

Assignment 1

Q-1 Solve the following loops & calculate time complexity

y void function (int n) {

 int i, j, k, count = 0;

 for (i = n/2; i <= n; i++) {

 for (j = 1; j + n/2 <= n; j++) {

 for (k = 1; k <= n; k *= 2) {

 count++;

 }

(n logⁿ)² =

- Here, i-loop will run $n/2$ times but how?

j-loop will run $n/2$ times $i = n/2$?

$3(i+1) : n = 3(i+1) : n/2$

no. of times j-loop runs $i \rightarrow n - n/2 = n/2$

: ("x") times

and for j-loop $\rightarrow j + n/2 \leq n$: Xored

$j \leq n - n/2$

$j \leq n/2$

so it will run $n/2$ times

and for k-loop

Step K

0 2 = 2^0 for k-th step, it will be 2^n

1 $4 = 2^2$

2 8 = 2^3

Assuming $2^K \geq n$ ($0 = t(n)$, $i = n$)

$(i+1) : 10 = x_i * i : 1 = i$

$+ t(n)$

$$\text{So, } \log_2 k \geq \log n$$

$$\text{So, } k \geq \log n$$

So, total complexity $\rightarrow \frac{n}{2} \times \frac{n}{2} \times \log n$

$$\frac{n^2}{4} \log n$$

$$= O(n^2 \log n)$$

For asymptotic notation $\rightarrow \frac{n}{2} \times \frac{n}{2} \times \log n$

$$= \frac{n^2}{4} \log n$$

$$= O(n^2 \log n)$$

3) void function (int n) {

if (n == 1) return;

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

for (int j=1; j <= n; j++) {

printf ("K");

break;

}

- Here - i-loop will run n-times and inner loop will be run only 1 time in every iteration so,

$$T(n) = 2n \text{ for } n \geq 1$$

so, asymptotic time complexity = $O(n)$

so, asymptotic time complexity = $O(n)$

4) void function (int n) {

int i, count = 0;

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

count++

}

- Here let's see the event (iteration part)

Step 8 (int i, n; i = 1; i <= n; i++)

0 1 ; ("*") ; + n²

1 2

:

111

SUPPOSE, $K^2 \geq n$

$K > \sqrt{n}$

8 99+2

↳ So, the Asymptotic time complexity - $O(\sqrt{n})$

4) Function (int n) {

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

 for (int j = 1; j <= n; j *= 2) {

} }

- Here, the outer loop will be executed n times

: $O = n \times 2 = n^2$

↳ and for inner loop : $O = i^0 + i^1 + \dots + i^{n-1} = i^n$

$i = 2^0 + 2^1 + \dots + 2^{n-1} = 2^n - 1$

Step

j

$i = 2^0 + 2^1 + \dots + 2^{n-1} = 2^n - 1$

0 1 = 2^0

SUPPOSE $2^n - 1 \leq n^2$

1 3 ($2^0 + 2^1 = 2^1 \times 2^0$; $2^0 = K + i^0$)

2 4 = 2^2

$2^2 \geq n^2$

3 :

