

$$\begin{array}{c}
 \textcircled{\#} \quad \textcircled{1} \text{ omega } \\
 \downarrow \\
 \Sigma \\
 \hline
 \mathcal{O}(n^0) \quad \overbrace{\dots}^{\mathcal{O}(n^1)} \quad \mathcal{O}(n^1)
 \end{array}
 \quad \text{Big(OH)} \quad \mathcal{O}(1) \rightarrow \text{worst case}$$

# 2 How we do calculate the  $\mathcal{T}(C)$ ?

- ① Analyse the code.
- ② Assume that each instruction consumes a one unit of time
- ③ Calculate the time taken by each instruction and collectively frame the expression
- ④ You will get a equation
- ⑤ ignore the constant and choose a higher degree function.

$$\begin{array}{c}
 x^2 + x + 3 \\
 \hline
 \mathcal{O}(n^2)
 \end{array}$$

$n = 5$

$n = 10$

$n = 100$

$100^{20}$

Best

$O(1)$   
case  $O(1)$

void fun1 (int  $n$ ) {  
 ↓  
 int  $a = 10$  → ①  
 ↓  
 int  $b = 20$  → ②  
 ↓  
 int  $sum = a + b + n$  → ③  
 ↓  
 int  $pro = a * b * n$  → ④  
 ↓  
 $S \times O(Sum + " " + pro)$  → ⑤  
 ↓  
 3

fun  
 $1 + 1 + 3 + 3 + 1$

9  
 $O(9)$

$O(9)$

$O(1)$

$O(9)$  constant

result change?

$O(1)$   
 $O(200)$

$O(1)$

Ex-2

Void sum\_2 (int n) {

for (int i = 0; i <= n; i++) {

    sum\_0 (i);

    sum\_0 (i);

    3

    1 + (n+2) + n + 1 + n

    1 + 2 + 1 + n + 3 + 3

~~X + 3 n~~

$O(n)$

$n = \underline{\underline{5}}$      $n-1 = \underline{\underline{4}}$   
 $n+2 = \underline{\underline{7}}$      $n+1 = \underline{\underline{6}}$   
for (     $i = \underline{\underline{0}}$      $i = \underline{\underline{1}}$      $i = \underline{\underline{2}}$      $i = \underline{\underline{3}}$      $i = \underline{\underline{4}}$ )

$i = \underline{\underline{0}}$      $i = \underline{\underline{1}}$   
 $i = \underline{\underline{2}}$      $i = \underline{\underline{3}}$

$0 \leq 5$

$1 \leq 5$

$2 \leq 5$

$3 \leq 5$

$4 \leq 5$

$5 \leq 5$      $n$

~~X~~  $\leq 5$

$n + 1$

$s + 1 = \underline{\underline{6}}$

~~(Ans)~~  $n = \underline{\text{even}}$   $\therefore$   
 void f3(int n) {  
 $\rightarrow$  if ( $n \times 2 == 0$ ) {  $\circlearrowleft 1$   
 $\quad \text{System}(\text{"even"}) ;$   
 $\rightarrow$  }  
 $\text{else } \{$   
 $\quad \text{for (int } i=0; i < n; i++) \{$   
 $\quad \quad \text{System}(i);$   
 $\}$   
 $\}$   
 $\therefore$   $\underline{\text{Best}} \rightarrow O(1)$   
 $\underline{\text{Average}}$   
 $\underline{\text{Worst}} \rightarrow O(n)$   
 $n = \frac{1}{n}$

$$S_{\text{eff}} = 1 + \sigma + 1 + \sigma + \sigma = \cancel{\sigma} + \cancel{\sigma} = \cancel{\sigma}$$

$\boxed{O(n)}$

Ex-4

```

void fun4 (int n) {
    i
    j
    for (int i=1; i<n; i++) {
        k
        for (int j=0; j<=i; j++) {
            l
            cout << i << " - " << j;
        }
    }
}

```

$n=4$

$i$        $j$   
 $k$        $l$   
 $z = j/2$   
 $z = j+2$

$n=2$

$(n)^2$

$(4)^2$

Q8

$(8)^2 = (64)$

$O(n^2)$

$O \leq 1$

$i \leq j$

$1 + 2 + 3 + 4 + \dots + n$

$\frac{n(n+1)}{2}$

$O(n^2)$

$\frac{n^2 + n}{2}$

$$m = 8 \quad n < 8$$

void sumS (int n) {

for (int i=0; i<n; i+=2)

