Recitation R3

Response to 'Muddiest Part of the Recitation Cards'

(10 respondents)

1) *When do we need all the log identities in finding Big-O*?

You don't need all of the log identities at all times. You have to pick and choose the required identity in order to solve/ simplify the recurrence equation. For example, when you have to find the base case, you use $n = 2^k$, then you can represent k as $\log_2 n$

2) *In algorithm classes, do they prove similar forms to the master method for different recurrence equations? Is that the non-simplified master-method*?

What we looked at in class is the solution to recurrence equations of the form:

$$\mathbf{T(n) = aT(n/b) + cn^k}$$

The form that you will see in algorithm classes, treats the second term in the right hand side to be generic i.e.

$$\mathbf{T(n) = aT(n/b) + f(n)}$$

In this case, the master theorem appears as shown below:

$$T(n) = \begin{cases} \Theta\left(n^{\log_b a}\right) & f(n) = O\left(n^{\log_b a - \varepsilon}\right) \\ \Theta\left(n^{\log_b a} \log n\right) & f(n) = \Theta\left(n^{\log_b a}\right) \\ \Theta(f(n)) & f(n) = \Omega\left(n^{\log_b a + \varepsilon}\right) \text{AND} \\ & af(n/b) < cf(n) \text{ for large } n \end{cases} \quad \begin{matrix} \varepsilon > 0 \\ c < 1 \end{matrix}$$

3) *Why cannot I from example 2 be expressed in terms of N*?
Example 2, uses the following code snippet

```
type Int_Array is array (Integer range <>) of Integer;

procedure Measure (A : Int_Array ) is
```

```
   Sum   : Integer := 0;
begin

   for I in A'range loop

      for J in 1 .. I loop –- only change to Ex 1
         Sum := Sum + A(J);
      end loop;

   end loop;

end Measure;
```

The 'I' value in the outer loop changes in every iteration with a maximum value of n. It cannot be expressed as a simple function in n.

4) Still muddy about coming up with T(n) equations from recurrence problems. Specifically unclear on how to solve for O(n) w/o Master method.

When you are not using the master method, use iteration (Lecture 13 last semester) to solve the recurrence equation. If the T(n) is a homogeneous function in n (all the terms are functions of n), then find the most significant term to determine the Big-O. See examples in both the Recitation 3 and Lecture 9 slides.

6) *No mud, cool stuff, good lecture* (6 students)