PROBLEM SOLVING

Presentation by Grace Hephzibah M
RA2011026020084

OBJECTIVES:

- FORMULATION OF RECURRENCE EQUATION
- SOLVING RECURRENCE EQUATION
    1. GUESS AND VERIFY
    2. SUBSTITUTION
    3. RECURRENCE TREE

Formulation Of Recurrence Equations

1. 1000, 2000, 4000, 8000
2. 7 , 21/4 , 63/16 , 189/64
3. Maximum possible edges in a graph
4. Staircase Problem
5. Triangular Number

Solving Recurrence Equation

- Solution must be non-recursive
- This solution is called closed-form solution
- Sometimes, there might not be a closed form solution

Closed – Form Solution

- **General Solution**

  Example : $T(n) = n!$

- **Particular Solution**

  Example : $T(n) = T(0) + 5$

Guess And Verify

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

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

3. $T(n) = 3T(n/2)$

Substitution Method

- Also called iteration method.
- Or the Plug and Chug Method

- Plug means Substitute
- Chug means Evaluate

Forms

Forward Substitution

From base to T(n)

Backward Substitution

From T(n) to base

Substitution Method

1. $T(n) = T(n-1) + 3$
   $T(1) = 4$
2. Compound Interest for \$100 at 3%

3. $T(n) = n*T(n-1)$
   $T(0) = 1$

4. $T(n) = k*T(n-1)$

Recurrence Tree Method

1. $$T(n) = \begin{cases} 1 & n=1 \\ T(n-1) + a & n>1 \end{cases}$$

Find the time complexity when a=1 and when a=n

Recurrence Tree Method

$$
1.\ T(n) = \begin{cases} 1 & n=1 \\ 8*T(n/2) & n>1 \end{cases}
$$

Find the time complexity

THANK YOU

Formulation of recurrence equation

1. 1000, 2000, 4000, 8000

$t_0 = 1000$
$t_1 = 2000 = 2 \times t_0 \quad \Rightarrow t_n = 2 \times t_{n-1}$
$t_2 = 4000 = 2 \times t_1$
$t_3 = 8000 = 2 \times t_2$

2. $7, \frac{21}{4}, \frac{63}{16}, \frac{189}{64}$

$t_0 = 7$
$t_1 = t_0 \times \frac{3}{4}$
$\Rightarrow t_n = t_{n-1} * \frac{3}{4}$
$t_2 = t_1 \times \frac{3}{4}$
$t_3 = t_2 \times \frac{3}{4}$

3. Max possible edges among a graph

$n \geq 1$

~~$t_0 = 0$~~
$t_1 = 0$
$t_2 = 1$

$t_3 = 2$

$t_4 = 6$.

$t_{n-1} = (n-1) + t_{n-1}$.

4. Staircase Problem (1, 2 steps)

$t_1 = 1$.

$t_2 = 2$

$t_3 = 3$.

$t_4 = 5$.

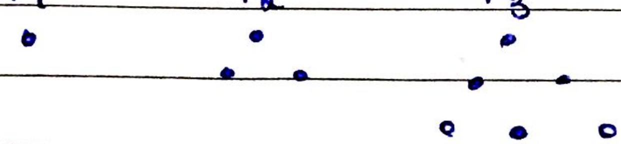
$t_n = t_{n-1} + t_{n-2}$

5. Towers of Hanoi

No. of disk	Moves
1	1
2	3
3	7
4	15
5	31

6. Triangular number.

T_1 T_2 T_3



$t_n = n + t_{n-1}$

Techniques for solving recurrence equations.

→ must be non recursive.

→ such solution

Closed form solution → general solution
$$t_n = \frac{n(n+1)}{2}$$

Closed form solution → particular solution
$$t_n = t_0 \pm\!*\!/ \;(\;\;\;\;)$$

→ sometimes there might not be a closed form solution.

① Guess & verify method.

