CS381 Homework 1

Out: Aug. 31, Thursday
Due: Sep. 14, Thursday, in (or before) class
Submissions will not be accepted afterwards.

Instructions:

Read the course policy (including academic dishonesty) in the course webpage at `http://www.cs.purdue.edu/homes/gopal/cs381`. Text refers to the Introduction to Algorithms (second edition) book. **Justify your answers. Show appropriate work.**

Reading: Chapter 3, Chapter 4, and Appendix A of Text.

Problem 1

Prove by induction the following statements:

1. $\sum_{i=0}^{n} a^i = \frac{a^{n+1}-1}{a-1}$ where $a \neq 1$ is some fixed real number.

(By the way, this is the sum of a geometric series, a useful formula to remember.)

2. $x^n - y^n$ is divisible by $x - y$ for all natural numbers x, y ($x \neq y$), and n. (Hint: You will need strong induction.)

3. The famous Four Color Theorem states that every map can be colored by using four colors. Incidently, this was the first major mathematical theorem that was shown with the help of a computer and its proof is not easy. Your goal here is something much more modest. Consider n distinct (infinite) lines on a plane. We are interested in assigning colors to the regions formed by these lines such that neighboring regions have different colors. (Two regions are considered neighbors if and only if they have an edge in

common.) Show that it is possible to color the regions formed by n lines in the plane with only two colors. (Hint: You may need to alter colorings in the induction step!)

Note that to get full credit on induction proofs you should state the induction variable (usually n, but not always), prove the base cases, set up the induction hypothesis clearly, and show where you use the hypothesis in the induction step.

Problem 2

Rank the following functions by order of growth; that is, find an arrangement $g_1, g_2, g_3, \ldots$ of the functions satisfying $g_1 = O(g_2)$, $g_2 = O(g_3), \ldots$.

$$n^2, n/\log n, n \log n, (1.001)^n, 1/n^2, \log^{100} n, F_n, 1/\log n,\ 4^{\lg n}, n!, n^{\lg \lg n}.$$

(lg means logarithm to the base 2. F_n is the nth Fibonacci number.)

Problem 3

Prove the asymptotic bound for the following recurrences by using induction. Assume that base cases of all the recurrences are constants i.e., $T(n) = \Theta(1)$, for $n < c$ where c is some constant.

1. $T(n) \leq 2T(n/2) + n^2$. Then, $T(n) = O(n^2 \log n)$.
2. $T(n) \leq T(5n/6) + O(n)$. Then, $T(n) = O(n)$.
3. $T(n) = T(5n/6) + T(n/6) + n$. Then, $T(n) = O(n \log n)$.

Problem 4

Solve the following recurrences and give the result exactly (i.e., without using the big-oh notation). Assume that n is a power of 2 for both problems. (Hint: Use the recursion tree method.)

a) $T(1) = 1$ and for $n > 1$, $T(n) = T(n/2) + n$.

b) $T(1) = 1$ and for $n > 1$, $T(n) = 2T(n/2) + n$.

Problem 5

We are given a *sequence* of n positive numbers $a_1, \ldots, a_n$ and a fixed number $k > 0$. We want to find a pair of numbers a_i and a_j such that $j > i$ and $a_j - a_i = k$. If there is no such pair we want to output “false”. This can be solved easily in $\Theta(n^2)$ time by examining all pairs of numbers. Your task is to give an algorithm that runs in time $O(n \log n)$. Analyze the running time of your algorithm. (Hint: Merge Sort can prove helpful.)