CS & IT ENGINEERING

Algorithms

Analysis of Algorithms

Lecture No.- 07

By- Aditya Jain sir

Topics to be Covered

Topic

Topic Analysis of Recursive Algo

Topic

About Aditya Jain sir

1. Appeared for GATE during BTech and secured AIR 60 in GATE in very first attempt - City topper
2. Represented college as the first Google DSC Ambassador.
3. The only student from the batch to secure an internship at Amazon. (9+ CGPA)
4. Had offer from IIT Bombay and IISc Bangalore to join the Masters program
5. Joined IIT Bombay for my 2 year Masters program, specialization in Data Science
6. Published multiple research papers in well known conferences along with the team
7. Received the prestigious excellence in Research award from IIT Bombay for my Masters thesis
8. Completed my Masters with an overall GPA of 9.36/10
9. Joined Dream11 as a Data Scientist
10. Have mentored 12,000+ students & working professions in field of Data Science and Analytics
11. Have been mentoring & teaching GATE aspirants to secure a great rank in limited time
12. Have got around 27.5K followers on Linkedin where I share my insights and guide students and professionals.

Telegram Link for Aditya Jain sir: https://t.me/AdityaSir_PW

TTTe

Topic : Analysis of Algorithms

```
2.      Algo AJ(n)
        {
                if (n ==1)
T(n)                    return 1;           b
                else
                        return ((AJ(n-1) + AJ (n-1));
        }                         T(n-1)      T(n-1)
```

$$T(n) = 2T(n-1) + a, \; n > 1$$

$$T(n) = b, \; n = 1$$

$$T(n) = 2T\underline{(n-1)} + a \; - \textcircled{1}$$

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

$$T(n) = 2^{2}T(n-2) + 3a - \textcircled{2}$$

$$T(n) = 2^{3}T(n-3) + 7a - \textcircled{3}$$

$$\vdots$$

$$T(n) = 2^{K}T(n-K) + (2^{K}-1) * a - \textcircled{4}$$

For B.C, $n-k=1$

$$\boxed{k=n-1}$$

$$T(n)=2^{(n-1)}T(1)+\left(2^{n-1}-1\right)*a$$

$$\boxed{T(n)=\frac{2^n}{2}*b+\left(\frac{2^n}{2}-1\right)*a}$$

S3: TC: $\underline{O(2^n)}$

Topic : Analysis of Algorithms

#Q.
```
Algo AJ(n)
{
    if (n==2)
        return 2
    return (AJ($\sqrt{n}$) + AJ ($\sqrt{n}$))
}
```

A O(n)

B $O(\sqrt{n})$

[✓ C] O(logn)

D O(log(logn))

$$T(n) = b, \; n = 2$$

$$T(n) = 2T(\sqrt{n}) + a, \; n > 2$$

$$T(n) = 2T\underline{(n^{1/2})} + a \; - \textcircled{1}$$

$$T(n^{1/2}) = 2T(n^{1/2^2}) + a$$

$$T(n) = 2^2 T(n^{1/2^2}) + 3a \; - \textcircled{2}$$

$$T(n) = 2^3 T(n^{1/2^3}) + 7a \; - \textcircled{3}$$

$$T(n) = 2^k T(n^{1/2^k}) + (2^k - 1) * a$$

$$n^{1/2^k} = 2$$

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

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

$$T(n) = \log n * b + (\log n - 1) a$$

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

Topic : Analysis of Algorithms

#Q. Algo AJ(n) , Given AJ2(n) taking O(1) time

```
{
  if (n==1)
  return 2;
  else
  return (AJ1(n/2) + AJ1(n/2) + AJ2(n));
  }
}
```

↓

C

- **A** $O(\sqrt{n})$
- **B** O(n)
- **C** $O(n^2)$
- **D** O(logn)

$T(n) = b, \; n = 1$

$T(n) = 2T(n/2) + a, \; n > 1$

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

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

$T(n) = 2^3 T(n/2^3) + 7a$

$T(n) = 2^k T(n/2^k) + (2^k - 1)a$

$n/2^k = 1 \quad \boxed{2^k = n}$

$T(n) = n\,T(1) + (n-1) * a$

$T(n) = (b * n + a * n - a)$

$\boxed{TC : O(n)}$

Topic : Analysis of Algorithms

#Q. Algo AJ(n) , Given AJ2(n) taking O(n) time

```
{
  if (n ==1)
  return 1
  else
  {
          AJ (n/2)
          AJ (n/2)
          AJ 2 (n)
  }
}
```

- A O(n)
- [✓ B] O(n logn)
- C O(logn)
- D O(n^2 logn)

