Homework 2: Recurrence Relations

Arjun Koshal

October 28, 2022

Problem 1

Solve the following recurrence relations (using 5-step backward substitution method with a demonstration of all the steps):

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

b) $x(n) = 3x(n-1)$ for $n > 1$, $x(1) = 4$

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

d) $x(n) = x(\frac{n}{2}) + n$ for $n > 1$, $x(1) = 1$ (solve for $n = 2^k$)

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

Solution:

a) Step 1:
$x(n-1) = x(n-2) + 5$
$x(n) = x(n-2) + 10$

Step 2:
$x(n-2) = x(n-3) + 5$
$x(n) = x(n-3) + 15$

Step 3:
$x(n) = x(n-i) + 5i$

Step 4:
Initial Condition $= x(1) = 0$, $n - i = 1$, $i = n - 1$

Step 5:
$x(n) = x(n-(n-1)) + 5(n-1)$
$x(n) = x(1) + 5(n-1)$
$x(n) = 0 + 5(n-1)$
$x(n) = 5(n-1)$

b) Step 1:
$x(n-1) = 3x(n-2)$
$x(n) = 3 \cdot (3x(n-2))$
$x(n) = 9x(n-2)$

Step 2:
$x(n-2) = 3x(n-3)$
$x(n) = 9 \cdot (3x(n-3))$
$x(n) = 27x(n-3)$

Step 3:
$x(n) = 3^i x(n-i)$

Step 4:
Initial Condition $= x(1) = 4$, $n - i = 1$, $i = n-1$

Step 5:
$x(n) = 3^{n-1} \cdot x(n-(n-1))$
$x(n) = 3^{n-1} \cdot x(1)$
$x(n) = 3^{n-1} \cdot 4$

c) Step 1:
$x(n-1) = x(n-2) + (n-1)$
$x(n)(x(n-2) + (n-1)) + n$

Step 2:
$x(n-2) = x(n-3) + (n-2)$
$x(n) = (x(n-3) + (n-2)) + (n-1) + n$

Step 3:
$x(n) = x(n-i) + (n-i+1) + (n-i+1) + ... + (n-i+i)$

Step 4:
$x(0) = 0$, $n - i = 0$, $n = i$

Step 5:
$x(n) = x(n-n) + (n-n+1) + (n-n+2) + ... + (n-n+n)$
$x(n) = x(0) + 1 + 2 + ... + n$
$x(n) = 0 + 1 + 2 + ... + n$
$x(n) = \frac{n(n+1)}{2}$

d) Step 1:
$x(2^k) = x(\frac{2^k}{2^1}) + 2^k$
$x(2^k) = x(2^{k-1}) + 2^k$

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

Step 3:
$x(2^{k-2}) = x(2^{k-3}) + 2^{k-2}$
$x(2^k) = (x(2^{k-3}) + 2^{k-2}) + 2^{k-1} + 2^k$

Step 4:
$x(2^k) = x(2^{k-i} + 2^{k-i+1} + 2^{k-i+2} + 2^{k-k+k})$

Step 5:
$x(1) = 1,\ 2^{k-i} \cdot log(2) = \log(1)$
$(k-i) \cdot \log(2) = 0,\ k-i = 0,\ i = k$
$x(2^k) = x(2^{k-k}) + 2^{k-k+1} + 2^{k-k+2} + 2^{k-k+k}$
$x(2^k) = x(2^0) + 2^1 + 2^2 + ... + 2^k$
$x(2^k) = x(1) + 2^1 + 2^2 + ... + 2^k$
$x(2^k) = 1 + (2^1 + 2^2 + ... + 2^k)$
$x(2^k) = 1 + (\sum_{i=1}^{n} A^i = (\frac{A^{n+1}-1}{A-1} - 1))$
$x(2^k) = 1 + \frac{2^{k+1}-1}{2-1} - 1$
$x(2^k) = 1 + 2^{k+1} - 1 - 1$
$x(2^k) = 2^{k+1} - 1$
$n = 2^k$
$k = \log_2 n$
$x(n) = 2^{\log_2 n+1} - 1$
$x(n) = 2^{\log_2 n} \cdot 2^1 - 1$
$x(n) = 2n - 1$

e) Step 1:
$x(3^k) = x(\frac{3^k}{3^1} + 1$
$x(3^k) = x(3^{k-1}) + 1$

Step 2:
$x(3^{k-1}) = x(3^{k-2}) + 1$
$x(3^k) = (x(3^{k-2}) + 1) + 1$
$x(3^k) = x(3^{k-2}) + 2$

