22C:31 Homework 2

Due in class on Tuesday, September 29th

1. Recall the "first-fit" greedy algorithm for the INTERVAL PARTITIONING problem (discussed on pages 122-125 in the textbook). In class, I stated this algorithm as follows:

   (a) Consider the intervals in left-to-right ordering of start time.

   (b) For each interval considered in this order, assign the smallest color from $\{1, 2, 3, \ldots\}$ that is available.

   A slight variant of this algorithm, that works as well (check this!) is the following:

   (a) Initialize `max_color` $\leftarrow 0$.

   (b) Consider the intervals in left-to-right ordering of start time.

   (c) For each interval a considered in this order, if there is an available color in the range [1..`max_color`], pick an arbitrary available color and assign it to a; otherwise, increment `max_color` and assign `max_color` to a.

   For this problem, I want you to describe an implementation of this variant that runs in $O(n \log n)$ time, where n is the number of intervals in the input. Step (2) is just sorting and therefore it can be implemented in $O(n \log n)$ time. One approach to implementing Step (3) in $O(n \log n)$ time, is to maintain a data structure for the colors currently in use that can perform the following operations in $O(\log n)$ time:

   (a) Answer the `SEARCH` query: is there a color available for interval a?

   (b) Perform an `INSERT`: Add a new color, which is currently being only used for the interval a.

   Describe such a data structure and the information that is maintained in it. Describe how the two operations listed above are performed on this data structure and why each of them runs in $O(\log n)$ time in the worst case. Finally, put everything together to show that your complete algorithm runs in $O(n \log n)$ time.

   This may seem like a lot of work, but it is not, because the data structure I am referring to, should be quite familiar to you. And so your answer can simply appeal to known facts about this data structure.

2. Solve the BASE STATION PLACEMENT problem, i.e., Problem 5, Page 190.

3. Consider this 2-dimensional version of the BASE STATION PLACEMENT problem. You are given a set P of n points in the plane. You can locate a cell phone base station at each point $p \in P$. A base station located at p is said to *cover* all points in P that are within 1 unit of distance from p. This basically means that the transmission range of each base station is a unit disk (i.e., disk of radius one) centered at that base station.

   (a) State a greedy algorithm for solving the problem of finding a minimum set of base stations that cover all points in P. Whatever greedy algorithm you state should "pass" the basic sanity check that if there is an optimal solution to the problem with one base station, then your algorithm should find it.

   (b) Find a simple counterexample for this greedy algorithm.

4. Problem 7, Pages 191-192 on how a search engine should schedule the parallel computation of a search index.

5. Problem 12, Pages 193-194.
6. Problem 19, Pages 198-199.
7. Problem 24, Pages 200-201.