$K \geq \log n$

k 2^k

- So, final Asymptotic time complexity $O(n \log n)$

5) Function (int n) {

```
for (int i=1; i<=n; i++) {
```

```
    for (int j=1; j<=i; j++) {
```

```
        printf ("*");
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

- So, in that case, the outer-loop will run n -times and in each iteration the k -loop will run n loop.

- So, total iteration will be $n \times n$

- So, Asymptotic time complexity $\Theta(n \times n) = \Theta(n^2)$

Q-2) calculate order of complexity using master's theorem:

$$1) T(n) = 3T(n/2) + n^2$$

Comparing with master's theorem

$$a=3$$

$$k=2$$

$$b=2$$

$$P=0$$

- So, $a < b^k$ ($\because 3 < 2^2$)

according to case 3-A ($\therefore d < P$, 02)

$$T(n) = \Theta(n^k \log^P n)$$

$$= \Theta(n^2 \log^0 n)$$

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

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

Comparing with master's theorem

$$a=4, b=2, k=2, P=0$$

So, $a \geq b^k$ ($\because 4 \geq 2^2$)

according to case 2-A

$$\begin{aligned} T(n) &= O(n \log n \log^p n) \\ &= O(n \log_2 4 \log^p n) \\ &= O(n^2 \log n) \end{aligned}$$

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

- $a=1, b=2, k=2, p=0$

so, $a < b^k \quad (\because 1 < 2^2)$

$$\begin{aligned} T(n) &= O(n^2 \log^p n) \\ &= O(n^2 \log^0 n) \\ &= O(n^2) \end{aligned}$$

4) $T(n) = 16T(n/4) + n$

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

so, $a > b^k \quad (\because 16 > 4^1)$

$$\begin{aligned} T(n) &= O(n^{\log_b a}) \\ &= O(n^{\log_4 16}) \\ &= O(n^2) \end{aligned}$$

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

$a=2, b=2, k=1, p=1$

so, $a = b^k \quad (\because 2 = 2^1)$

$(\because a=b \quad \therefore p=0)$

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

$$\begin{aligned} T(n) &= O(n^{\log_2 4} \cdot \log^{p+1} n) + (\alpha)^r T \\ &= O(n^{\log_2 2} \log^{1+1} n) \\ &= O(n \log^2 n) \quad \alpha = 4, \beta = 1, p = d, r = 0 \end{aligned}$$

6) $T(n) = 2T(n/2) + n \log^{-1} n$ $\forall d > 0, \alpha$

$$\begin{aligned} a = 2, b = 2, k = 1, p = -1 \quad (\alpha^0 \text{pol } S_m) \alpha &= (\alpha)^r \\ (\alpha^0 \text{pol } S_m) \alpha &= \\ \text{so, } a = b^k \quad (\because 2 = 2^1) \quad (S_m) \alpha &= \end{aligned}$$

$$\begin{aligned} T(n) &= O(n^{\log_2 4} \log \log n) \quad (\alpha)^r T \quad (\alpha) \\ &= O(n^{\log_2 2} \log \log n) \\ &= O(n \log(\log n)) \quad \alpha = 2, \beta = d, r = 0 \end{aligned}$$

7) $T(n) = 2T(n/4) + n^{0.5}$ $(\beta < p \because) \quad \forall d < 0, \alpha$

$$\begin{aligned} a = 2, b = 4, k = \frac{5}{100}, p = 0 \quad (\beta < p \text{ pol } n) \alpha &= (\alpha)^r \\ (\beta < p \text{ pol } n) \alpha &= \\ \text{so, } a < b^k \quad (\because 2 < 4^{5/100}) \quad (S_m) \alpha &= \end{aligned}$$

$$\begin{aligned} T(n) &= O(n^{5/100} \log^0 n) \quad (\alpha^0 \text{pol } 1) \alpha + (\alpha)^r T \quad (\alpha) \\ &= O(n^{0.5}) \\ \alpha = 2, \beta = d, r = 0, \alpha &= 0 \end{aligned}$$

8) $T(n) = 6T(n/3) + n^2 \log n$
 $(n^2 \log n \because) \quad \forall d > 0, \alpha$

