

Day - 2

Sunday, 21 December 2025 12:08 PM

$\text{Q} \quad \text{printArr}(\text{int arr}[], \text{int } n) \{$ [L \rightarrow R]



$\text{for } (i=0; i < n; i++) \{$

$\text{print}(arr[i])$

$\}$

$\Rightarrow TC = O(n)$
 $SC = O(1)$

$\text{Q} \quad \text{printArr}(\text{int arr}[], \text{int } n) \{$ [R \rightarrow L]

$\text{for } (i=n-1; i \geq 0; i--) \{$

$\boxed{\text{print}(arr[i])}$

$\}$

$TC = O(n)$
 $SC = O(1)$.

$\text{Q} \quad \text{printArr}(\text{int arr}[], \text{int } n) \{$

$\text{for } (i=1; i < n; i *= 2) \{$

$\text{print}(arr[i])$

$\}$

$TC = O(\log n)$
 $SC = O(1)$

$\text{Q} \quad \text{for } (i=n, i>1; i/2) \{$

$\text{print}(i)$

$$\begin{array}{ccccccc}
 i = & n & n/2 & n/4 & n/8 & \dots & n \\
 & \Downarrow & \Downarrow & \Downarrow & \Downarrow & \dots & \\
 & \frac{n}{2^0} & \frac{n}{2^1} & \frac{n}{2^2} & \frac{n}{2^3} & \dots & \left(\frac{n}{2^{k-1}} \right) > 1 \\
 & \text{1st Step} & \text{2nd Step} & & & \text{kth Step} & \leq 1 \\
 & & & & & & \underbrace{\dots}_{n/2^k}
 \end{array}$$

$$\begin{array}{c}
 \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \checkmark \\
 O(1) \quad O(1) \quad O(1) \quad O(1) \quad \frac{n}{2^k} \leq 1
 \end{array}$$

K-times

$$\Rightarrow K \cdot O(1) \Rightarrow \underline{\underline{O(K)}}$$

$T_C = \underline{\underline{O(\log_2 n)}}$

$$\begin{aligned}
 &\Rightarrow \boxed{2^k = n} \\
 &\Rightarrow \boxed{K = \log_2 n}
 \end{aligned}$$

$$SC = \underline{\underline{O(1)}} \quad \checkmark$$

\Leftrightarrow

$\text{for } (i=0, i < n, i++) \{$

$\Rightarrow \underline{\underline{n}} \cdot \text{times}$

$\text{for } (j=0, j < n, j++) \{$

$\text{print}(“\ell^i”);$

$\Rightarrow \underline{\underline{n}} \cdot \text{times} \Rightarrow \underline{\underline{O(n)}}$

$T_C = n * O(n) \Rightarrow \underline{\underline{O(n^2)}}$

\Leftrightarrow

$\text{for } (i=0, i < n, i++) \{$

$\text{for } (j=0, j < m, j++) \{$

$\text{print}(“\ell^i”);$

$T_C = \underline{\underline{O(n \cdot m)}}$
 $SC = \underline{\underline{O(1)}}$

\Leftrightarrow

$\text{for } (i=0, i < n, i++) \{ \Rightarrow n \cdot \text{times}$

$\text{for } (j=1, j < n, j * 2) \{$

$\text{print}(“\ell^i”);$

$T_C = \underline{\underline{O(\log_2 n) \cdot n}}$
 $\Rightarrow O(\log n)$

$$TC = O(n \log_2 n)$$

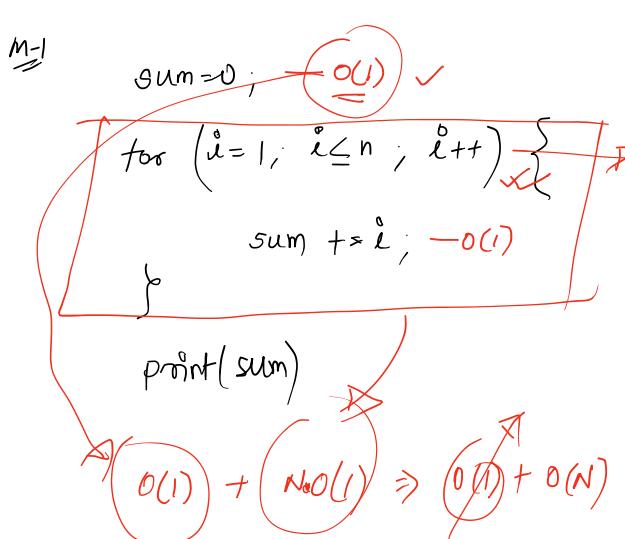
$$SC = O(1)$$

Given a no. "n".

