HW2

Ronan Wallace

February 28, 2019

Section 2.4 p.76-77

1c) $x(n) = x(n-1) + n$ for $n > 0,\ x(0) = 0$

Forwards Substitution

$x(0) = 0$
$x(1) = x(0) + 1 \rightarrow 1$
$x(2) = x(1) + 2 \rightarrow 3$
$x(3) = x(2) + 3 \rightarrow 6$
$x(4) = x(3) + 4 \rightarrow 10$
$x(5) = x(4) + 5 \rightarrow 15$

As n increases by one, n is added to the previous answer to get the next one. Because of this, the relation is a summation:

$$\sum_{i=0}^{n} i = \frac{n(n+1)}{2} \rightarrow \Theta(n^2)$$

1e) $x(n) = x(\frac{n}{3}) + 1$ for $n > 1,\ x(1) = 1$ (solve for $n = 3^k$)

Forwards Substitution

$x(1) = 1$
$x(3) = x(1) + 1 \rightarrow 2$
$x(9) = x(3) + 1 \rightarrow 3$
$x(27) = x(9) + 1 \rightarrow 4$
$x(81) = x(27) + 1 \rightarrow 5$

A pattern can be seen in the powers of 3:

$1 = 3^{0\leftarrow}$
$3 = 3^{1\leftarrow}$
$9 = 3^{2\leftarrow}$
$27 = 3^{3\leftarrow}$
$81 = 3^{4\leftarrow}$

$log_3 1 = 0$
$log_3 3 = 1$
$log_3 9 = 2$
$log_3 27 = 3$
$log_3 81 = 4$

$x(3^0) = log_3 1 + 1 = 1$
$x(3^1) = log_3 3 + 1 = 2$
$x(3^2) = log_3 9 + 1 = 3$
$x(3^3) = log_3 27 + 1 = 4$
$x(3^4) = log_3 81 + 1 = 5$

Recurrence Relation:
$log_3 n + 1$

4b) ALGORITHM $Q(n)$
//Input: A positive integer n
if $n = 1$ **return** 1
else return $Q(n-1) + 2 * n - 1$

Tracing when $n = 4$:
$Q(4) \to Q(3) + 2*4 - 1 \to Q(2) + 2*3 - 1 + 7 \to Q(1) + 2*2 - 1 + 12 \to$
$1 + 15 \to 16$ (three multiplications have been made)

$Q(n-1) + 1$ for $n > 1$

Forwards Substitution:
$Q(1) = 0$ multiplications $\leftarrow$ initial condition
$Q(2) = 1$ multiplication
$Q(3) = 2$
$Q(4) = 3$
$Q(5) = 4$

$Q(n) = n - 1$

Recurrence Relation:
$n - 1$

5a) Tower of Hanoi

$H(n)$ = least number of moves to "win" where n = number of disks

To find a recurrence relation, I played the game to record some of the outputs of $H(n)$ where $n \geq 1$

$H(1) = 1$ move $\leftarrow$ initial condition
$H(2) = 3$ moves
$H(3) = 7$
$H(4) = 15$
$H(5) = 31$
$H(6) = 63$

A pattern can be seen that when n increases by 1, the output increases by 2^n. For example, when n increases from 1 to 2 ($H(1) \rightarrow H(2)$), the output increases by 2^1. Second example, when n increases from 4 to 5 ($H(4) \rightarrow H(5)$), the output increases by 2^4. We can see a pattern that the output increases by 2^n each time from the initial condition. Subtracting 1 from 2^n will give us the recurrence relation for $H(n)$.

Recurrence Relation:
$2^n - 1$

Now that we have the relation, we can set $n = 64$ disks to calculate the amount of moves it will take to "win":

$$2^{64} - 1 = \text{number of moves}$$

Now that we have the number of moves it would take to complete a game of Hanoi at 64 disks, we can find how long it will take to solve it at one disk move per minute. Since each move takes one minute, then the game will take $2^{64} - 1$ minutes. Having the minutes, we can divide by minutes in an hour, hours in a day, and days in a year to get the number of years it will take to finish the game:

$$\frac{2^{64} - 1}{60 * 24 * 365} = 35,096,545,041,304 \text{ years}$$

If immortal monks were to play a game of Tower of Hanoi with 64 disks, moving a disk every minute without sleeping or eating, it would take approximately 35 trillion years.

5c) Included in .zip folder