The Master Theorem

Let $a \geq 1$ and $b > 1$ be constants, let $f(n)$ be a function, and let $T(n)$ be defined on the nonnegative integers by the recurrence $T(n) = aT(n/b) + f(n)$, where we interpret $n/b$ to mean either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$. Then $T(n)$ has the following asymptotic bounds:

1. If $f(n) \in O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) \in \Theta(n^{\log_b a})$.
2. If $f(n) \in \Theta(n^{\log_b a})$, then $T(n) \in \Theta(n^{\log_b a \log n})$.
3. If $f(n) \in \Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large $n$, then $T(n) \in \Theta(f(n))$.

Transcribed and slightly modified from Cormen, Leiserson, Rivest, and Stein.

The “plus one”

In the preparatory video, we developed a recursive function definition for the running time of merge sort, $T(n) = cn + 2T(n/2)$. We then showed that the function $T(n) = cn \log_2 n$ was compatible with that recursive definition. Using a different analysis technique, we concluded that $T(n) = cn(\log_2 n + 1)$. Show that this new definition of $T(n)$ is also compatible with the recursive definition.

Summation problems

As you analyze recursively defined procedures, you will find that some of the approaches lead you to summations. It will help to have identified the value of some of those summations in advance.

Solve the following summations and sketch a proof for each solution.

1. What is $\sum_{i=0}^{k} 2^i$? That is, what is $1 + 2 + 4 + 8 + ... + 2^k$?
2. What is $\sum_{i=1}^{k} \frac{1}{2^i}$? That is, what is $\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + ... + \frac{1}{2^k}$?
3. What is $\sum_{i=a}^{b} i$? That is, what is $a + (a + 1) + ... + (b - 1) + b$?

If you discover that you need other summations to solve the problems below, you should feel free to look them up.

**Recursive function problems**

Find a closed form of each of the following recursively defined functions, using one of the three manual approaches (recursion trees, top-down expansion, bottom-up exploration) and using the master method. In both cases, you can express your solution using Big-Oh or Big-Theta notation.

1. $T(n) = c + 2T(n/2)$.
2. $T(n) = cn^2 + 2T(n/2)$.
3. $T(n) = cn + 4T(n/4)$.
4. $T(n) = c + 4T(n/2)$. In solving this recurrence, you may want to look at the note about sums of powers at the end of this handout.
5. $T(n) = cn + 4T(n/2)$.
6. $T(n) = cn^2 + 4T(n/2)$.

**An alternate Master Method (optional challenge)**

Some people have tried to “clarify” the master method as follows.

1. If $f(n) \in \Theta(n^c)$ where $c < \log_b a$ then $T(n) \in \Theta(n^{\log_b a})$
2. If $f(n) \in \Theta(n^c)$ where $c = \log_b a$ then $T(n) \in \Theta(n^c \log n)$
3. If $f(n) \in \Theta(n^c)$ where $c > \log_b a$ then $T(n) \in \Theta(f(n))$

What is wrong with this clarification? There are at least two separate issues. Come up with an example of each.

**Sums of powers**

You may find the following sum useful. (The proof is left as an exercise for the reader.)

$$\sum_{i=0}^{i=k} x^i = \frac{x^{k+1} - 1}{x - 1}$$