Single_NestedLoop_Complexity

Monday, August 2, 2021 3:11 PM

Time Complexity → no of key steps in terms of input size n

```
print("Hello")  # 1
for (i=1; i<n; i++)
    print(i)  #2 . n → K
print("Bye")  # 1
```

Iteration	1	2	3	4	...	K	K+1
i	1	2	3	4	...	K	K+1

→ Condition False

1, 2, 3, 4 ... n

---- n+1

i = n → False

K+1 = n

K = n-1

T(n) = K

T(n) = n-1

$f(n) = 3n$

$g(n) = n$

$\boxed{f(n) = O(g(n))}$

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

$= 3\boxed{n} + 3$ → constant

$= O(n)$

$\boxed{T(n) = O(n)}$

$g(n) \geq c \cdot f(n)$

$\boxed{2n = 5n = 6n \Rightarrow O(n)}$

```
i = 1  # 1
while i <= n
DO
    print(i)  # K
    i = i*2   5 → log_2(n)
Done
```

Azure

$T(n) = K \quad \cdots\cdots (i)$

$\boxed{T(n) = \log(n)}$

(K+1)

Iteration	1	2	3	4	5	...	K	K+1
Variable i	1 $2^0=1$	2 $2^1=2$	4 $2^2=4$	8 $2^3=8$	16 $2^4=16$	...	2^{K-1}	2^K

K+1 → Loop condition will be false

$i = n$

$2^K = n \Rightarrow \log 2^K = \log n$

$K \log_2 2 = \log_2 n$

$\boxed{K = \log n}$

```
i = n
while i > 1   False
Do
    print(i)  # K
```

K+1

K+1 iteration will make

i > 1 condition false

Anchor con

End condition

Iteration	1	2	3	4	5	...	K	K+1

Do

print(i) # K

$i = i/2$

Done

End condition

iteration	1	2	3	4	5	...	K	K+1
i →	$n = \frac{n}{2^0}$	$\frac{n}{2} = \frac{n}{2^1}$	$\frac{n}{4} = \frac{n}{2^2}$	$\frac{n}{8} = \frac{n}{2^3}$	$\frac{n}{16} = \frac{n}{2^4}$	...	$\frac{n}{2^{k-1}}$	$\frac{n}{2^k}$

$T(n) = K$

$$\boxed{T(n) = \log n}$$

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

$\log_2 2^k = \log_2 n \Rightarrow k \cancel{\log_2 2} = \log_2 n$

$$\boxed{K = \log_2(n)}$$

$i = 2$

while $i <= n$

Do

print(i) # K

$i = i^2$

Done

$T(n) = K$

$$\boxed{T(n) = O(\log \log n)}$$

Anchor

iterations	1	2	3	4	5	...	K	K+1
i	$2 = 2^{2^0}$	$4 = 2^{2^2}$	$16 = 2^{2^2}$	$256 = 2^{2^3}$	$65535 = 2^{2^4}$	...	$2^{2^{k-1}}$	2^{2^k}

$2^1 \quad 2^2 \quad 2^4 \quad 2^8 \quad 2^{16}$

$2^{2^0} \quad 2^{2^1} \quad 2^{2^2} \quad 2^{2^3} \quad 2^{2^4}$

$\boxed{i > n}$ → Anchor

$2^{(2^k)} = n$

$\log_2 2^{(2^k)} = \log_2(n)$

$2^k \cancel{\log_2(2)} = \log_2(n)$

$\log 2^k = \log_2 \log_2 n$

$k \cancel{\log_2 2} = \log_2 \log_2 n$

$$\boxed{k = \log_2 \log_2 n}$$

Question

H/W

```
i = n
WHILE  i > 1
DO
    print(i)
    i = √i
Done
```

log log n

Nested loop time Complexity

Independent

```
for (i=1; i<n; i++)
    for (j=1; j<n; j=j*2)
        print(i,j)
```

Dependent

```
for (i=1; i<n; i++)
    for (j=1; j<=i; j++)
        print(i,j)
```

inner loop depends on outer loop variable

```
for ( i=1; i<n ; i++ )
    for ( j=1; j<n ; j=j*2 )
Body    print (i,j)
```

B

A

$B = \log(n)$

$A = n$

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

$$T(n) = O(A \cdot B)$$ for independent loop

iteration	1	2	3	...	K

$= 2+2+2+2$

$= 2 \times 4$

iteration	1	2	3	...	k
i	1	2	3	...	n → n
j	log(n)	log(n)	log(n)	...	log(n)

