CS & IT ENGINEERING

Algorithms

Analysis of Algorithms

Lecture No.- [redacted] 10

By- Aditya sir

Recap of Previous Lecture

Topic

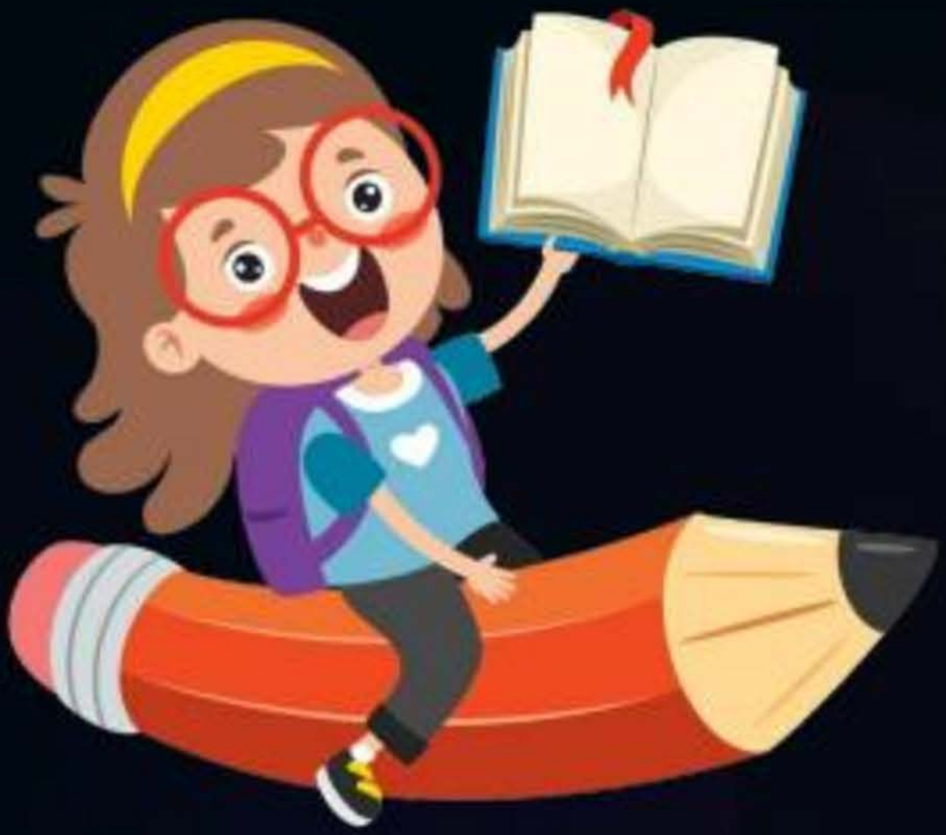
Topic

Topic

Time Complexity Analysis
of Non-Recursive Algo.

Time Complexity for
Recursive Algo.

Topics to be Covered



Topic

Topic

TC for Recursive Algo

Soln to HW Questions:-

① Algo AJ(n) $\rightarrow T(n)$

```
{
    if(n==1)
        return 1
    else
    return (AJ(n-1)+AJ(n-1))
}
```

$\underline{\underline{AJ(n-1)}} \rightarrow T(n-1)$ $\underline{\underline{AJ(n-1)}} \rightarrow T(n-1)$

Ans:- $T(n) = \underline{\underline{O(2^n)}}$

Step1 :- Recurrance Relation

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

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

$$= 2T(n-1) + a, \quad n \geq 1$$

Step2 : Solve Recurrance

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

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

$$T(n) = 2\left[2T(n-2) + a\right] + a$$

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

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

$$T(n-2) = 2T(n-3) + a.$$

$$T(n) = 2^2\left(2T(n-3) + a\right) + 3a$$

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

$$= 2^3 T(n-3) + 4a + 3a$$

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

$$= 2^3 T(n-3) + (8-1)a \longrightarrow (2^3 - 1)a.$$

Generalised form

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

for $\underline{\underline{B.C}}$, $(n-k) = 1$

$$k = (n-1)$$

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

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

<u>Step 3</u> → Apply Asymptotic Notation.

