McGill University COMP360 Winter 2011 **Instructor: Phuong Nguyen**

Assignment 4

Due February 7 at the beginning of lecture

The work you submit must be your own. You may discuss problems with each others; however, you should prepare written solutions alone. Copying assignments is a serious academic offense, and will be dealt with accordingly.

Question 1 (10pt) Consider the following variant of the problem Interval Scheduling, which we call MIS (M for multiple). In this problem, each request consists of a *set* of intervals (instead of just one interval as in Interval Scheduling). There is a single processor that can process only one request at a time. Thus, if we accept a request R then we cannot accept any other request that contains some interval which overlaps one of the intervals in R. For example, suppose that there are 4 requests:

$$R_1 = \{(1,3),(5,6),(7,8)\}$$
$$R_2 = \{(2,4)\}$$
$$R_3 = \{(4,6),(8,9)\}$$
$$R_4 = \{(3,4),(8,9)\}$$

Then the maximum number of requests that can be scheduled together is 2, e.g.: R_1 and R_4, or R_2 and R_3. The MIS problem is as follows.

Input:

- A set of n requests $R_1, R_2, \ldots, R_n$, each request R_i is specified by a list of intervals

  $$R_i = (s^i_1, f^i_1), (s^i_2, f^i_2), \ldots, (s^i_{m_i}, f^i_{m_i})$$

  where all s^i_j, f^i_j are nonnegative integers.

- An integer k

(Representation of numbers are not very important here, for example, they are written in unary.)

Output: Accept if we can schedule at least k requests together.

Show that the MIS problem is NP-complete, by giving a nondeterministic polytime algorithm for it, and many-one reducing CLIQUE to it. Give formal proof of correctness for your reduction.

Question 2 (4pt) Consider the following greedy algorithm that attempts to solve the Knapsack problem (see Assignment 3). Informally, the idea is to always select elements with highest possible ratio value/weights. Then output Yes if the selected elements have total value at least V.

(a) (2pt) Give pseudo-code for the algorithm.

(b) (2pt) Exhibit an example where the algorithm fails to solve Knapsack. Clearly describe why the algorithm fails on your example.

Question 3 (6pt) In class we discussed an greedy algorithm for Interval Scheduling that works by always selecting requests that finish as early as possible. Another greedy algorithm that also produces an optimal solution is to always select requests that start as late as possible.

(a) (2pt) Give pseudo-code for the algorithm.

(b) (4pt) Proof that the algorithm always output an optimal schedule.