$T(n) = b, \ n = 1$

$T(n) = 2T(n/2) + n + a, \ n > 1$

$T(n) = 2T(n/2) + n, \ n > 1$

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

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

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

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

$T(n) = 2^k T(n/2^k) + k * n$

$$\boxed{\begin{aligned} n/2^k &= 1 \\ 2^k &= n \\ k &= \log(n) \end{aligned}}$$

$T(n) = n * T(1) + n * \log n$

$$\boxed{T(n) = n * b + n\log n}$$

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

Topic : Analysis of Algorithms

#Q. Given a Recurrence relation find out time complexity

$$T(n) = 2 \quad , \quad n = 2$$

$$T(n) = \sqrt{n} * T(\sqrt{n}) + n \ , n > 2$$

A O(nlogn)

B O(nlog (logn))

C $O(n^2)$

D O(log (logn))

$$T(n) = n^{1/2}\,\underline{T\left(n^{1/2}\right)} + n$$

$$T\left(n^{1/2}\right) = n^{1/2^2}\,T\left(n^{1/2^2}\right) + n^{1/2}$$

$$T(n) = \underline{n^{1/2}}\left[n^{1/2^2}\,T\left(n^{1/2^2}\right) + n^{1/2}\right] + n$$

$$T(n) = n^{\left(1/2 + 1/2^2\right)}\,T\left(n^{1/2^2}\right) + 2n$$

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

$$\left.\begin{array}{l} a = \frac{1}{2} \\ r = \frac{1}{2} \\ n = k \end{array}\right] \quad \frac{1}{2}+\frac{1}{2^2}\cdots\frac{1}{2^k} = \frac{a(1-r^n)}{1-r} = \frac{\cancel{\frac{1}{2}}\left(1-\frac{1}{2^k}\right)}{\cancel{1-\frac{1}{2}}} = \left(1-\frac{1}{2^k}\right)$$

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

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

$$\frac{1}{2^k}\log n = 1 \Rightarrow 2^k = \log n \Rightarrow \underline{K = \log(\log n)}$$

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

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

$$T(n) = \frac{n}{\cancel{2}} * \cancel{T(2)} + \log(\log n)*n$$

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

Topic : Analysis of Algorithms

(Q)

```
Algo AJ( n)
{
        if (n ==1)
            return 1
        else
            return [AJ($\sqrt{n}$) + 10]
}
```

$$T(n) = T(n^{1/2}) + a \quad - \textcircled{1}$$

$$T(n^{1/2}) = T(n^{1/2^2}) + a$$

$$T(n) = T(n^{1/2^2}) + 2a$$

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

$$\vdots$$

$$T(n) = T(n^{1/2^k}) + k * a$$

$$n^{1/2^k} = 2$$

$$\frac{1}{2^k} \log n = 1 \Rightarrow K = \underline{\log(\log n)}$$

$$T(n) = b + \log(\log n) * a$$

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

Topic : Analysis of Algorithms

```
Algo AJ( n)
{
  if (n ==1)
            return n
  else
            return (AJ (n/2) + 10)
}
```

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

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

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

$$\vdots$$

$$T(n) = T(n/2^k) + k * a$$

$$\boxed{\begin{array}{c} \dfrac{n}{2^k} = 1 \\ 2^k = n \\ k = \log n \end{array}}$$