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

$= \log(n)\,(1 + 1 + 1 + 1 + \cdots n)$

$= n \log(n)$

```
for (i = 2 ; i <= n^2 ; i = i + 2)
    for (j = n/2 ; j <= n ; j = j * 2)
        for (k = n^2 ; k >= 2 ; k = √k)
            print ("Hello")
```

$Z = O(n^2)$

$y = O(1)$

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

$T(n) = O(x \cdot y \cdot z) \Rightarrow O(n^2 \cdot 1 \cdot \log\log n) \to O(n^2 \log \cdot \log(n))$

Iteration	1	2	3	4	5	...	z	z+1
i	2 2×1	4 2×2	6 2×3	8 2×4	10 2×5	...	2×z	2z+1

$i = 2 * z$ when condition will be False

$2 * z = n^2 \Rightarrow z = \frac{n^2}{2}$

$\boxed{z = O(n^2)}$

$$\rightarrow \boxed{Z = O(n^2)}$$

$\frac{1}{2}$

Iteration	1	2	3	4	5	...	k	k+1
j	$n/2$	n	$2*n$	$4*n$	$8*n$	...		

$2^{-1}*n$ $\quad 2^0*n$ $\quad 2^1*n$ $\quad 2^2*n$ $\quad 2^3*n$ $\quad 2^{k-2}*n$

$1-2 \rightarrow -1$
$2-2 \rightarrow 0$
$3-2 \rightarrow 1$
$4-2 \rightarrow 2$

$$\boxed{2^k \cdot \cancel{n} = \cancel{n}}$$

$$2^k = 1$$

$$\log_2(2^k) = \underline{\log(1)}$$

$$\boxed{k = c}$$

$k = n^2$; $k >= 2$; $k = (\underline{k})^{1/2}$

Iteration	1	2	3	4	5	...	k	k+1
k	n^2	n	$n^{1/2}$	$n^{1/4}$	$n^{1/8}$	...	.	

$n^{\frac{1}{2^{-1}}}$ $\quad n^{\frac{1}{2^0}}$ $\quad n^{1/2^1}$ $\quad n^{\frac{1}{2^2}}$ $\quad n^{\frac{1}{2^3}}$ $\quad n^{\frac{1}{2^{k-2}}}$

$$n^{\frac{1}{2^k}} = 2 \Rightarrow \frac{1}{2^k}\log(n) = \cancel{\log_2(2)}^{\,1}$$

$$\Rightarrow 2^k = \log(n)$$

$$\Rightarrow \quad 2^k = \log(n)$$

$$\Rightarrow \quad k \cancel{\log_2(2)} = \log\log(n)$$

$$\Rightarrow \quad \boxed{y = \log\log n}$$

$\rightarrow n \quad \rightarrow \quad 1 \quad 2 \quad 3 \quad \cdots\cdots \quad n$

$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$

$n \quad n \quad n \quad n \quad n \rightarrow ?$

$\rightarrow (n + n + n + n + n \cdots\cdots + n)$

$n\,(1+1+1+1+1+1+\cdots\cdots+1)$

$n\,(n) \rightarrow n^2$

```
for (i=1; i<n; i++)   →(n)
    for (j=1; j<n; j++)   → m
        print(i, j) # k   ] → n
```

m m

$2. \Rightarrow$

$k = m$

Iteration	1	2	3	4	....	k
i	1	2	3	4	---	n
→ j	n m	n m	n m	n m	n m	n m

$(n+n+n+ \dots + n) \Rightarrow n(1+1+1+1+ \dots +1)$ nk

$(m+m+m+ \dots +m) \Rightarrow m(1+1+1+1+ \dots +1)$

$\Rightarrow n^2$

$\Rightarrow m \cdot n$

→ for (i = n ; i > 1 ; i = i - n/2) → $O(1)$

iter	1	2	3 ↓False	
i	n	n/2	0	

for (j = 1 ; j <= n^2 ; j = j + 10) → $O(n^2)$

j x 10

iteration	1	2	3	4	...	y	x+y
j	1	11	21	31	...		

$1+10 \times \dots$ $1+10 \times 1$ ...

