Algorithms
Computer Science 140 & Mathematics 168
Instructor: B. Thom
Spring 2004
Homework 7a
Due on 3/4/03 (in class)

1. [**25 Points**] **Greed Must be used with Extreme Caution!** In class we talked about the problem of finding the maximum number of non-conflicting courses from a given set of courses. We argued that if the courses are sorted in order of completion time, then the following greedy algorithm is guaranteed to find the maximum number of non-conflicting courses: Choose the course with earliest completion time. Cancel out all courses that conflict with that course. Now repeat the process for the remaining courses.

   Professor I. Lai of the Pasadena Institute of Technology has proposed the following four alternative greedy algorithms for this problem.

   Algorithm 1: Sort the courses by ending time as before. Now, run our original greedy algorithm in the opposite direction. That is, choose the course that ends **latest**. Then cancel out all courses that conflict with that course. Now repeat the process for the remaining courses.

   Algorithm 2: Sort the courses by increasing starting time (rather than ending time). Now, choose the course that starts first. Then cancel out all courses that conflict with that course. Now repeat the process for the remaining courses.

   Algorithm 3: Forget about sorting the courses. Choose a course of shortest duration (that is the course that has the least length). Then cancel out all courses that conflict with that course. Now repeat the process for the remaining courses.

   Algorithm 4: Don't sort the courses. Choose a course that conflicts with the fewest other courses,*breaking ties arbitrarily.* Then cancel out the courses that conflict with that course. Now repeat the process for the remaining courses.

   In this problem you will explore various aspects of greed in the context of the Registrar problem.

   (a) To show that an algorithm is incorrect—e.g. in the case of greed, does not always find an optimal solution—one only needs to demonstrate a specific example in which the algorithm might select a solution that is suboptimal. Explain in one or two sentences why this is sufficient to demonstrate an algorithm is incorrect.

   (b) Show that each of the algorithms is not correct by providing a counterexample. Be sure to indicate both what the greedy algorithm's suboptimal solution would be and what a better solution is.

   (c) Consider a registrar algorithm that first sorts on start time and then uses the following greedy choice: choose the course with the latest start time (remove conflicts and repeat). Using a procedure similar to the one we saw in class, argue that this choice is correct. Take care to state your argument clearly and precisely (don't skip any steps).

2. **[10 Points Extra Credit! (due Fri, 6pm)] An Accomplice for Greed: Eliminating Conflicts Quickly**

   In class I mentioned that there is a scheme for eliminating conflicts in `Greedy_Registrar` that costs $\Theta(n)$ overall. In this problem, you will describe a way of managing the data so that this can be done. In solving this problem, you can assume that in advance the data has been sorted some constant number of times. However, if you rely on sorted data in handling linear conflict resolution, be sure to describe what sorting you're assuming exists. Describe your data handling scheme and argue that its run-time is linear over the entire algorithm. You should use the following notation: the original input is $A = \{a_1, \cdots, a_n\}$, which contains n items. Item a_i refers to activity i (in the original input, the i activities are not necessarily sorted). Each activity has two auxiliary fields: s_i, an activity's start time, and e_i, its end time.