ASSIGNMENT 3

DUE: Wednesday October 2, 5 PM. DO NOT COPY. ACKNOWLEDGE YOUR SOURCES.

Please read http://www.student.cs.uwaterloo.ca/~cs341 for general instructions and policies. When you are asked to design an algorithm you must: (1) describe the idea of your algorithm clearly in English; (2) give pseudocode; (3) give a proof of correctness; and (4) analyze the run time.

1. Does greed bring happiness? [6 marks]

The happiest interval scheduling problem is defined as follows: Given a set of $n$ intervals defined by start and finish times $(s_1, f_1), \ldots, (s_n, f_n)$ and $n$ positive integers $h_1, \ldots, h_n$ denoting the happiness value of each interval, find a subset $I \subseteq \{1, 2, \ldots, n\}$ of disjoint intervals that maximizes $\sum_{i \in I} h_i$.

Determine which of the following greedy algorithms correctly solves the happiest interval scheduling problem. Justify each answer with a proof of correctness or a counter-example.

(a) Earliest finish time. Choose the interval $i$ with the earliest finish time, discard all intervals that overlap with $i$, and recurse.

(b) Highest happiness per minute. Choose the interval $i$ with the maximum happiness per minute ratio $\frac{h_i}{f_i-s_i}$, discard all intervals that overlap with $i$, and recurse.

(c) Kondo’s algorithm. If no intervals overlap, select them all. Otherwise, discard the interval $i$ with the smallest happiness value $h_i$.

2. Fire hydrant placement [8 marks]

There are $n$ houses along a long straight road. You must place fire hydrants along the road so that every house is within 90 meters of a hydrant. Model the road as a line and the houses as points $p_1, p_2, \ldots, p_n$ along the line, given in sorted order.

Design a greedy algorithm to find the minimum number of fire hydrants needed and their position on the road.

3. Agreeable matrices [10 marks]

Suppose we are given two arrays $C[1..n]$ and $R[1..n]$ of positive integers. An $n \times n$ matrix $M$ of non-negative integers agrees with $R$ and $C$ if for every index $i$, the sum of the entries in the $i$th row of $M$ equals $R[i]$ and the sum of the $i$th column of $M$ equals $C[i]$.

Design an algorithm that outputs a matrix $M$ that agrees with $R$ and $C$ when such a matrix exists and correctly reports that no such matrix exists otherwise.