$1 + 10 * y = n^2$

$\boxed{y = n^2}$

for (k = n^{10} ; k > 1 ; k = k/2) → $O(\log(n))$

print(i, j, k) # K

$$\boxed{T(n) = O(n^2 \log(n))}$$

iter	1	2	3	4	5	k
k	n^{10}	$\frac{n^{10}}{2}$	$\frac{n^{10}}{4}$	$\frac{n^{10}}{8}$	$\frac{n^{10}}{16}$	-1

$\frac{n^{10}}{2^k} = 1 \rightarrow 2^k = n^{10}$

$k \log_2(2) = 10 \log_2 n$

$\boxed{k = \log_2(n)}$

→ Nested Loop

Dependent Loops

A ——————— B

n

n/2

for (i = 1 ; i <= n ; i ++)

for (j = 1 ; j <= n ; j = j + i) → (j = 1 ; j <= n ; j = j + 2)

j = j + 2

print(i, j) # K

Iter	1	2	3	4	—	k
j	1	3	5	7		→

print(i,j) # k

ith	1	2	3	4	—	k
j	1	3	5	7	—	→
	1+2×0	1+2×1	1+2×2	1+2×3		$\boxed{1+2\times k}$

$1+2k < n$

$\boxed{k = n/2}$

	1	3	.	1	1
1	2	3	4	...	n	n+1
n	n/2	n/3	n/4	...	n/n	

$n^{th} = a + (n-1)d$

$d = 3,\ a = 1$

$= 1 + (k-1)3$

$= 3k - 2$

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

$$T(n) = n\,(\underbrace{1 + 1/2 + 1/3 + 1/4 + \cdots + 1/n}_{\log(n)})$$

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

$$\boxed{1+2+3+4+\cdots n \Rightarrow \frac{n(n+1)}{2}}$$

1

1/n

```
for (i=1; i<=n; i++)
  for (j=1; j<=i^2; j++) → 9.
    for (k=1; k<=n/2; k++)
        print("Hi")
```

n/2 →

iteration	1	2	3	k
i	1	2	3	n
j	1	4	9	n^2
k	1.n/2	4.n/2	9.n/2	$n^2 \cdot n/2$

$$T(n) = \frac{n}{2} + 4\cdot\frac{n}{2} + 9\cdot\frac{n}{2} + \cdots + n^2\cdot n/2$$

$$= \frac{n}{2}\left(1 + 4 + 9 + \cdots + n^2\right)$$

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

$$T(n) = \frac{n}{2}\left(\frac{\quad}{6}\right)$$

$$= \frac{n}{12}\left((n^2+n)\cdot(2n+1)\right)$$

$$= \frac{n}{12}\left(2n^3+n^2+2n^2+n\right)$$

$$= \frac{n}{12}\left(2n^3+3n^2+n\right)$$

$$= \frac{2n^4+3n^3+n^2}{12}$$

$$= n^4+n^3+n$$

$n^4 >> n^3 >> n$

$\uparrow$

'g 'hat

$$\boxed{T(n) \qquad (n^4)}$$

Recursive Algorithms →

* How write a recursive eq^n

→ 1 Back Substitution

→ 2 Recursion Tree Method

→ 3 Masters theorem

— Qu -

→ Recursive Function

n n 1

1) –

```
def fact(n):
    if n > 1:
        return c * fact(n-1)
    return 1 → c
```

Base Condition

$T(n)$

$T(n-1)$

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

① Back Substitution

problem

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

$T(n) = c + T(n-1)$ —— ①

$T(n-1) = c + T(n-2)$ —— ②

$T(n-2) = c + T(n-3)$ —— ③

$T(n-3) = c + T(n-4)$ —— ④

② in ①

$T(n) = c + c + T(n-2) \Rightarrow 2c + T(n-2)$

③ in

$3c + T(n-3)$

③ in

$T(n) = c + c + c + T(n-3) \Rightarrow 3c + T(n-3)$

$$\boxed{T(n) = k \cdot c + T(n-k)}$$

we know $T(1) = c$

so $(n-k) = 1$

$k = n-1$

$(n) - (n \quad c + T(n-n+1)$

$T(n) = (n-1) \cdot c + T(1)$

$T(n) = (n-1) \cdot c + c$

$$\boxed{T(n) = O(n)}$$

→ Time Complexity for Recursive functions

```
def fact(n):
    if n>1: → c
        return n*fact(n-1)
```

