1 Recursion Trees

Draw a recursion tree for the recurrence

$$T(n) = 3T\left(\frac{3n}{4}\right) + cn$$

Draw it at least to the third layer (that is, the grandchildren of the root); each node in the tree should show the local cost of that particular part of the recursion (see Slide Deck 04, slide 44).

Then make an argument (not a formal proof) that the tree will eventually have $\Theta(\log_4 n)$ layers, and that it will have $\Theta(3^{\log_4 3} n)$ leaves.

(My apologies to those who print out their homework; I know that this part will be hard to do on a computer. Draw it by hand, or draw a picture in a Paint program. Or, if you really want to go crazy, write a .dot file to draw it for you.)

Solution:

Since, at every step of recursion, size of the datasets is reduced by a factor of $\frac{4}{3}$, it will take $\log_{\frac{4}{3}} n = \Theta(\log n)$ steps to reach a fixed-size base case. At each layer, the number of elements increases by a factor of $3$, meaning that there are $3^{\log_{\frac{4}{3}} n}$ leaves.

2 Master Method

Solve the following recurrences with the Master Method, if possible. Be clear to show the value of the constants $a, b$. Also identify exactly which case you are using. If a logarithm can be easily simplified (such as $\log_2 4 = 2$), do so; if not (such as $\log_5 7$), you may either convert it to a decimal value, or keep it in logarithm form.

If the recurrence cannot be solved by the Master Method, state why.

(a) 

$$T(n) = 4T\left(\frac{n}{2}\right) + n$$
Solution:

\[ a = 4 \]
\[ b = 2 \]
\[ \log_b a = \log_2 4 = 2 \]
\[ f(n) = n \]

This is Case 1, because \( f(n) = O(n^{2-\epsilon}) \).
\[ T(n) = \Theta(n^2) \]

(b)
\[ T(n) = 3T\left(\frac{n}{3}\right) + n^3 \]

Solution:

\[ a = 3 \]
\[ b = 3 \]
\[ \log_b a = \log_3 3 = 1 \]
\[ f(n) = n^3 \]

This is Case 3, because \( f(n) = \Omega(n^{1+\epsilon}) \).
\[ T(n) = \Theta(n^3) \]

(c)
\[ T(n) = 2T\left(\frac{n}{4}\right) + \sqrt{n} \]

Solution:

\[ a = 2 \]
\[ b = 4 \]
\[ \log_b a = \log_4 2 = \frac{1}{2} \]
\[ f(n) = \sqrt{n} \]

This is Case 2, because \( f(n) = \Theta(n^{1/2}) \).
\[ T(n) = \Theta(\sqrt{n} \log n) \]

(d)
\[ T(n) = 4T\left(\frac{9n}{10}\right) + n^2 \]
Solution:

\[ a = 4 \]
\[ b = \frac{10}{9} \]

\[ \log_b a = \log_{10/9} 4 \approx 13.2 \]
\[ f(n) = n^2 \]

This is Case 3, because \( f(n) = O(n^{13.2-\epsilon}) \).

\[ T(n) = \Theta(n^{13.2}) \]

(e)

\[ T(n) = 4T\left(\frac{10n}{9}\right) + n^2 \]

Solution:

\[ a = 4 \]
\[ b = 9/10 \]

This cannot be solved by the Master Method, because \( b < 1 \).

(f)

\[ T(n) = 2T\left(\frac{n}{2}\right) + n \log n \]

Solution:

\[ a = 2 \]
\[ b = 2 \]

\[ \log_b a = \log_2 2 = 1 \]
\[ f(n) = n \log n \]

This cannot be solved by the Master Method, because \( f(n) = n \log n \) is not polynomially different than \( n \).

(g)

\[ T(n) = 7T\left(\frac{n}{8}\right) + n^2 \log n \]

Solution:

\[ a = 7 \]
\[ b = 8 \]

\[ \log_b a = \log_8 7 \approx .94 \]
\[ f(n) = n^2 \log n \]
This is Case 3, because $f(n) = \Omega(n^{0.94+\epsilon})$.

\[ T(n) = \Theta(n^2 \lg n) \]

(h)

\[ T(n) = 8T\left(\frac{n}{2}\right) + n^2 \lg n \]

Solution:

\[ a = 8 \]
\[ b = 2 \]
\[ \log_b a = \log_2 8 = 3 \]

This is Case 1, because $f(n) = O(n^{0.94+\epsilon})$.

\[ T(n) = \Theta(n^2 \lg n) \]

(i)

\[ T(n) = 2T\left(\frac{n}{2}\right) + \lg n \]

Solution:

\[ a = 2 \]
\[ b = 2 \]
\[ \log_b a = \log_2 2 = 1 \]

This is Case 1, because $f(n) = O(n^{1-\epsilon})$.

\[ T(n) = \Theta(n) \]