$$\begin{aligned} a = 6, b = 3, k = 2, p = 1 \quad (\alpha^q \text{pol } \alpha) \alpha &= (\alpha)^r \\ (\alpha^0 \text{pol } \alpha) \alpha &= \\ \text{so, } a < b^k \quad (\because 6 < 3^2) \quad (\alpha^r) \alpha &= \end{aligned}$$

$$\begin{aligned} T(n) &= O(n^2 \log^1 n) \\ &= O(n^2 \log n) \end{aligned}$$

g) $T(n) = 7T(n/3) + n^2 \log^3 n$ $\alpha = 7$
 $(\alpha < b^k \text{ or } \alpha = b^k \text{ and } m = 0)$
 $a = 7, b = 3, k = 2, p = 0$ $(\alpha > b^k \text{ or } \alpha = b^k \text{ and } m > 0)$

so, $a < b^k$ ($\because 7 < 3^2$) $\log^3 n + (\text{small terms}) Tn = (\alpha) T$ [2]

$$\begin{aligned} T(n) &= O(n^2 \log^3 n) \quad i=9, l=d, s=d, m=0 \\ &= O(n^2 \log^3 n) \\ &= O(n^2) \quad ('s=d' \Rightarrow d=0, m=0) \end{aligned}$$

10) $T(n) = 4T(n/2) + \log n$ $\alpha = 4$
 $(\alpha < b^k \text{ or } \alpha = b^k \text{ and } m = 0)$
 $a = 4, b = 2, k = 0, p = 1$ ($\alpha > b^k \text{ or } \alpha = b^k \text{ and } m > 0$)

so, $a > b^k$ ($\because 4 > 2^0$) $\log n + (\text{small terms}) Tn = (\alpha) T$ [F]

$$\begin{aligned} T(n) &= O(n^{\log_2 4}) \quad \alpha=4, \log_2 4 = 2, \mu=d, m=0 \\ &= O(n^2) \quad (\log_2 4 > s \therefore d > 0, m=0) \end{aligned}$$

11) $T(n) = 16T(n/4) + n! \log^{0.00112} n$ $\alpha = 16$
 $(12.0 \text{ or } 12.1 \text{ or } 12.2 \dots)$

$a = 16, b = 4, k = n, p = 0$

$\alpha \log n + (\text{small terms}) Tn = (\alpha) T$ [3]

so, $a < b^k$ ($\because 16 < 4^n$)

$i=9, s=k, \beta=d, \delta=0$

$$\begin{aligned} T(n) &= O(n^k \log^p n) \quad (\beta > 0 \therefore d > 0, m=0) \\ &= O(n^n \log^0 n) \\ &= O(n^n) \quad (\alpha > 1 \text{ or } \alpha = 1 \text{ and } m > 0) \end{aligned}$$

$(\alpha > 1 \text{ or } \alpha = 1 \text{ and } m > 0) \alpha = (\alpha) T$

$(\alpha > 1 \text{ or } \alpha = 1 \text{ and } m > 0) \alpha =$

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

$a = 3, b = 2, k = 1, P = 0 \quad O = 9, I = 4, S = d, m = 0$
 $('S = 5 \because') \quad \therefore d = 0, 02$

$\text{so, } a > b^k \quad (\because 3 > 2^1)$

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

$$= O(n \log^2 n)$$

$$(m^{1+0} \text{Pol}^0, m) O = (m)^I$$

$$(m^{1+0} \text{Pol}^1, m) O =$$

$$(m \text{Pol}^1, m) O =$$

13) $T(n) = 3T(n/4) + n \log n$ alpha 9.9 dt 2.809 m/s E-R
89+1098C

$a = 3, b = 4, k = 1, P = 1$

$\text{so, } a < b^k \quad (\because 3 < 4^1)$

$$T(n) = O(n^k \log^P n)$$

$$= O(n^1 \log^1 n)$$

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

$$m \text{ Pol } \alpha = (m)^I$$

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

$$m \text{ Pol } \alpha = (m)^I, 02$$

$$S(m \text{ Pol}) \quad S = (m)^I$$

$a = 3, b = 3, k = 1, P = 0$

$\text{so, } a < b^k \quad (\because 3 < 4^1)$

$$T(n) = O(n^{10g_b a} \log^{P+1} n)$$

$$= O(n^{10g_3 3} \log^{0+1} n)$$

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

$$m \text{ Pol } S = (m)^I, 02$$

$$\bar{m} \alpha = (m)^I$$

$$m \text{ Pol } S = (m)^I$$

$$\bar{m} \alpha = (m)^I \quad m = (m)^I, 02$$

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

$$\alpha + (\beta T(n)) \neq \epsilon - (\gamma T(n))$$

$$a=2, b=2, k=1, p=0 \quad \alpha=9, \beta=d, \gamma=d, \epsilon=0$$