$T(n)$

$T(n-1)$

$f(n) \rightarrow T(n)$

$f(n-1) \rightarrow T(n-1)$

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

return $n * \boxed{fact(n-1)}$ $\Rightarrow$ $T(n-1)$ if $n > 1$

$T(n-1)$

return 1

$$\boxed{T(n) = c + T(n-1)} \quad \cdots 1)$$

① Back substitution method

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

$\quad\quad\quad\; = c + T(n-2) \rightarrow ②$

$T(n-2) = c + T(n-3) \rightarrow ③$

$\cdots \quad\quad\quad \cdots$

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

② in this eqn

$$T(n) = c + c + T(n-2) \Rightarrow 2c + T(n-2)$$

③ in this eqn

$$T(n) = 2c + c + T(n-3) \Rightarrow 3c + T(n-3)$$

$T(n) = 4c + T(n-4)$

After k callings we will get $T(1)$

$$\boxed{T(n) = kc + T(n-(k+1))} \rightarrow (5)$$

if $n=1$ then $T(1) = c$

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

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

$$\boxed{k = n-2}$$

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

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

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

$\rightarrow T(1) \Rightarrow c$

Replace k in eq^n (5)

$$\boxed{k = n-2}$$

$$T(n) = k \cdot c + T(n-(k+1))$$

$$T(n) = (n-2) \cdot c + T(n-(n-2+1))$$

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

$- \; n) = nc \;\; 2c + Tc$

$\cancel{l(n) = nc - 2c\, 1}$

$$\boxed{T\;n) - O(\;)}$$

- _) -   - - n) ; _ > 1

- C ; n = 1

```
def func(n):
    if n > 1:
n [   for i=1 to n:
          print(i)
T(n-1) -> fun(n-1)
C -> return 1
```

$$T(n) = n + T(n-1) \rightarrow ①$$

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

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

eqn ② subst in eqn ①

$$T(n) = n + (n-1) + T(n-2) \;\cdots\cdots ④$$

eqn ③ substn in eqn ④

$$T(n) = n + (n-1) + (n-2) + T(n-3) \cdot ⑤$$

$n \quad 1 \quad) \; (\quad n \;) \; 1(\underline{\; - 1 \;}) \rightarrow 6$

we know $T(n) = C$ if $n = 1$

$$T(\underline{n-(k+1)}) = T(1)$$

$$n - (k+1) = 1 \Rightarrow \boxed{k = n-2} \rightarrow 7$$

Substitute value of k from eq^n 7 to eq^n 6 we get

$$T(n) = n + (n-1) + (n-2) + (n-3) + \cdots + (n-k) + T(n-(k+1))$$

$$T(n) = n + (n-1) + (n-2) + (n-3) + \cdots + (\cancel{n} - (\cancel{n}-2)) + T(n-(n-2+1))$$

$$T(n) = \underbrace{n + (n-1) + (n-2) + (n-3) + \cdots + 2}_{} + \overset{C}{\cancel{T(1)}}$$

$$T(n) = \frac{n \cdot (n+1)}{2} + C$$

$$\boxed{T(n) = O(n^2)}$$

---

def $f(n)$:

$f(n) = T(n)$

$f(n/2) = T(n/2)$

deT | n) :

$T(n) = T(n)$

$-(n/2) = T(n/2)$

$- n > 1:$

return $\underbrace{f(n/2)}_{v_1} + \underbrace{f(n/2)}_{v_2}$

return 1

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

$$T(n) = c + 2T(n/2) \text{ --- } (1)$$

) - - | n

(n

n

$T = - 2 1 \quad (2) \to (1) \Rightarrow \boxed{2^0 c + 2^1 T\left(\frac{n}{2^1}\right)}$

substitute value of $T(n/2)$ from eq (2) to eqn (1)

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

$$T(n) = c + 2c + 4T(n/4) \Rightarrow \boxed{2^0 c + 2^1 c + 2^2 T\left(\frac{n}{2^2}\right)}$$

substitute value of $T(n/4)$ from eqn (3) to eqn (4)

$$T(n) = c + 2c + 4 \cdot c + 2 \cdot T(n/8)$$

$$= c + 2c + 4c + 8T(n/8) \Rightarrow \boxed{2^0 c + 2^1 c + 2^2 c + 2^3 T\left(\frac{n}{2^3}\right)}$$

