CS5720 Design Annd Analysis of Algorithms

Raja Kantheti

Problem 1

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

Method: Back Substitution

To solve this, let's write out the first few terms:

$$\begin{aligned}
x(2) &= x(1) + 5 = 0 + 5 = 5, \\
x(3) &= x(2) + 5 = 5 + 5 = 10, \\
x(4) &= x(3) + 5 = 10 + 5 = 15, \\
&\vdots \\
x(n) &= x(n-1) + 5.
\end{aligned}$$

We can see the pattern:

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

General solution:

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

Big Theta notation:

$$x(n) = \Theta(n)$$

(b) $x(n) = 3x(n-1)$ for $n > 1$, $x(1) = 0$

Method: Back Substitution

For $n > 1$:

$$\begin{aligned}
x(2) &= 3x(1) = 3 \cdot 0 = 0, \\
x(3) &= 3x(2) = 3 \cdot 0 = 0.
\end{aligned}$$

We can see that:

$$x(n) = 0$$

General solution:

$$x(n) = 0$$

Big Theta notation:

$$x(n) = \Theta(1)$$

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

Method: Back Substitution

To solve this, let's write out the first few terms:

$$\begin{aligned} x(1) &= x(0) + 1 = 0 + 1 = 1, \\ x(2) &= x(1) + 2 = 1 + 2 = 3, \\ x(3) &= x(2) + 3 = 3 + 3 = 6, \\ x(4) &= x(3) + 4 = 6 + 4 = 10, \\ &\vdots \\ x(n) &= x(n-1) + n. \end{aligned}$$

We can see the pattern:

$$x(n) = \frac{n(n+1)}{2}$$

General solution:

$$x(n) = \frac{n(n+1)}{2}$$

Big Theta notation:

$$x(n) = \Theta(n^2)$$

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

Method: Back Substitution

Let $n = 2^k$.

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

To solve this, let's use back substitution:

$$\begin{aligned} x(2^k) &= x(2^{k-1}) + 2^k, \\ x(2^{k-1}) &= x(2^{k-2}) + 2^{k-1}, \\ x(2^k) &= x(2^{k-2}) + 2^{k-1} + 2^k, \\ &\vdots \\ x(2^k) &= x(1) + 2 + 4 + \ldots + 2^k. \end{aligned}$$

Sum of the geometric series:

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

Since $n = 2^k$:

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

General solution:

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

Big Theta notation:

$$x(n) = \Theta(n)$$

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

Method: Back Substitution

Let $n = 3^k$.

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

To solve this, let's use back substitution:

$$\begin{aligned} x(3^k) &= x(3^{k-1}) + 1, \\ x(3^{k-1}) &= x(3^{k-2}) + 1, \\ x(3^k) &= x(3^{k-2}) + 1 + 1, \\ &\vdots \\ x(3^k) &= x(1) + k. \end{aligned}$$

Since $x(1) = 1$ and $k = \log_3 n$:

$$x(n) = 1 + \log_3 n$$

General solution:

$$x(n) = 1 + \log_3 n$$

Big Theta notation:

$$x(n) = \Theta(\log n)$$

Problem 2

Master Theorem: Solved some of this probems using a direct method called the Master theorem.

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

Method: Master Theorem

To solve this, we use the Master Theorem for divide-and-conquer recurrences:

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

Here, $a = 2$, $b = 2$, and $f(n) = n^3$.
We compare $f(n)$ with $n^{\log_b a}$:

$$\log_b a = \log_2 2 = 1$$

Since $f(n) = n^3$ which is $\Theta(n^3)$, and $n^3 > n^1$: By case 3 of the Master Theorem:

$$T(n) = \Theta(n^3)$$

General solution:

$$T(n) = \Theta(n^3)$$

(b) $T(n) = T(\sqrt{n})\sqrt{n} + n$. Here, assume that $T(2) = c$.

Method: Back Substitution

Let $n = 2^k$. Then $\sqrt{n} = 2^{k/2}$.

$$T(2^k) = T(2^{k/2}) \cdot 2^{k/2} + 2^k$$

To solve this, we use back substitution:

$$\begin{aligned}
T(2^k) &= T(2^{k/2}) \cdot 2^{k/2} + 2^k, \\
T(2^{k/2}) &= T(2^{k/4}) \cdot 2^{k/4} + 2^{k/2}, \\
T(2^k) &= \left(T(2^{k/4}) \cdot 2^{k/4} + 2^{k/2}\right) \cdot 2^{k/2} + 2^k.
\end{aligned}$$

Let's denote $T(2^k) = T(2^{k/2}) \cdot 2^{k/2} \cdot 2^{k/4} \cdot 2^{k/8} \cdot \ldots + k \cdot 2^k$.

Therefore, the general form can be derived, but since it's complex, we often use asymptotic notation:

$$T(n) = \Theta(n \log \log n)$$

General solution:

$$T(n) = \Theta(n \log \log n)$$

(c) $T(n) = 2T\left(\frac{n}{2}\right) + n \log n$

Method: Master Theorem

To solve this, we use the Master Theorem for divide-and-conquer recurrences:

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

Here, $a = 2$, $b = 2$, and $f(n) = n \log n$.

We compare $f(n)$ with $n^{\log_b a}$:

$$\log_b a = \log_2 2 = 1$$

Since $f(n) = n \log n$ which is $\Theta(n \log n)$, and $n \log n$ is asymptotically equal to $n^{\log_b a} \cdot \log^k n$ where $k = 1$: By case 2 of the Master Theorem:

$$T(n) = \Theta(n \log^2 n)$$

General solution:

$$T(n) = \Theta(n \log^2 n)$$

(d) $T(n) = 3T\left(\frac{n}{2}\right) + n\log n$

Method: Master Theorem

To solve this, we use the Master Theorem for divide-and-conquer recurrences:

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

Here, $a = 3$, $b = 2$, and $f(n) = n\log n$.
We compare $f(n)$ with $n^{\log_b a}$:

$$\log_b a = \log_2 3 \approx 1.584$$

Since $f(n) = n\log n$ which is $O(n^{\log_b a})$, and $n\log n < n^{\log_b a}$: By case 1 of the Master Theorem:

$$T(n) = \Theta(n^{\log_b a})$$

General solution:

$$T(n) = \Theta(n^{\log_2 3})$$