1. $t_n = t_{n-1} + 2$

   $t_0 = 1$

$t_0 = 1$

$t_1 = t_{n-1} + 2 = 1 + 2$

$t_2 = t_{n-1} + 2 = (1+2) + 2$

$\quad = 1 + 2*2.$

$t_3 = t_2 + 2 = 1 + 3*2.$

Guess

Guess: $t_n = n*2 + 1$

Verify:

$$\left.\begin{aligned} t_0 &= 2(0)+1 = 1 \\ t_1 &= 2(1)+1 = 3 \\ t_2 &= 2(2)+2 = 6 \end{aligned}\right\}\text{ Part 1}$$

$$\cancel{t_{n+1} = 2(n+1)+2}$$
$$\cancel{= 2n}$$

$$\left.\begin{aligned} t_{n+1} &= t_n + 2 \\ &= (2n+1)+2 \\ &= 2n+2+1 \\ &= 2(n+1)+1 \end{aligned}\right\}\text{ Part 2}$$

2. $t_n = t_{n-1} + n^2$

$t_1 = 1$.

Solution: $t_n = \dfrac{(n)(n+1)(2n+1)}{6}$

3. $T(n) = 3T\left(\frac{n}{2}\right)$

$T(1) = 1$

$T(1) = 1$

$T(2) = 3T(1) = 3$

$T(4) = 3T(2)$

$= 3[T(1) * 3]$

$= 3^2$

$T(8) = 3T(4)$

$= 3^3$

$T(n) = 3^{\log_2 n}$ → Guess

$T(2n) = 3T\left(\frac{2n}{2}\right)$ // Verify

$= 3T(n)$

$= 3 * 3^{\log_2 n}$

$= 3^{\log_2 n + 1}$

$= 3^{\log_2 n + \log_2 2}$

$= 3^{\log_2 2n}$ $\leftrightarrow$.

Substitution Method.

→ iteration method

→ Plug & Chug

From base to t_n

↳ forward substitution

from t_n to base

↳ backward substitution

(or)

backtracking method.

1. $t_n = t_{n-1} + 3$

$t_1 = 4$

$$
\begin{aligned}
t_n &= t_{n-1} + 3 \\
&= (t_{n-2} + 3) + 3 \quad \text{// Plug} \\
&= t_{n-2} + 3 + 3 \\
&= t_{n-2} + 2*3 \quad \text{// Chug} \\
&= (t_{n-3} + 3) + 3*2 \quad \text{// P} \\
&= t_{n-3} + 3*3 \quad \text{// C.}
\end{aligned}
$$

$$
\begin{aligned}
t_n &= t_{n-(n-1)} + 3*(n-1) \\
&= t_1 + (n-1)*3 \\
&= 4 + 3n - 3 \\
&= 3n + 1.
\end{aligned}
$$

2. Compound interest

\$100 3%.

[50th month.

$t_0 = 100$

$t_1 = t_0 + 0.03\,t_0 = 1.03\,t_0.$

$t_2 = t_1 + 0.03\,t_1 = 1.03\,t_1$

$t_n = 1.03\,t_n$ → recurrence sol.

$$t_n = 1.03\, t_{n-1} \quad \text{/Plug}$$
$$= (1.03)^2 t_{n-2} \quad \text{//Chug}$$
$$= (1.03)^n t_0.$$
$$= 100\,(1.03)^n$$

$$t_{50} = 100\,(1.03)^{50}.$$

3. $t_n = n t_{n-1}$
$t_0 = 1$

$$t_n = n t_{n-1}$$
$$= n(n-1) t_{n-2}$$
$$= n(n-1)(n-2) t_{n-3}$$
$$= n!.$$

4. $t_n = 7 t_{n-1}$ Solution: 7^n
$t_0 = 1$

5. $t_n = k t_{n-1}$ where $t_3 = 343$
$t_4 = 2401$
$t_0 = 1$

$$t_n = k^n . t_0 = k^n.$$

$t_3 = k^3$

$t_4 = k^4$

$\Rightarrow \frac{t_4}{t_3} = \frac{k^4}{k^3} = K = \frac{2401}{343} = 7$

Forward Substitution

1. $t_n = t_{n-1} + 3$

   $t_0 = 4.$

$t_0 = 4$

$t_1 = t_0 + 3$ plug.

$t_2 = t_1 + 3$ plug

$= t_0 + 3 + 3$

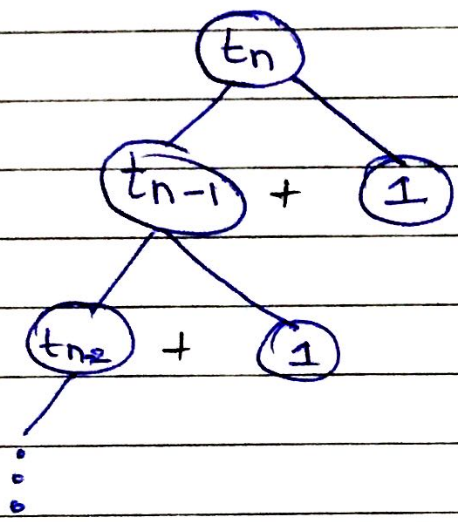
$= t_0 + 2 * 3$ chug.