$$\boxed{T\left(\frac{n}{8}\right) = C + 2T\left(\frac{n}{16}\right)}$$

$$= c + 2c + 4c + 8\cdot\left(c + 2T\left(\frac{n}{16}\right)\right)$$

$$= c + 2c + 4c + 8c + 16T\left(\frac{n}{16}\right)$$

$$\Rightarrow 2^0c + 2^1c + 2c + {}^3 + 2^4 - \left(\frac{n}{2^4}\right)$$

et a we wi call 1 1 times to re n- or 1)

then general e wi be $2^{k-}c$ < $\frac{n}{k}$

$$> 2c - + {}^2c + 2^3c + \cdots + 2c$$

$- 1\left(\frac{}{2^k}\right) = T(1)$ then $\frac{n}{2^k} = 1 \Rightarrow k = \log(n$

eq^n will be $\rightarrow c[2^0 + 2^1 + 2^2 + 2^3 + \cdots \cdots 2^{k-1}] + C$ $2^{k-1}] + T(1)$

$$\Rightarrow c\,[\underbrace{1+2+4+8+16+\cdots\cdots 2^{k-1}}_{(2n-1)}] + C$$

so $T(n) = c\cdot(2n-1) + C =$

Assignment $\rightarrow$

$$\Rightarrow \boxed{T(n) = O(n)}$$

```
def fab(n):
    if n == 1:
        return 0
    if n == 2:
        return 1
    return fab(n-1) + fab(n-2)
```

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

$$T(n) - \begin{cases} c & \text{if } n = 1 \text{ or } n = 2 \\ c + T(n-1) + T(n-2) \end{cases}$$

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

Assume T 1 n-)

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

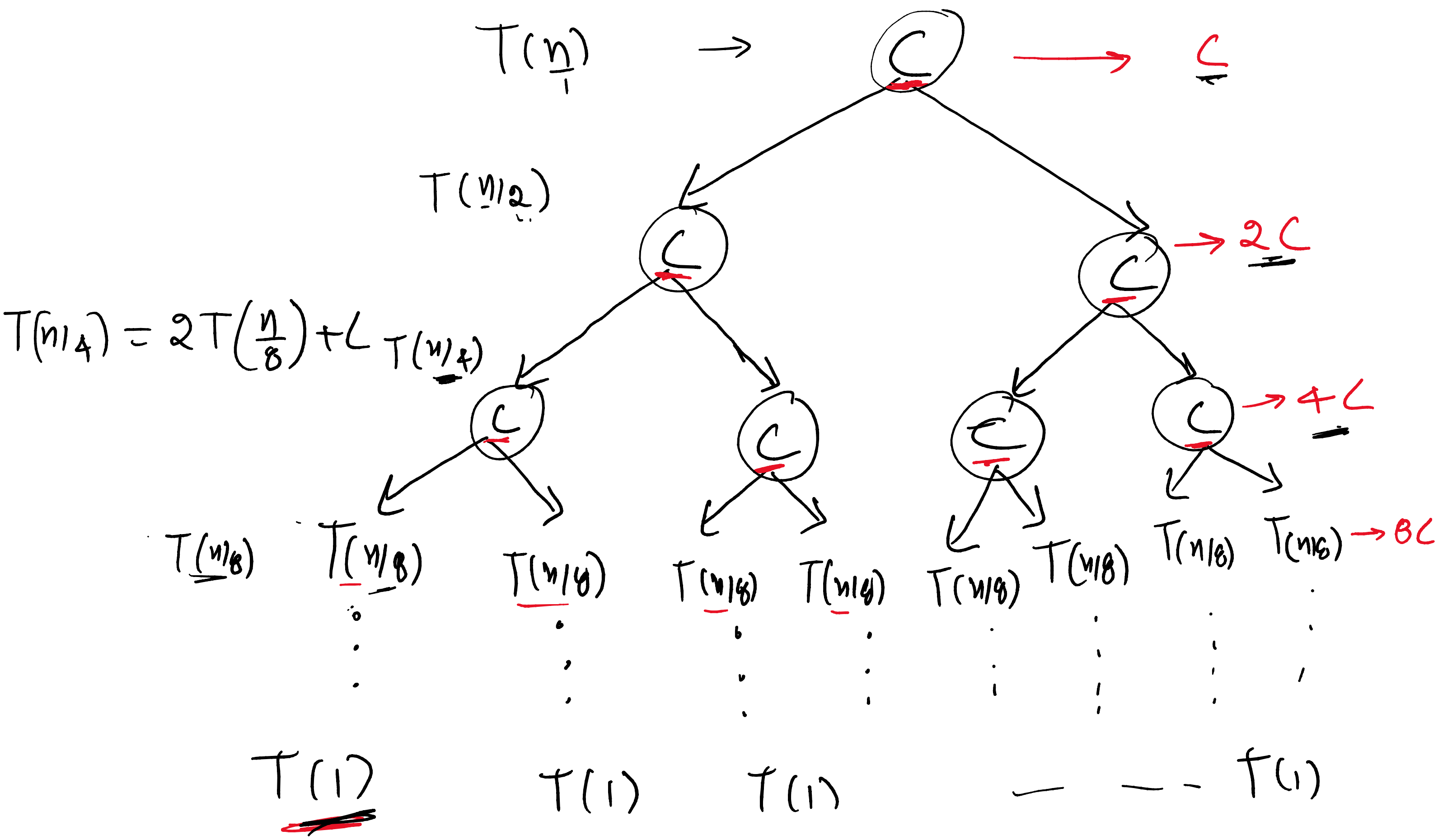
$$\boxed{T(n) = c + 2T(n-2)} \rightarrow$$