Step 3:
$x(3^{k-2}) = x(3^{k-3}) + 1$
$x(3^k) = (x(3^{k-3}) + 1) + 2$
$x(3^k) = x(3^{k-3}) + 3$

Step 4:
$x(3^k) = x(3^{k-i}) + i$

Step 5:
$x(1) = 1$, $3^{k-i} = 1$, $(k–i) \cdot \ln(3) = \ln(1)$, $(k–i) \cdot \ln(3) = 0$, $k - i = 0$, $i = k$
$x(3^k) = x(3^{k-k}) + k$
$x(3^k) = x(3^0) + k$
$x(3^k) = x(1) + k$
$x(3^k) = 1 + k$
$n = 3^k$
$k = \log_3 n$
$x(n) = 1 + \log_3 n$

Problem 2

For each function below, compute the recurrence relation for its running time and then use the Master Theorem to find its complexity by specifying the different terms of the term explicitly:

```
int f(int arr[], int n) {
    if (n == 0) {
        return 0;
    }
    int sum = 0;
    for (int j = 0; j < n; ++j) {
        sum += arr[j];
    }
    return f(arr, n / 2) + sum + f(arr, n / 2);
}
```

Solution:

a = Number of times recursive function is called
b = Number that's dividing
d = Power of time complexity of f(n)

$a = 2$
$b = 2$
For loop runs n times so $f(n) = \Theta(n^1)$ so $d = 1$

Check if $a = b^d$, $a > b^d$ or $a < b^d$

$2 = 2^1$
$2 = 2$
Since $a = b^d$, $T(n) = \Theta(n^d \log_b n) = \Theta(n \log_2 n)$.

```
void g(int n, int arrA[], int arrB[]) {
    if (n == 0) {
        return;
    }
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; ++j) {
            arrB[j] += arrA[i];
            }
        }
    }
    g(n / 2, arrA, arrB);
}
```

Solution:

a = Number of times recursive function is called
b = Number that's dividing
d = Power of time complexity of f(n)

$a = 1$
$b = 2$
$g(n) = \Theta(n^2)$ so $d = 2$

Check if $a = b^d$, $a > b^d$ or $a < b^d$

$1 < 2^2$
$1 < 4$
Since $a < b^d$, $T(n) = \Theta(n^d) = \Theta(n^2)$.

Problem 3

Use the Master Theorem to find the complexity of each of the following recurrence relations (show all the steps and the values of different terms to apply the theorem):

a) $T(n) = T(\frac{n}{2}) + n^2$

b) $T(n) = 4T(\frac{n}{2}) + n^2$

c) $T(n) = 3T(\frac{n}{3}) + \sqrt{n}$

Solution:

a) $a = 1$
$b = 2$
$f(n) = n^2$ so $d = 2$
Check if $a = b^d$, $a > b^d$ or $a < b^d$

$1 < 2^2$
$1 < 4$
Since $a < b^d$, $T(n) = \Theta(n^d) = \Theta(n^2)$.

b) $a = 4$
$b = 2$
$f(n) = n^2$ so $d = 2$
Check if $a = b^d$, $a > b^d$ or $a < b^d$

$4 = 2^2$
$4 = 4$
Since $a = b^d$, $T(n) = \Theta(n^d \log_b n) = \Theta(n^2 \log_2 n)$.

c) $a = 3$
$b = 3$
$f(n) = n^{\frac{1}{2}}$ so $d = \frac{1}{2}$
Check if $a = b^d$, $a > b^d$ or $a < b^d$

$3 > 3^{\frac{1}{2}}$
Since $a > b^d$, $T(n) = \Theta(n^{\log_b a}) = \Theta(n^{\log_3 3}) = \Theta(n)$.