$$T(n) = T(1) + \log n * a$$

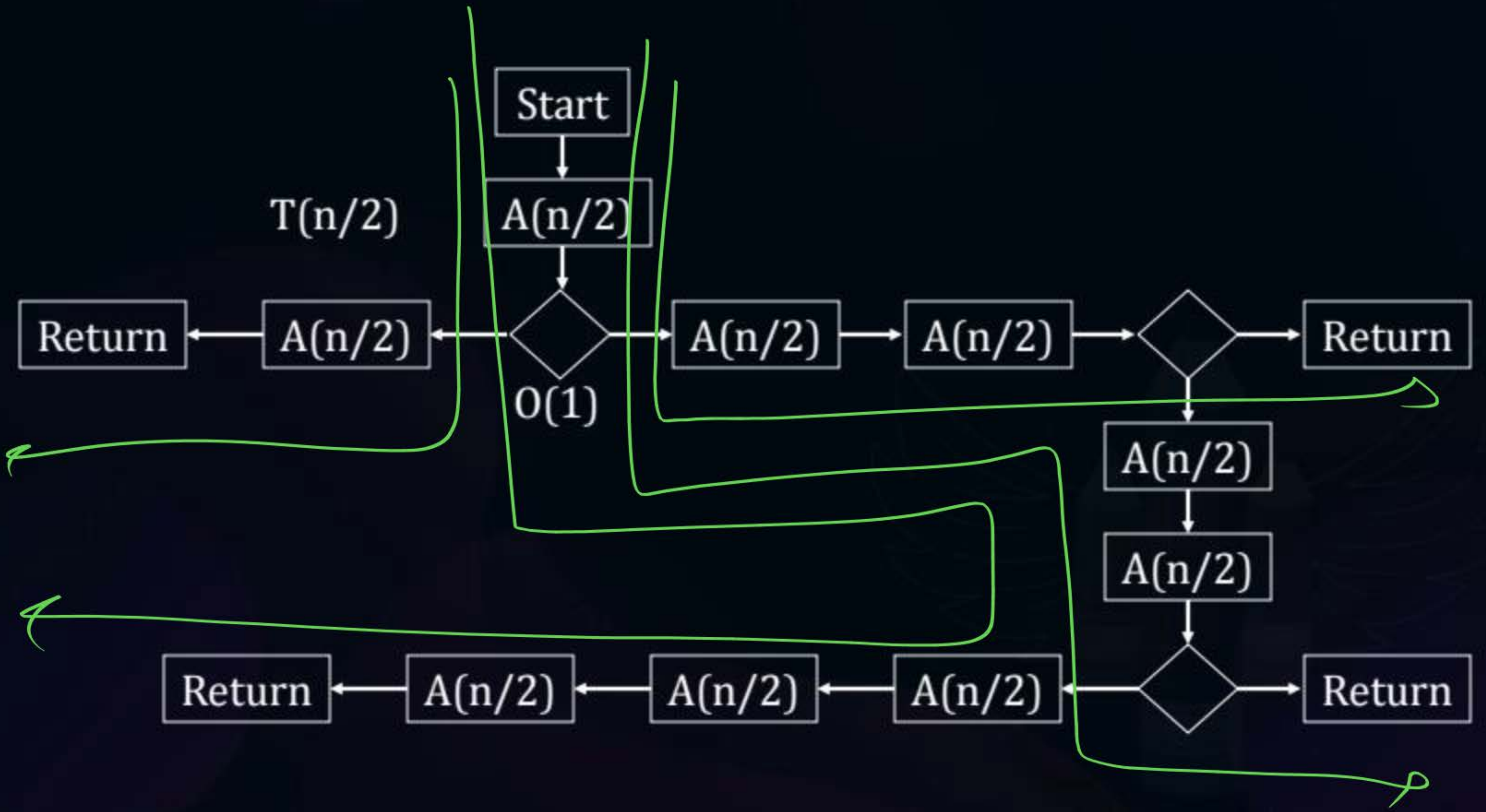
$$T(n) = b + \underline{a * \log n} \Rightarrow \underline{TC : O(\log n)}$$

Topic : Time Complexity Framework for Recursive Algorithms

#Q. The given diagram represents the flowchart of recursive algorithm A(n). Assume that all statement except for the recursive calls have order (1) time complexity. Then the best case and worst case time of this algorithm is ________. **[NAT]**

PYQ

Topic : Time Complexity Framework for Recursive Algorithms



Topic : Time Complexity Framework for Recursive Algorithms

Best Case:

- Case when algorithm takes min steps/does min work/effort.

Worst Case:

- Case when algorithm takes max steps/work/effort.

Topic : Time Complexity Framework for Recursive Algorithms

Case 1:

a) 2 RC ✓ → B.C

b) 3 RC

c) 5 RC

d) 8 RC → W.C

Case 2:

A) $O(n^2)$

B) $O(n^3)$

C) $O(n \log n)$

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

Topic : Time Complexity Framework for Recursive Algorithms

Case 1: Best Case Analysis:

Step 1: Recurrence

$$T(n) = 2T(n/2) + a, n > 1$$

$$T(n) = b, n = 1$$

Topic : Time Complexity Framework for Recursive Algorithms

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

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

$T(n) = 2[2T(n/2^2) + a] + a = 2^2\,T(n/2^2) + 3a$

$T(n) = 2^2\,T(n/2^2) + (2^2 - 1)\,a$

$T(n) = 2^3\,T(n/2^3) + (2^3 - 1)\,a$

.

.

$T(n) = 2^k\,T(n/2^k) + (2^k - 1)\,a$

Topic : Time Complexity Framework for Recursive Algorithms

For BC,

$$n/2^k = 1$$

$$2^k = n$$

$$T(n) = n * T(1) + (n - 1)\,a$$

$$= n * b + an - a$$

$$= O(n)$$

Topic : Time Complexity Framework for Recursive Algorithms

Case 1: Worst Case Analysis:

Step 1: $T(n) = 8T(n/2) + a, n > 1$

$T(n) = b, n = 1$

HW

Topic : Time Complexity Framework for Recursive Algorithms

$T(n) = 8T(n/2) + a$

$T(n/2) = 8T(n/2^2) + a$

$T(n) = 8[8T(n/2^2) + a] + a = 8^2\ T(n/2^2) + 9a$

$T(n) = 8^3\ T(n/2^3) + 8^2a + 9a$

$T(n) = 8^3\ T(n/2^3) + (8^2 + 8^1 + 8^0)a$

.

