Problem of the day (a possible recurrence relation for mergeSort):

Assume $n = 2^k$ for some $k \geq 3$. Define $T(n)$ by:

$T(8) = 11$, and for $n \geq 16$, $T(n) = n + 2\ T(n/2)$.

1. What is $T(128)$?

2. What is $T(42)$?

3. Solve this recurrence relation by repeated substitution.
   Note: it's easier to express things in terms of k:

   $T(2^3) = 11$, $T(2^k) = 2^k + 2\ T(2^{k-1})$

Solve it this way then convert things back to a formula that is in terms of n when you are done.

4. Prove your answer is correct by induction.

If you send me e-mail, it would help a lot if you include the course name in your subject header and an informative title.

The two classes I am teaching are both algorithms classes and it will help me to know which one you are asking me about.

For example:
CSC 425: Big Oh notation

Change the title on subsequent messages- it will help ensure I do not accidentally miss any of your questions.

My teaching philosophy:

Goal: Train problem solvers.

Template problem solving has some value in engineering.

Fails as a problem solving tactic for problems not matching template.

To meet exciting real world challenges of computer science: need problem solving skills.

Why recurrences:

Very useful for determining time/space complexity of algorithms.

If you understand methodology for repeated substitution instead of partially guessing, unusual base cases will not throw you off.

Example of same type of thinking as designing recursive algorithms.

Three formulas that often come up:

$$\sum_{i=1}^{n} i$$

$$\sum_{i=0}^{n} 2^i$$

$$\sum_{i=0}^{n} \frac{1}{2^i}$$