CS 3343 Analysis of Algorithms – Spring 06

3. Homework

Due: Tuesday 2/21/06 before class, or **Excuse:** Thursday 2/23/06 before class

1. **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 $n \leq 2$.

   - $T(n) = 32T(\frac{n}{2}) + n$

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

   - **(3 points)**
     $T(n) = 16T(\frac{n}{4}) + n^2 \log^5 n$

   - $T(n) = T(\frac{7n}{8}) + n$

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

2. **Divide and Conquer**
   Let $A[1..n]$ be an array of n numbers. A number in A is a *majority element* if A contains this number more than $\lfloor n/2 \rfloor$ times.

   Write a divide-and-conquer algorithm that determines whether a given array $A[1..n]$ contains a majority element, and if so, returns it. Your algorithm should run in $O(n \log n)$ time. You are **not** allowed to sort the array.

   Set up and solve a recurrence relation for the runtime of your algorithm.

   Hint: Start by applying the generic divide-and-conquer approach. Try to divide by two. Then try to combine the results. From the given runtime you should be able to guess how much time you are allowed to spend for dividing and combining.