CS 312: Algorithms
Homework 6

Dan Sheldon

March 14, 2013

Instructions

Complete all problems and submit by Wednesday, March 27. You may work together with other students, but *your written work must be your own.* I highly encourage you to attempt the problems first on your own, especially the simpler ones.

Please make sure to:

- Write your name on your submission
- Write the name of all students with whom you collaborated
- Cite any sources you used other than the textbook or course notes.

Problems

1. (12 points) Consider an algorithm whose running time $T(n)$ on an input of size n satisfies the following recurrence:

$$T(n) \leq aT(n/b) + cn,$$

where we assume the recurrence holds when $n \geq 2$, and that $T(2) \leq c$.

   (a) (2 points) How many subproblems are there at level i of the recursion tree?

   (b) (2 points) How big are the subproblems at level i of the recursion tree?

   (c) (2 points) What is the total work done at level i of the recursion tree? (Just as in class, count only the work done outside of the recursive calls.)

   (d) (2 points) How many levels are in the recursion tree?

   (e) (2 points) If $a < b$, what is the running time of the algorithm? Give your answer in big-O form.

   (f) (2 points) If $a = b$, what is the running time of the algorithm? Give your answer in big-O form.

   Hint: remember the following fact that we showed about a geometric sum when $0 < r < 1$:

$$\sum_{i=0}^{d} r^d = 1 + r + r^2 \ldots + r^d = \frac{1 - r^{d+1}}{1 - r} \leq \frac{1}{1 - r}$$

2. (8 points) Consider the following recurrence that we saw in our first cut at the closest pair algorithm:

$$T(n) \leq 2T(n/2) + cn \log n,$$

   where we again assume the recurrence holds for $n \geq 2$ and that $T(2) \leq c$. Prove by induction that $T(n) \leq cn(\log n)^2$. (Another way to say this is to say that $T(n)$ is $O(n \log^2 n)$.) You should assume that the logarithm is base 2, so that $\log(n/2) = \log n - 1$.

3. (10 points) Chapter 5, Exercise 1
4. (Piazza) Chapter 5, Exercise 5