Problem Set 3

1. Recall the CLASS SCHEDULING algorithm discussed in the class. Consider the following algorithm for the same problem. Prove or disprove its correctness.

   Let x be the class with the earliest start time, and let y be the class with the second earliest start time.

   - If x and y are disjoint, choose x and recurse on everything but x.
   - If x completely contains y, discard x and recurse.
   - Otherwise, discard y and recurse.

2. Now consider a weighted version of the class scheduling problem, where different classes offer different number of credit hours (totally unrelated to the duration of the class lectures). Your goal is now to choose a set of non-conflicting classes that give you the largest possible number of credit hours, given arrays of start times, end times, and credit hours as input.

   (a) Prove that the greedy algorithm described in the classChoose the class that ends first and recursedoes not always return an optimal schedule.

   (b) Describe and analyze an polynomial time algorithm that always computes an optimal schedule.

3. Let X be a set of n intervals on the real line. We say that a subset of intervals $Y \subseteq X$ covers X if the union of all intervals in Y is equal to the union of all intervals in X. The size of a cover is just the number of intervals. Describe and analyze an efficient algorithm to compute the smallest cover of X. Assume that your input consists of two arrays $L[1 \ldots n]$ and $R[1 \ldots n]$, representing the left and right endpoints of the intervals in X. If you use a greedy algorithm, you must prove that it is correct.

4. Describe in detail how to implement the Gale-Shapley stable matching algorithm (discussed in the class), so that the worst-case running time is $\mathcal{O}(n^2)$.

5. Consider a generalization of the stable matching problem, where some doctors do not rank all hospitals and some hospitals do not rank all doctors, and a doctor can be assigned to a hospital only if each appears in the others preference list. In this case, there are three additional unstable situations:

   - A matched hospital prefers an unmatched doctor to its assigned match.

- A matched doctor prefers an unmatched hospital to her assigned match.
- An unmatched doctor and an unmatched hospital appear in each others preference lists.

A stable matching in this setting may leave some doctors and/or hospitals unmatched, even though their preference lists are non-empty. For example, if every doctor lists AIIMS Delhi as their only acceptable hospital, and every hospital lists Dr. Arjun as their only acceptable intern, then only Arjun and AIIMS Delhi will be matched. Describe and analyze an efficient algorithm that computes a stable matching in this more general setting.

6. Suppose we are given two arrays $C[1 \ldots n]$ and $R[1 \ldots n]$ of positive integers. An $n \times n$ matrix of 0s and 1s *agrees* with R and C if, for every index i, the ith row contains $R[i]$ 1s, and the ith column contains $C[i]$ 1s. Describe and analyze an algorithm that either constructs a matrix that agrees with R and C, or correctly reports that no such matrix exists.