CS 5633 Analysis of Algorithms – Spring 05

2/3/05

2. Homework

Due **2/10/05** before class

1. **Guessing and induction (10 points)**

   For each of the following recurrences find a good guess of what it could solve to (make your guess as tight as possible). You can use either the expansion method or the recursion tree method to find your guess. Then prove that $T(n)$ is in big-Oh of your guess by induction (inductive step and base case). Hint: For simplicity you may want to use $\log_4 n$ instead of $\log_2 n$.

   Every recursion below is stated for $n \geq 2$, and the base case is $T(n) = 1$.

   (a) **(5 points)**
   $T(n) = 4T(\frac{n}{4}) + n$

   (b) **(5 points)**
   $T(n) = 4T(\frac{n}{4}) + n^2$

2. **Master theorem (9 points)**
   Use the master theorem to find tight asymptotic bounds for the following recurrences. Justify your results.

   Assume that $T(n)$ is constant for $n \leq 2$.

   - **(3 points)**
     $T(n) = 9T(\frac{n}{3}) + \sqrt{n}$
   - **(3 points)**
     $T(n) = 3T(\frac{n}{2}) + n^2$
   - **(3 points)**
     $T(n) = 16T(\frac{n}{4}) + n^2 \log n$

3. **Quicksort (10 points)**
   Assume for deterministic quicksort we have an oracle which in $\Theta(1)$ time gives us a pivot that guaranteed splits the array into 1/3 and 2/3. (Of course, the partitioning itself still takes $\Theta(n)$ time.

   - What is the resulting runtime recurrence?
   - Can you use the master theorem to solve the recurrence?
   - Show the resulting recursion tree, and argue why this recursion tree resembles a runtime estimate of $\Theta(n \log n)$.
   - Prove by induction that the runtime defined by your recurrence is in $\Theta(n \log n)$.