② Recursion Tree method

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

$$T(n) = 2T(n/2) + c$$

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



$$T\left(\frac{n}{n}\right) = T(1)$$

$$\boxed{n = 2^k}$$

$$T\left(\frac{n}{2^k}\right) = T(1)$$

$c + 2c + 4c + 8c + \ldots\ldots + nc$

$c + 2c + 4c + 8c + \ldots\ldots + 2^k \cdot c$

$c\,(1 + 2 + 4 + 8 + \ldots\ldots + 2^k) \rightarrow$ G.P

$$\frac{a\cdot(r^n - 1)}{r-1} \quad ; \; a = 1,\; r = 2,\; n = k$$

$$1\cdot\frac{(2^k - 1)}{1} \rightarrow 2^k - 1 \rightarrow 2^{\log(n)}$$

$n = 2^k \rightarrow k = \log(n) \longrightarrow \quad \Rightarrow n \rightarrow O(n)$

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

$\qquad\qquad c \qquad\qquad ; \; n = 1$

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

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

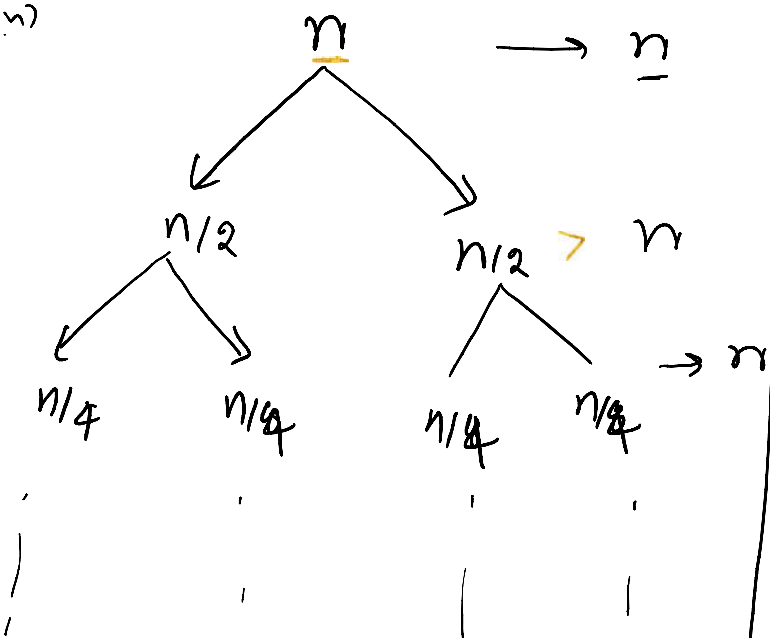
$T(n/4) = 2T(\frac{n}{8}) + \frac{n}{4}$

$T(\frac{n}{2^0}) \quad T(n)$

$T(\frac{n}{2^1}) \quad T(n/2)$

$T(\frac{n}{2^2}) \quad T(n/4)$

$T(\frac{n}{2^3}) \quad T(n/8)$



$\rightarrow k \cdot n$

$T(n) = k \cdot n$

$= \log(n) \cdot n$

$$\boxed{T(n) = n \cdot \log(n)}$$

$T\left(\frac{n}{2^k}\right) \; T\left(\frac{n}{n}\right) \quad T(1) \qquad T(1) \qquad (1)$

$$T\left(\frac{n}{2^k}\right) = T(1)$$

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

$$\boxed{k = \log(n)}$$

→ Master's Theorem

$(\log n)^2 \qquad \log^2 n \to \log\log(n)$

$\log^0 n \to 1$

$$\boxed{T(n) = a \cdot T\left(\frac{n}{b}\right) + O\left(n^k \log^p n\right)}$$