$$\boxed{T(n) = O(2^n)}$$

Homework Soln:-

(2)

Algo $AJ(n)$ → $T(n)$

{

if(n==2)

return 2

else

return $\underbrace{AJ(\sqrt{n})}_{T(\sqrt{n})}$

}

Step1 : Find the Recurrance

$$\boxed{T(n) = b, \quad n = 2}$$

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

Step2:- Solve Recurrence using Back substitution.

$$T(n) = T(\sqrt{n}) + a$$

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

$$T(n^{1/2}) = T(n^{1/4}) + a$$

$$T(n) = \left(T(n^{1/4}) + a\right) + a$$

$$= T(n^{1/4}) + 2a$$

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

$$T(n) = T(n^{1/4}) + 2a$$

$$T(n^{1/4}) = T(n^{1/8}) + a$$

$$T(n) = T(n^{1/8}) + 3a$$

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

Genralised form

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

for Base Condition,

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

$$\log_2\left(n^{1/2^k}\right) = \log_2 2$$

$$\frac{1}{2^k}\log_2 n = 1$$

$$2^k = \log_2 n$$

$$K = \log_2(\log_2 n)$$

$$T(n) = T(2) + K * a$$

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

Step3 :- Apply Asymptotic Notation.

$$\boxed{T(n) = O(\log(\log n))}$$

③ Algo $AJ(n) \rightarrow T(n)$

```
{
    if (n == 2)
        return 2

    return (AJ(√n) + AJ(√n))
}
```

$$\underbrace{AJ(\sqrt{n})}_{T(\sqrt{n})} + \underbrace{AJ(\sqrt{n})}_{T(\sqrt{n})}$$

49.3%

A) $O(n)$

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

C) $O(\log n)$ ✓

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

Soln:- Step1 → Recurrance

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

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

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

Step 2:- Solve Recurrence

$$T(n) = 2T(n^{1/2}) + a \quad — ①$$

$$T(n^{1/2}) = 2T(n^{1/4}) + a$$

$$T(n) = 2\left[2T(n^{1/4}) + a\right] + a$$

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

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

$$T(n) = 2^2\left[2T(n^{1/2^3}) + a\right] + 3a$$

$$= 2^3 T(n^{1/2^3}) + 7a \quad — ②$$

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

$$T(n) = 2^4 T(n^{1/2^4}) + (2^4 - 1)a$$

$\vdots$

Generalised form

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

For Base Condition

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

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

$$\underline{k = \log(\log n)}$$

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

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

Step 3 → Apply Asymptotic Notation

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

$T(n)$

(4) Algo AJ1(n)

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

$$\underbrace{AJ1(n/2)}_{T(n/2)} + \underbrace{AJ1(n/2)}_{T(n/2)} + \underbrace{AJ2(n)}_{O(1)}$$

68%

Case 1: $AJ2(n) = O(1)$ → constant TC.

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

B) $O(n)$ ✓

C) $O(n^2)$

D) $O(\log n)$

Soln:

Step1: Recurrance Relation

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

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

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

Step2 :- Solve Recurrance using Back-Sub

$$T(n) = 2T(n/2) + a \quad — ①$$

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

$$T(n) = 2\left[2T(n/2^2) + a\right] + a$$

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

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

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

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

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

$$T(n) = 2^4 T(n/2^4) + (2^4 - 1)a$$

$\vdots$

Genralise eqn.

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

For Base Condition,

$$n/2^k = 1$$
$$2^k = n$$
$$k = (\log_2 n)$$

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

value of Recurrance

$$T(n) = n \times b + na - a$$
$$\boxed{T(n) = n(a+b) - a}$$

Step 3:- Apply asymptotic notation.

$$T(n) = O(n)$$ ✓

Imp Recursion

⑤ Algo AJ(n)