$$\text{so, } u = b^k \quad (\because 2=2^1)$$

(use E:) $\beta d < 0, \alpha$

$$\begin{aligned} T(n) &= O(n^{10g_2 2}) \log^{p+1} n \\ &= O(n^{10g_2 2}) \log^{0+1} n \quad (\alpha > 0, \beta = 0) \\ &= O(n \log n) \quad (\beta = 0, \alpha = 0) \end{aligned}$$

Q-3) compare the fun^{ns}s and determines which is greater

$$\alpha=9, \beta=d, \gamma=d, \epsilon=0$$

1) $f(n) = n \log n$

$$g(n) = n \log n$$

(H > E:) $\beta d > 0, \alpha$

$$f(n) = n \log n$$

$$g(n) = n \log n \quad (\alpha > 0, \beta = 0)$$

$$g(n) = n \log n = 2^{\frac{(\log n)^2}{(\alpha \log n)}} \quad (\alpha > 0, \beta = 0)$$

$$\text{so, } f(n) = n \log n$$

$$g(n) = 2^{(\log n)^2}$$

$$\alpha + (\beta T(n)) \neq \epsilon - (\gamma T(n))$$

$$\therefore A \quad f(n) < g(n)$$

2) $f(n) = 2^{\log n}$

$$g(n) = n^{\sqrt{n}}$$

$$(\alpha > 0, \beta = 0) \quad \alpha = (\alpha)T$$

$$f(n) = 2^{\log n}$$

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

$$\text{so, } f(n) = n$$

$$g(n) = n^{\sqrt{n}}$$

$$n^{\sqrt{n}} = e^{\sqrt{n} \ln n}$$

$$n = 1^{\log n} (n-m) T \times S = (m)T$$

$$(s-m) T \times S = (1-m)T$$

$$\text{so, } f(n) = n \ll g(n) = n^{\sqrt{n}} (m) T \times S = (s-m) T$$

$$g(n) \gg f(n)$$

$$(m) T \times S \dots s \cdot s \cdot s = (m)T$$

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

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

$$\therefore T = (m)T \quad g(n)^2$$

$$m^2 = (m)T$$

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

$$(m^2) : r + i \in O(19) \text{ mod } 3m \Rightarrow T$$

$$\text{so, } g(n) = (f(n))^2$$

$$(m^2) + 2mT \text{ b/w } T$$

$$g(n) \gg f(n)$$

$$(1 < m) \quad ?$$

a-4) Determine time complexity for

$$= ("T D I") \quad T + m i d q$$

$$\Downarrow T(n) = \begin{cases} T(n-1) + n, n > 0 \\ 1, n = 0 \end{cases} \quad ; (SIT) + 2mT$$

$$T(n) = T(n-1) = T(n-2) = \dots = T(0)$$

$$(1 < m < 2m), (1)0 + (sm)T = (m)T \leftarrow$$

$$T(n) = O(n) \leq m < 2m, (1)0 = (m)T$$

b) void test (int n)

at Priyanshu

{

if (n > 0)

{

test+(n-1);

test+(n-1);

}

}

n <

sm <

m <

!

=> m

$$\begin{aligned}
 T(n) &= 2 * T(n-1) & 1 = m & n = m + 1 \\
 T(n-1) &= 2 * T(n-2) \\
 T(n-2) &= 2 * T(n-3) = (\alpha)^2 \times m = (\alpha)^2 \cdot m \\
 &\vdots \\
 \text{So,} & & (\alpha)^2 < (\alpha)^1 \\
 T(n) &= 2 \cdot 2 \cdot 2 \dots 2T^m \\
 \text{Since } T(0) &= 1; & m_1 = (\alpha)^2 & m_2 = (\alpha)^1 \\
 T(n) &= 2^n & &
 \end{aligned}$$

↳ Time complexity: $O(2^n)$

3) void Test(int n)

\$

if ($n > 1$)

\$

printf ("ICT");

Test (n/2);

\$

\$

$$(O)T = \dots = (s-m)T = (1-m)T = (\alpha)T$$

$$\rightarrow T(n) = T(n/2) + O(1), \text{ For } n > 1$$

$$T(n) = O(1), \text{ For } n \leq r_m \quad O = (\alpha)T$$

Solving the

$$(m+m^2) + 29T \text{ by } \frac{1}{2}$$

$\rightarrow n$

$\rightarrow n/2$

$\rightarrow n/4$

\vdots

$n \leftarrow 1$

$$(\alpha m)^2 + 29T$$

$$(\alpha m)^4 + 29T$$

$$(\alpha m)^6 + 29T$$

g

C

→ Time complexity $O(\log n)$

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

- The function divides the problem with 2 subproblem of size $n/2$

number of levels is $\log_2 n$

$$T(n) = n \log_2 n$$

Time complexity: $O(n \log n)$