CS 5150/6150: Assignment 6
Due: Nov 23, 2011

This assignment has 5 questions, for a total of 100 points and 0 bonus points. Unless otherwise specified, complete and reasoned arguments will be expected for all answers.

Question 1: Pebbling a graph . . . . . **[20]**

Solve Question 13 from `http://compgeom.cs.uiuc.edu/~jeffe/teaching/algorithms/notes/29-nphard.pdf`

Question 2: Solving SAT . . . . . **[20]**

Solve Question 4 from `http://compgeom.cs.uiuc.edu/~jeffe/teaching/algorithms/notes/29-nphard.pdf`. The breakdown of points is 12 + 8. Please be precise in part (b) - calculations matter !

Question 3: Binary Search . . . . . **[20]**

(a) [**10**] I'm given a collection of numbers $X = \{x_1, \ldots, x_n\}$ that all lie in $[0, 1]$. I wish to build an *approximate* search structure on these numbers. Specifically, for a query $y \in [0, 1]$ I wish to return some $x \in X$ such that $|x - y| \leq \epsilon$, or `NONE` if no such number exists.

Give an algorithm that can process a query correctly in time $O(\log \frac{1}{\epsilon})$. You may preprocess the input as appropriate in a reasonable amount of time.

(b) [**10**] In the previous example, my search incurred an *absolute* error. But suppose I want *relative* error ? Assume now that I'm given a collection X of n numbers in the range $[1..M]$. Given a query y, I'd like to return an answer $x \in X$ such that $x \leq y \leq x(1 + \epsilon)$ or `NONE` if no such point exists. Given an algorithm that can process such a query in time $O(\log \log M + f(1/\epsilon))$, for some function f. Again, you may preprocess the input as you see fit.

Question 4: Partition . . . . . **[20]**

Solve Question 5 from `http://compgeom.cs.uiuc.edu/~jeffe/teaching/algorithms/notes/30-approx.pdf`

Question 5: Approximate Graph Coloring . . . . . **[20]**

Solve Question 6 from `http://compgeom.cs.uiuc.edu/~jeffe/teaching/algorithms/notes/30-approx.pdf`. **HINT:** You need to modify the input graph in some manner before presenting it to the claimed approximation algorithm.