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

     AJ2(n) → O(n)
  }
}
```

A) $O(n)$

B) $O(n\log n)$

C) $O(\log n)$

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

Soln :- step1 → Recurrance.

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

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

$$\boxed{T(n) = 2T(n/2) + n}$$

Step2: Solve Recurrence.

$$T(n) = 2T(n/2) + n + a \quad ①$$

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

$$T(n) = 2\left[2T(n/2^2) + n/2 + a\right] + n + a$$

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

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

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

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

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

$$T(n) = 2^3 T(n/2^3) + 3n + 7a \quad (8-1)$$

$$T(n) = 2^4 T(n/2^4) + 4n + (2^4 - 1)a$$

$$\vdots$$

Generalised eqn

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

For Base Condition, $n/2^k = 1$

$$2^k = n$$

$$k = (\log n)$$

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

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

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

Step3:- Asymptotic Notation

$$\boxed{T(n) = O(n\log n)}$$

50%

(6) Given a Recurrence, determine the time Complexity.

Advanced Questn

$$\boxed{T(n) = 2, \; n = 2}$$

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

A) $O(n\log n)$
B) $O(n^2)$
C) $O(n\log(\log n))$
D) $O(\log(\log n))$

given →

Step1: $T(n) = \sqrt{n}\, T(\sqrt{n}) + n.$

Step2: Solve using Back substitution.

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

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

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

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

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

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

$$T(n) = n^{(1/2^3 + \frac{1}{2^2} + \frac{1}{2})}\, T(n^{1/2^3}) + n^{(1/2 + 1/2^2)}\, n^{1/2^2} + 2n$$

$$= n^{\left(\frac{1}{2^3} + \frac{1}{2^2} + \frac{1}{2}\right)}\, T(n^{1/2^3}) + n^{\left(1/2 + \frac{1}{2}\right)} + 2n$$

$$= n^{\left(\frac{1}{2^3} + \frac{1}{2^2} + \frac{1}{2}\right)}\, T(n^{1/2^3}) + 3n$$

$$\vdots$$

$$\left[\frac{1}{2^1}+\frac{1}{2^2}+\frac{1}{2^3}\cdots\cdots\frac{1}{2^k}\right] \rightarrow \underline{\underline{GP}}$$

$\underline{\underline{GP}}$ $\left|\begin{array}{l} n = k \\ a = \frac{1}{2} \\ r = 1/2 \end{array}\right.$

$$\Rightarrow \text{Sum} = \frac{a(1-r^n)}{1-r}$$

$$= \frac{\frac{1}{2}\left(1-\left(1/2\right)^k\right)}{1-1/2}$$

$$= \boxed{\left(1-\frac{1}{2^k}\right)}$$

$$n^{(a-b)} = \frac{n^a}{n^b}$$

Generalised form.

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

For Base Condition

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

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

$$\log n = 2^k$$

$$\underline{k = \log(\log n)}$$

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

$$= \frac{n}{2} * 2 + n\log(\log n) \implies \underline{O(n\log(\log n))}$$

(7) Algo AJ(n)

```
{
    if (n == 1)
        return 1
    else
        return [AJ(√n) + 10]
}
```

Soln :

Step1 : $T(n) = b,\ n = 2$

Step2 : $T(n) = T(\sqrt{n}) + a,\ n > 1$

$T(n^{1/2}) = T(n^{1/4}) + a$

$$T(n) = \left(T(n^{1/4}) + a\right) + a$$

$$T(n) = T(n^{1/4}) + 2a$$

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

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

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

genralised,

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

For B.C,

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

$$\frac{1}{2^k}\log n = \log_2 2$$

$$2^k = \log n$$

$$k = \log(\log n) \ldots$$

⑧ Algo AJ(n)

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

HW I : $\underline{\underline{O(n)}}$

return $AJ(n/2) + \underline{\underline{B(n)}}$

$\downarrow$

$O(n)$

Soln:-

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

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

V. famous Recurrance

(Binary Search)

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

$$T(n) = T(n/2^3) + 3a$$
$$= T(n/2^4) + 4a$$
$$\vdots$$

general

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

for B.C $\quad n/2^k = 1$

$$2^k = n$$
$$k = (\log_2 n)$$

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

$$T(\rightarrow \underline{O(\log_2 n)}$$

2 mins Summary

Topic

Topic

Imp questions on

Recursive Algo Time Complexity Analysis

using Back Substitution.

THANK - YOU

Telegram Link: https://t.me/AdityaSir_PW