.

$T(n) = 8^k\ T(n/2^k) + (8^{k-1} + 8^{k-2} + \dots + 8^0)a$

Topic : Time Complexity Framework for Recursive Algorithms

$T(n) = 8^k\, T(n/2^k) + (8^{k-1} + 8^{k-2} + \ldots + 8^0)a$

GP: $a = 1$, $r = 8$, $n = k$

H.W

Topic : Time Complexity Framework for Recursive Algorithms

#Q. Given: [MCQ]

$$T(n) = 2T(n/2) + n \log n, n > 1$$

$$T(n) = 1, n > 1$$

A $O(n \log n)$

B $O(n^2)$

C $O(n)$

D None of these

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

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

HW

Loop Complexity

Topic : Loop Complexity

Types of Loops:

(1) for (2) while (3) do while

✓ ✓

Topic : Loop Complexity

For Loop:

```
for (i = 1; i ≤ n; i++)
{
        S1;
        S2;
        S3;
}
```

Topic : Loop Complexity

While Loop:

```
i = 1;
while (i ≤ n)
{
        S1;
        S2;
        S3;
        i++;
}
```

Topic : Loop Complexity

How to determine the Time Complexity of a Loop:

Time Complexity of any loop depends on two important factors:

(1) The number of times the loop is running/iterating/repeated.

(2) The complexity of all the individual statements within it (inside loop body).

Topic : Loop Complexity

Example 1:

```
for (j = 1; j ≤ n; j++)
{
        a = a + 5; → O(1)
}
```

TC : $\underline{O(n)}$

Topic : Loop Complexity

Example 2:

```
for (i = 1; i <= n/2; i++)
{
        a = a + 10;
}
```

$O(n/2) = \underline{O(n)}$

Topic : Loop Complexity

Example 3:

```
for (i = 1; i ≤ n; i++)
{
        a = a + 3;
        break;
}
```

→ $\underline{O(1)}$

Topic : Loop Complexity

Example 4:

```
a = 0;
for (i = 1; i ≤ n; i++)
{
        a = a + 5;
}
```

Q1. What is the TC of code? → $O(n)$

Q2. What is the value of 'a' after code ends? → $\underline{5 * n}$

Topic : Loop Complexity

Example 5:

```
a = 0;
for (i = 1; i ≤ n; i++)
{
        a = a + i ;
}
print(a)
```

Q1. What is the TC of code?

$O(n)$

Q2. What is the output of given code?

$0+1+2+3 \cdots n$

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

Topic : Loop Complexity

Example 6:

```
Algo AJ(x, n)
{
        for(i = 1; i ≤ n; i++)
        {
                if (x % i == 0) {
                        break;
                }
        }
}
```

TC : $\underline{O(1)}$

Q1. What is the best and worst case time complexity of AJ() and for what type of input?

Topic : Loop Complexity

Example 7:

```
Algo AJ(x, n)
{
    for(i = 2; i ≤ n; i++)
    {
        if (x % i == 0) {
            break;
        }
    }
}
```

n=10
n=13

B.C → O(1)

WC → O(n)

Q1. What is the time complexity of AJ(x,n)?

Topic : Loop Complexity

Example 8:

```
Algo AJ(n)
{
        for(i = 1; i ≤ n; i++)
        {
                AJ2(n);
        }
}
```

$\underline{n * O(AJ2(n))}$

Q1. What is the time complexity of AJ(n)?

Topic : Loop Complexity

C program

Example 9:

```
int c = 0, i ;
for (i = 1; i ≤ n; i++);
c = c + i;
```

Q1. What is the time complexity of given code? → $O(n)$

Q2. What is the exact value of C after the code ends?

$$\boxed{C = n+1}$$

Topic : Loop Complexity

While Loop:

```
i = 1;              // initialization
while (i ≤n)        // condition
{
        printf(i);
        i++;        // updation
}
```

Topic : Loop Complexity

[MCQ]

#Q.

```
i = 1, a = 0;
while (i = 2)
{
        a = a + 1;
}
```

→ always True

TC : ∞

A O(n)

B O(n log n)

C O(1)

D none

Topic : Loop Complexity

#Q. [NAT]

```
a = 0
for (i = 1; i ≤ n; i=+5) {
        a = a + 3;
}
```

(i) What is the time complexity?

(ii) What is the value of a after code ends?

$a = \boxed{\frac{n}{5} * 3}$

1 6 11 16 . . .

$TC : O(n/5)$

$= \underline{O(n)}$

THANK - YOU