

Tutorial-2

Answer 1) → Void function (int n)

```
int j=1, i=0;
while (i < n)
{
    i = i+j;
    i++;
}
```

$$i = 1, \quad i = 0+1$$

$$i = 2, \quad i = 0+1+2$$

$$i = 3, \quad i = 0+1+2+3$$

Loop ends when $i \geq n$

$$0+1+2+3 \dots n > n$$

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

$$k^2 > n$$

$$k = \sqrt{n}$$

$$O(\sqrt{n}).$$

Answer 2) →

Recurrence Relation For Fibonacci Series:

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

$$T(0) = T(1) = 1.$$

⊙ If $T(n-1) \approx T(n-2)$

(Lower Bound)

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

$$= 2 \times 2T(n-4) = 4T(n-4)$$

$$= 4(2T(n-6))$$

$$= 8T(n-6)$$

$$= 8(2T(n-8))$$

$$= 16T(n-8)$$

$$T(n) = 2^k T(n-2k)$$

$$n - 2k = 0$$

$$n = 2k$$

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

$$T(n) = 2(2^{n/2})$$

$$T(n) = 2^{n/2} T(0) = 2^{n/2}$$

⊙ if $T(n-2) \cup T(n-1)$

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

$$= 2(2T(n-2)) = 4T(n-2)$$

$$= 4(2T(n-3)) = 8T(n-3)$$

$$= 2^k T(n-k)$$

$$n-k=0$$

$$\boxed{k=n}$$

$$T(n) = 2^k \times T(0) = 2^n$$

$$= T(n) = O(2^n) \quad (\text{Upper Bound})$$

Answer 3) → ⊙ $O(n \log n) \Rightarrow$ for (int i=0; i<n; i++)

$$\quad \swarrow$$

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

$$\quad \swarrow$$

$$\quad \parallel \text{ some } O(1)$$

⌋

⊙ $O(n^3) \Rightarrow$ for (int i=0; i<n; i++)

$$\quad \swarrow$$

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

$$\quad \swarrow$$

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

$$\quad \swarrow$$

$$\quad \parallel \text{ some } O(1)$$

⌋

⊙ $O(\log \log n) \Rightarrow$ for (int i=1; i<=n; i=i*2)

$$\quad \swarrow$$

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

$$\quad \swarrow$$

$$\quad \parallel \text{ some } O(1)$$

⌋

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

let's assume $T(n/2) \geq T(n/4)$

so, $T(n) = 2T(n/2) + cn^2$

Applying Master's Theorem $(T(n) = aT(\frac{n}{b}) + f(n))$

$a=2, b=2, f(n)=n^2$

$c = \log b^a = \log_2^2 = 1$

$n^c = n$

compare n^c and $f(n) = n^2$

$f(n) > n^c$

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

Answer 5) int fun (int n)

do for (int i = 1, i <= n; i++)

do for (int j = 1; j < n; j += i)

do // sum O(1)

do
do
do

i = 1 ——— i = 1
i = 2
i = 3
i = n ——— n times

i = 2 ——— j = 1
j = 3
j = 5
j = 7 ——— Loop ends when j > n
1 + 3 + 5 + 7 > n
 $k > \frac{n}{2}$
—— n times

So, Total Complexity =
= $O(n^2 + n^2 + n^2 \dots)$
= $O(n^2)$

i = 3 ——— j = 1
j = 4
j = 7 ——— 1 + 4 + 7 > n
 $k > \frac{n}{3}$

i = 4 ——— $k > \frac{n}{4}$

i = n.

Answer 6) \rightarrow for (int $i=2$; $i \leq n$; $i = \text{low}(i, k)$)

α // same (1)

\forall

complexity of $\text{low}(i, k) = O(\log N)$
 $= \log(k)$

$$i = 2$$

$$i = 2^k$$

$$i = 2k^2$$

$$i = 2k^3$$

$$i = 2k^4$$

$$i = 2k^n$$

$$i = 2k^M$$

loops ends when $i > n$

$$2k^M > n$$

$$\log(2k^M) > \log n$$

$$k^M \log 2 > \log n$$

$$k^M > \log n$$

$$\log(k^M) > \log(\log n)$$

$$M \log k > \log(\log n)$$

$$M > \frac{\log(\log n)}{\log(k)}$$

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

Answer 8) \rightarrow a) $100 < \log n < \sqrt{n} < n < \log(\log n) < n \log n$
 $< \log n! < n! < n^2 < \log^{2n} < 2^n < 2^{2n} < 4^n$.

b) $1 < \sqrt{\log n} < \log n < 2 \log n < \log 2^N < N < 2N < 4N < \log(\log N) < N \log N < \log N! < N! < N^2 < 2 \times 2^N$.

c) $96 < \log_8 N < \log_2 N < n \log_6 N < n \log_2 N < \log n!$
 $< N! < 5N < 8N^2 < 7N^3 < 8^{2n}$.