$t_n = t_0 + n * 3.$

Recurrence - tree Method.

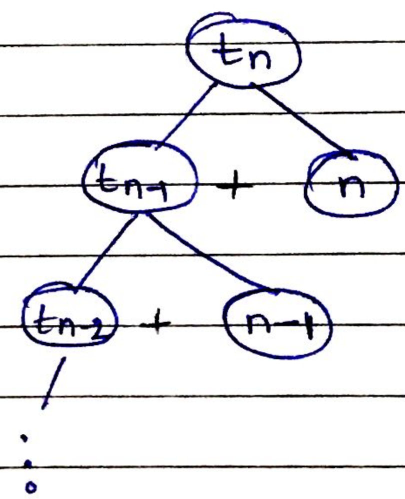
1. $t_n = \begin{cases} 1 & n=1 \\ t_{n-1}+a & n>1 \end{cases}$ $\quad a=1$, $a=n$.

When $a=1$



$$\sum_{i=1}^{n} 1 = n.$$

When $a=n$.

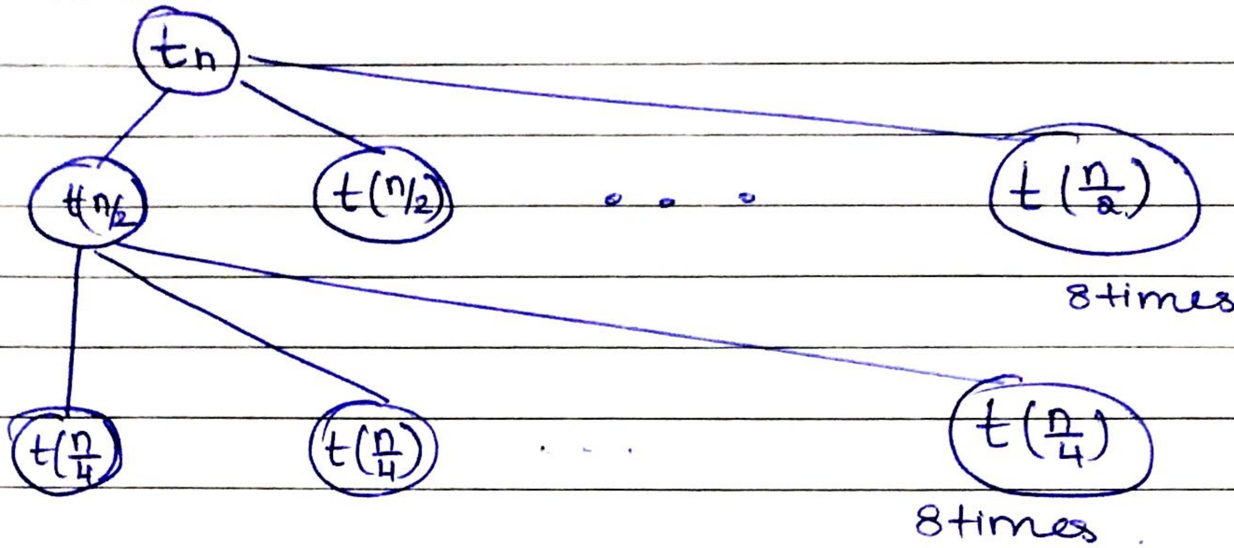


$\Rightarrow \cancel{n + (n-1) + (n-2) + \dots + 1}$

$\Rightarrow n + (n-1) + (n-2) + \dots + 1$

$\Rightarrow \dfrac{n(n+1)}{2}.$

2. $t_n = \begin{cases} 1 & n=1 \\ 8T\left(\frac{n}{2}\right) & n>1. \end{cases}$



<u>Levels</u>

$1, 8, 8^2, 8^3, \dots 8^i$

$8^{\log n} * T(1)$

$= 8^{\log n}$

$= n^{\log_2 8}$

$= n^3 \rightarrow O(n^3)$

asymptotic upper bound.