log 2 + \log 4 + \log 6 + \dots\dots + \log_2 n$$

$= \log 2 + \log 4 + \log 6 + \cdots + \log_2 n$ $\qquad n! = O(n^n)$

$= \log 2*1 + \log 2*2 + \log 2*3 + \cdots + \log 2*n/2$

$\log a*b = \log a + \log b$

$= \log 2 + \log 1 + \log 2 + \log 2 + \log 2 + \log 3 + \cdots + \log 2 + \log n/2$

$= \log_2 2\,(n/2) + (\log 1 + \log 2 + \log 3 + \cdots \log n/2)$

$\Theta(n \log n)$

$= n/2 + \log 1.2.3.\cdots n/2$ $\qquad = \Omega(n \log n)$

$= n/2 + \log (n/2)! = O(\log (n/2)!) = O(\log n^n)$

$= O(n \log n)$

$$T(n) = \begin{cases} 1 \\ T(n/2) + c \end{cases}$$

Easy in masters theorem method.

But we want to practice in substitution method just for practicing.

So in exam/inter use masters theorem for this example.

$$T(n) = T(n/2) + c \rightarrow (1)$$

$n/4 \qquad n/4$

$$T(n/2) = T(n/4) + c \rightarrow (2)$$

$$T(n/4) = T(n/8) + c \rightarrow (3)$$

(2) in (1)

$$T(n) = T(n/4) + c + c = T(n/4) + 2c \rightarrow (4)$$

(3) in (4)

$$T(n) = T(n/8) + 3c.$$

$$T(n) = T(n/8) + 3C$$

k times

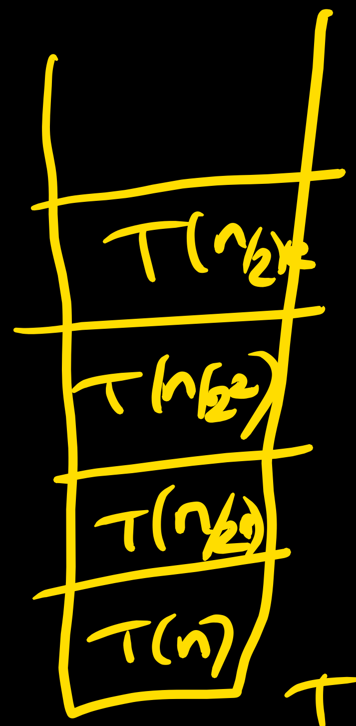
$$T(n) = T\left(n/2^k\right) + kC$$

let $n/2^k = 1$

$n = 2^k$; $k = \log_2 n$ $SC = O(\log n)$

$$T(n) = T(1) + \log_2 n \, C$$

$$\therefore T(n) = O(\log_2 n) \quad \Omega(\log_2 n) \quad \Theta(\log_2 n)$$



$T(n)$

$T(n/2^1)$

$T(n/2^2)$

$\vdots$

$T(n/2^k)$

k.

$$T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + n & \text{if } n > 1 \end{cases}$$

go ~~fote~~ for masters theorem in exam

Here we do it only for practice.

$$T(n) = T(n/2) + n \rightarrow (1)$$

$$T(n/2) = T(n/4) + n/2 \rightarrow (2)$$

$$T(n/4) = T(n/8) + n/4 \rightarrow (3)$$

Subs (2) in (1)

$$T(n) = T(n/4) + n/2 + n \rightarrow (4)$$

(3) in (4)

$$T(n) = \underbrace{T(n/8) + n/4 + n/2 + n}$$

$$T(n) = T(n/8) + n/4 + n/2 + n$$

$$= \underbrace{T(n/2^3) + n/2^2 + n/2^1 + \frac{n}{2^0}}_{k \text{ times}}$$

$$\frac{n}{2^k} = 1$$

$$n = 2^k$$

$$= T(n/2^k) + n/2^{k-1} + \frac{n}{2^{k-2}} + \cdots + \frac{n}{2^2} + \frac{n}{2^1} + \frac{n}{2^0}$$

let $\frac{n}{2^k} = 1$

$$\frac{n}{2^{k-1}} = \frac{n}{2^k * 2^{-1}} = \frac{n}{2^k} * 2 = 1 * 2$$

$$\frac{n}{2^k} = 1 \qquad \frac{n}{2^{k-2}} = \frac{n}{2^k * 2^{-2}} = \frac{n}{2^k} * 2^2 = 2^2$$

$$n = 2^k.$$

$$2^0 + 2^1 + 2^2 \ldots 2^k.$$

$= 1 + 2^1 + 2^2 + \dots\dots\dots 2^k$

$\frac{n}{2^k} = 1$

$n = 2^k$

$k = \log n$

$$\frac{a(r^n - 1)}{r - 1} \quad r > 1$$

$(\log_2 n + 1)$

$= \underbrace{2^0 + 2^1 + 2^2 + \dots + 2^{\log_2 n}}$

$= \frac{2^0 (2^{\log_2 n + 1} - 1)}{2 - 1}$

leading

$= 2^{\log_2 n}$

$= n = O(n)$

$\Theta(n) \quad \Omega(n)$

$$T(n) = \begin{cases} 1 \\ 2T(n/2) + n \end{cases}$$

In interview/exam go with masters

Here we just practice.

$$T(n) = 2T(n/2) + n \rightarrow (1)$$

$$T(n/2) = 2T(n/4) + n/2 \rightarrow (2)$$

$$T(n/4) = 2T(n/8) + n/4 \rightarrow (3)$$

Subs (2) in (1)

$$T(n) = 2\left(2T(n/4) + n/2\right) + n$$

$$= 2^2 T(n/2^2) + 2n \rightarrow (4)$$

$$T(n) = 2\left(2T(n/4) + n/2\right) + n$$

$$= 2^2 T(n/2^2) + 2n \rightarrow 4$$

Subs 3 in 4

$$= 2^2\left(2T(n/2^3) + \frac{n}{4}\right) + 2n$$

$$= 2^3 T(n/2^3) + 3n$$

$$2^k T(n/2^k) + kn = 2^k(1) + kn$$

$$\Downarrow \quad n/2^k = 1$$

$$= 2^{\log_2 n} + \log n (n)$$

$$= n + (n \log n)$$

$$\boxed{a^{\log_c b} = b^{\log_c a}}$$

$$2^{\log_2 n} \Rightarrow n^{\log_2 2}$$

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

$$= \Omega(n \log n) \quad \Theta(n \log n)$$