$a \geq 1$, $b \geq 1$, $k \geq 0$ and $p \in \mathbb{R}$

1. if $a > b^k$, Then $T(n) = O\left(n^{\log_b a}\right)$

2. if $a = b^k$

   a. if $p > -1$, Then $T(n) = O\left(n^{\log_b a} \cdot \log^{p+1} n\right)$

   Then $T(n) = O\left(n^{\log_b a} \cdot \log\log(n)\right)$

ⓑ if $p = -1$ Then $T(n) = O(n^{\log_b a} \cdot \log\log(n))$

ⓒ if $p < -1$ Then $T(n) = O(n^{\log_b a})$

③ if $a < b^k$

a - p>, e n (k n)

O <, h 1(- O k)

- $T(n) = aT(\frac{n}{b}) + O(n^k \log^p n)$
- $a \geq 1, b \geq 1, k \geq 0, p \in R$

$\boxed{2T(n/2)+c}$

① $T(n) = 2T(n/2) + n^0 \log^0 n$

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

$a \quad b^k$

$2 \quad 2^0$

$2 > 1 \rightarrow a > b^k$

$T(n) = O(n^{\log_b a})$

$= O(n^{\log_2 2})$

$T(n) = O(n$

$\boxed{1 \quad) - \quad (\,)}$

(2) $T(n) - 2T(\frac{n}{2} + n$

$T(n) = a \cdot T(\frac{n}{b}) + O(n^k \log^p(n))$

$a, b \geq 1,\ k \geq 0,\ p \in R$

$T(n) = 2 \cdot T(\frac{n}{2}) + O(n^1 \log^0 n)$

$a = \quad = 2,\ k - 1,\ p = 0$

$a \quad b$

$2 \quad > T(n) = O(n^{\log_b a} \cdot \log^{p+1} n)$

$T(n) = O(n^{\log_2 2} \cdot \log^{0+1} n)$

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

(3) $T(n) = \underbrace{2^n}_{a} T(\underbrace{\frac{n}{2}}_{b}) + n^n$

a is not constant so we can not apply master then

(4) $T(n) = 16\, T(\frac{n}{4}) + n^1 \log^0(n)$

$a = 16,\ b = 4,\ k = 1,\ p = 0$

$a > b^k \quad \rightarrow \quad T(n) = O(n^{\log_b a})$

$a > b^k$ $\quad >$ $\quad T(n = O\left(n^{\log_b a}\right)$

$6 \quad 4^1$

$b = 4$

$a = 16$

$4^{\underline{y}} = 16$

$\boxed{y = 2}$

$\log_b a = y$

$\boxed{b^y = a}$

$T(n) = O\left(n^{\log_{\underline{4}} 16}\right)$

$\boxed{T(n) = O(n^2)}$

⑤ $T(n) = 2T\left(\frac{n}{2}\right) + \frac{n}{\log(n)}$

$T(n) \quad 2T \; \frac{n}{2}) + n^1 \quad {}^1 n$

$a - 2, \; b - 2, \; k = \;\;, \; p = -1$

$a = b^k$

$2 = 2^1$

$T(n) = O\left(n^{\log_b a} \cdot \log\log(n)\right)$

$T(n) = O\left(n^{\log_2 2} \cdot \log\log(n)\right)$

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

$p > \underline{-1}$

$p = -$

$p \cancel{\leq} -1$

⑥ $T(n) = \underline{0.5}\, T\left(\frac{n}{2}\right) + \frac{1}{n}$

$a \geq 1 \qquad b \geq 1 \qquad k \geq 0$

$= 0 \qquad - \quad , \quad < \quad -$

F $\qquad$ F

F F

we can not apply masters theorem

⑦ $T(n) = 7T\left(\frac{n}{3}\right) + n^2$

$a=7 \quad b=3 \quad k=2 \quad p=0$

$a < b^k$

$7 < 3^2$

$T(n) = O\left(n^k \log^p n\right)$

$T(n) = O\left(n^2 \cdot \log^0 (n)\right)$

$T(n) = O(n^2)$