Find the sum of 1st 'n' natural nos. $(1, 2, 3, \dots, n)$

e.g. $n=3 \Rightarrow (1, 2, 3) \quad n=5 \quad (1, 2, 3, 4, 5)$

~~0/P~~ $\Rightarrow (6) \times$ ~~0/P~~ $\Rightarrow (15) \times$



$i=1$	sum = 0
1	$0+1=1$
2	$1+2=3$
3	$3+3=6$ \checkmark
4	$6+4=10$
5	$10+5=15$ \checkmark

$$TC = O(N)$$

$$SC = O(1)$$

M-2

$1+2+3+\dots+n = (AP)$

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

$a, a+d, a+2d, \dots, a+(n-1)d$

$\frac{n}{2} (a + \underbrace{a+(n-1)d}_{last})$

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

$n=3$

$\sum = \frac{3^2}{2} = 6 \times$

~~sum = $\frac{9 \times 3}{2} = 13.5$~~



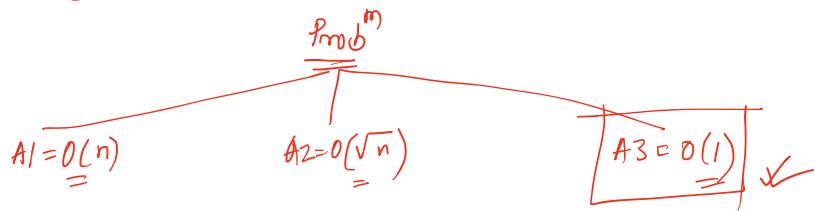
$sum = (n)(n+1)/2$

print(sum) $\Rightarrow O(1) \Rightarrow TC.$

sum ~ 2 - 100

SC = O(1)

Time Complexity



Q int i;
for ($i=0$; $i \leq n$; $i++$) {
 print(i) $\Rightarrow O(1)$
 }

 $K \cdot O(1) \Rightarrow O(K)$
 $\Rightarrow O(\sqrt{n}) \Rightarrow TC$

$i=0 \quad 0 \cdot 0 = 0 < n$
 $i=1 \quad 1 \cdot 1 = 1 < n \checkmark$
 $i=2 \quad 2 \cdot 2 = 4 < n$
 \vdots
 $i=K-1 \quad (K-1) \cdot (K-1) < n \times$
 $i=K \quad [(K) \cdot (K) \geq n] \times$

$K^2 \geq n$
 $K^2 = n$
 $K = \sqrt{n}$

Q for ($i=0$; $i \leq n$; $-$) {
 print("hi")
 }
 $i = i * 2;$

$i \geq 0 < n \checkmark$
 $i = 0 * 2 = 0 < n$
 \vdots
 $i = 0 * 2 = 0 < n \checkmark$

Q for ($i=1$; $i \leq n$; $i *= 2$) {
 print("hi")
 }

 $\boxed{\text{if work} \Rightarrow TC = O(n)}$
 $\boxed{\text{print("hi")}} \Rightarrow O(n)$

$TC = \log_2 n * O(n) \Rightarrow O(n \cdot \log_2 n)$
 $SC = O(1)$

for ($i=1$; $i \leq n$; $i = i * 3$) {
 } print("hi") $\Rightarrow O(\log_3 n)$

for ($i=0$; $i \leq 99$; $i++$) { "100" times.

for ($i=1$; $i \leq n$; $i++$) {
 } print("hi") $\Rightarrow O(n)$

$$TC \approx 100 \cdot O(n) \rightarrow O(100 \cdot n)$$

\circlearrowleft Drop constant. $\Rightarrow O(n)$
 $\Rightarrow O(n)$

$i=0$; $sum = 0$
 $K \text{ time} \Leftarrow$ for ($i=1$; $sum < n$; $i++$)
 } $sum \rightarrow sum + i \Rightarrow O(i)$ K steps

$$TC = K \cdot O(i) \Rightarrow O(K)$$

$$\Rightarrow O(\sqrt{n}) \checkmark$$

$$SC = O(1)$$

$i=0$	$sum=0$	
1	$0+1 = 1 < n \checkmark$	
2	$1+2 = 3 < n \checkmark$	
3	$3+3 = 6 < n \checkmark$	
4	$6+4 = 10 < n \checkmark$	
$(K-1)$	$+ (K-1) < n \checkmark$	
K	$\left[\frac{K \cdot (K+1)}{2} \right] \geq n \rightarrow \text{cond failed.}$	

$$\frac{K \cdot (K+1)}{2} \geq n \Rightarrow K^2 + K \geq 2n$$

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

$$\Rightarrow K + 1 = 2n/K$$

$$\Rightarrow K = \frac{2n}{K} - 1 \Rightarrow K = \frac{n}{K} \Rightarrow K^2 = n \Rightarrow K = \sqrt{n}$$

Rules of Big-O Notation

- If $f(n)$ is sum of several terms, then one with the largest growth rate (largest degree) will be kept and others can be removed.

$$TC = \cancel{N^3} + N^2 + N + \log n$$

X.

$$= O(N^3)$$

2. If $f(n)$ is product of several terms, any constants (that don't depends upon 'n') can be removed.

$$TC = \cancel{5} \cdot N^3 + \cancel{\frac{3n}{2}} + \cancel{2 \log n}$$

$$= O(N^3)$$

$$TC = \cancel{3} (\log n) \cdot N^3$$

$$\approx O(\log n \cdot N^3)$$

$$TC = n + \cancel{\frac{\log n}{n}} \Rightarrow O(n)$$

~~$n > \log n$~~

$$TC = \underline{n^2} + n \cdot \log n \Rightarrow O(n^2)$$

$$TC \sim \cancel{10^3} \cdot \cancel{2^n} \cdot \cancel{n^2} + n \log n$$

$\approx 2^{10}$

$$2^{\cancel{n^2}} + \underline{n \log n}$$

$$\Rightarrow O(2^n \cdot n^2) \times$$

Q) Compare the following:

1. $n! < n \log n$

2. $n > \log n$

3. $n^2 > \sqrt{n}$

$$4 \cdot n^2 > n \log n$$

$$5. \quad \cancel{n \log n} > \cancel{n \log(\log n)}$$

Dominance Relⁿ

8972411533 — Rahul