$$\begin{array}{l} j = i \times 2 \\ \hline j = i/2 \end{array}$$

$$\begin{aligned} S &= 2^{k-1} \\ &= 2^{3-1} \\ &= 2^2 \end{aligned}$$

$$\boxed{k = 2^{k-1}}$$

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

$$2^k = n$$

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

$$\boxed{2 \times 2^2} = \textcircled{8}$$

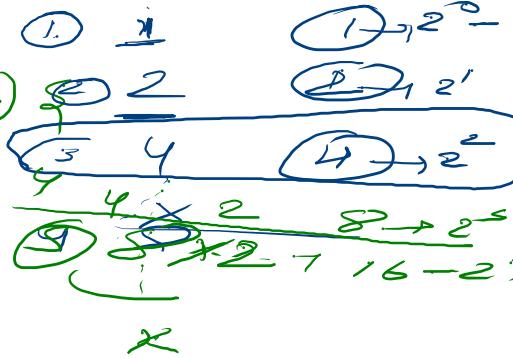
$$\log_2 8 \approx 3$$

Ans

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

(3) ...

$$\boxed{\Theta(\log n)}$$



$$\overbrace{n=10}^{\text{3 } \textcircled{1} \text{ } \delta < 10}$$

$$\log_2 = 10$$

$$3 \textcircled{1} \delta < 10$$

① Always

$$\begin{array}{l} i = i/2 \\ i = i - 2 \end{array} \quad \left. \right\}$$

$$\boxed{\log_2 4 \approx 2}$$

$$\overbrace{n=10}^{8 \times 2}$$

$$\cancel{\Theta(\log n)}$$

$$1 \leq \frac{5}{8}$$

$$\boxed{\log_2 2^4 \approx 4}$$

②

Binary Search ✓ ~~0(n)~~

$$\frac{\Theta(\log n)}{\cancel{\Theta(n)}}$$

$$\Leftrightarrow \Theta = 0$$

Ex-9 void sum() { int n=5 }

$$n = 20$$

```
goal (int i=n; i>0; i=i/2)
```

201

Ahmed

2012

S Siso (1)

S~~1~~

1

for divide  blocks true

27 10

24 16

2

12

2 / 4

218

12

1

1

$$\log_2 x = n$$

## On - is - es - k

$$\log_2 5 \approx 2.3$$

~~10~~ 5 10

SP-15

2182 — 12

~~Y~~ O.S. S

$$2^k = n$$

103

$$1 \leq k \leq n$$

142

$$\begin{array}{r} 20 = 10 \times 2^5 \\ \hline x = 10 \times 2 \end{array}$$

$$\underline{O(\log n)}$$

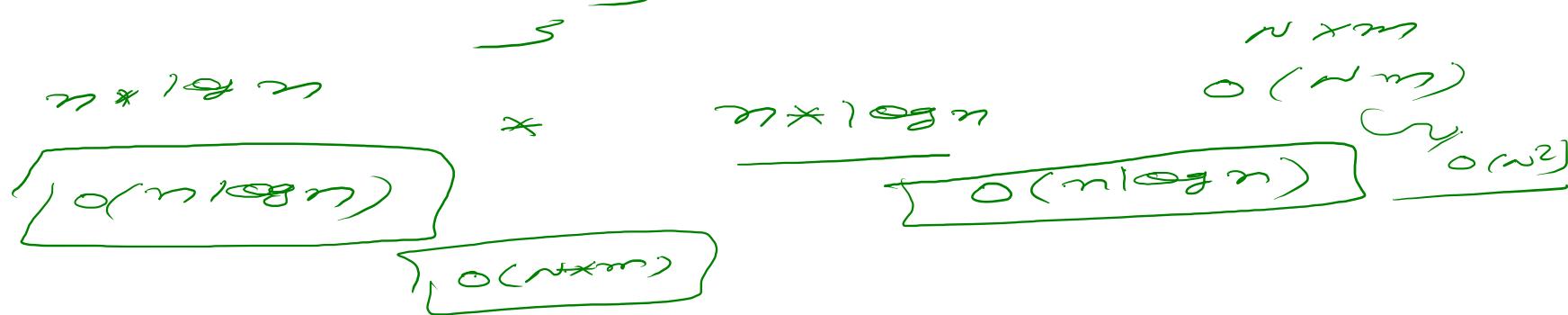
```

void fun8(int n) {
    for (int i=0; i<n; i++) { for (int j=i+1; j<n; j++) {
        cout << i << " " << j;
    }
}

```

Time:

For the inner loop, time complexity is  $O(n)$ .  
 For the outer loop, time complexity is  $O(n^2)$ .



void fun 3 (int i)

$n^{1.8n}$

for (int i=0; i<n; i++) {  
    sysol(i);  
    3 ← }

---

for (int i=0; i<n; i++) {  
    sysol(i);

$O(2^{0.8n} < O(n) < O(n^2) < O(n^2) \leq O(n^2) < O(n^2) \leq O(n^2)$

permutation  
Recursion

$$n+n = 2n$$

$= O(n)$