07/06/2024 **Analytical 1 - DAA**

A.J.NOWAL KHANNA
192825013
B.Tech (AI & ML)

1) solve the following recurrence relations

a) $x(n) = x(n-1) + 5$ for $n>1$ $x(1) = 0$

$\boxed{x(n) = n(n-1)+5}$ — ①

Let $(n = n-1)$ in ①

$x(n-1) = x(n-1-1)+5 = x(n-2)+5$

Let $(n = n-1)$ in ①

$n(n-2) = n(n-3) + 5$ — ③

Now,

$x(n) = x(n-1) + 5$

$x(n) = x(n-2)+10 \rightarrow$ ④

Put the valu of $(x-2)$ in ③, ④

$x(n) = x(n-3) + 5 + 5 + 5 = x(n-3)+15$ — ⑤

from or no ① ④ & ⑤

$x(n) = x(n-i) + 5i$ for $i<n$

Put $i = n-1$,

$x(n) = x(n-(n-1)) + 5(n-1) x(n)$

$= x(1) + 5(n-1)$

$x(n) = x(1) + 5(n-1)$

$(x(1) = 0)$ then;

$\boxed{x(n) = 5(n-1)}$

b). $x(n) = 3x(n-1) \rightarrow$ ① $(n(1) = 4)$

$\underline{n>1}$

Let $(n = n-1)$;

$x(n-1) = 3x(n-1-1)$

$x(n-1) = 3x(n-2)$ — ②

Let $(n = n-2)$; in ①

$x(n-2) = 3x(n-2-1)$

$x(n-2) = 3x(n-3)$ — ③

now $x(n-1)$ value in ①,

$x(n) = 3.3x(n-2) = 3^2 x(n-2) \rightarrow$ ④

put $x(n-2)$ in ④,

$x(n) = 3^3 x(n-3) \rightarrow$ ⑤

from ①, ④ & ⑤

$x(n) = 3^i x(n-i)$ for $i < n$

Now, $i = n-1$;

$x(n) = 3^{n-1} x(n-(n-1))$ $[x(n) = 3^{n-1} x(1)]$

$x(n) = 3^{n-1} x(1)$

but $x(1) = 4$ then from Question.

$\boxed{x(n) = 3^{n-1} 4}$

c) $x(n) = x(n/2) + n$ for $n > 0$ $x(1) = 1$ (for $n \cdot 2k$)

$x(n) = n(n/2) + n \rightarrow$ ①

$x(1) = 1$;

Sub $(n = 2k)$ then;

$n(2^k) = x(2^{k-1}) + 2k \rightarrow$ ②

Sub $(2k = 2^{k-1})$

$x(2^{k-1}) = x(2^{k-2}) + 2^{k-1} \longrightarrow$ ③

Sub $(2^k = 2^{k-1})$

$x(2^{k-2}) = x(2^{k-1})$

$x(2^k) = x(2^{k-3}) + 2^{k-2}$

when $(2k = 0)$

$x(2^0) = x(1) = 1$

from ⑤

The sum of the fir

$2^0 + 2^1 + 2^2 + \dots + 2^k$

So we have,

$x(2^k) = 2^{k+1}$

Sub $(2^k = n/2)$

$x(n) = 2^k$

d) $x(n) = x(n/3) + 1$ for

$x(n) = x(n/3) + 1$

But $(x(1) = 1)$;

Sub $(n = 3^k)$ in

$x(3^k) = x\left(\frac{3^k}{3}\right)$

$x(3^k) = x(3^{k-1}$

② Evaluate the fol

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

$n = 2^k$, $k = \log n$

$= T\left[\frac{2^k}{2}\right] + 1$

$T(2^k) = T(2^{k-1}$

Sub $(2^k = 2^{k-1})$

$$x(2^{k-2}) = x(2^{k-1}) + 2^{k-2} \longrightarrow ④$$

$$x(2^k) = x(2^{k-3}) + 2^{k-2} + 2^{k-1} + 2^k \longrightarrow ⑤$$

when $(2k = 0)$

$$x(2^0) = x(1) = 1$$

from ⑤

The sum of the first;

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

So we have,

$$x(2^k) = 2^{k+1} - 1$$

Sub $(2^k = n/2)$

$$x(n) = 2^{n/2} - 1 \Rightarrow \boxed{2^{n/2} - 1}$$

d) $x(n) = x(n/3) + 1$ for $n > 1$, $n(1) = 1$ (solve for $n = 3^k$)

$$x(n) = x(n/3) + 1 \longrightarrow ①$$

But $(x(1) = 1)$;

Sub $(n = 3^k)$ in ①

$$x(3^k) = x\left(\frac{3^k}{3}\right) + 1$$

$$x(3^k) = x(3^{k-1}) + 1 \longrightarrow ②$$

② Evaluate the following

i) $T(n) = T(n/2) + 1$ where $n = 2^k$ for all $k \geq 0$

$n = 2^k$, $k = \log n$

$$= T\left[\frac{2^k}{2}\right] + 1$$

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

$$= \left[T(2^{k-3}) + 1\right) + 2$$

$$= T(2^{k-3}) + 3$$

$$T(2^k) = T(2^{k-k}) + k$$

$$= T(2^0) + k \Rightarrow T(1) + k$$

$T(1) = 1$ then,

$T(2^k) = 1+k$

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

$\therefore$ we got $T(n) = \Theta(\log n)$

2) $T(n) = T(n/3) + T(2n/3) + cn$,

$T(n) \leq T(n/3) + T(n/3) + cn$

$\leq d(n/3)\log(n/3) + d(2n/3)\log(2n/3) + cn$

$= (d(n/3)\log n - d(n/3)\log 3)$

$+ (d(2n/3)\log n - d(2n/3)\log(3/2)) + cn$

$= dn\log n - dn(\log 3 - 2/3) + cn$

$\leq dn\log n$

$[\because d \geq c/(\log(3) - 2(3)]$ $\therefore$ order $= O(n\log n)$

3) consider the following recursion algorithm:

```
min1(A[0.....n-1])
if n=1 return A[0]
else temp = min1 [A[0.....n-2])
    if temp <= A[n-1] return temp
Else
    return A[n-1]
```

a) The recursive algorithm computes the min value in array A of size n it does this by comparing the cost element of the array n[n-1] with the minimum value of the rest of the array A[0....n-2] & returning smaller value.

b) The algorithm makes call to min A [0....n-1]) which is n-1, then

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

Time $(n) = 2n+3$

dn it poly $\}$ $O(n)$

c is the constant representing the time taken for the operation outside the call

```
def min1 (A, n):
    if n = = 1:
        return A [0]
    else:
        temp = min1 (A, n - 1)
        if temp <= A [n - 1] :
            return temp
        else :
            return A [n - 1]
```