CSE 5311 Lab Assignment 2

Due July 18, 2007

Goal:

Application of union-find trees.

Requirements:

1. Write (and test) a C/C++ program that implements Algorithm 1 (maximum cardinality k-coloring of a set of intervals) in the attached paper in $O(n \log n)$ time. This is a generalization of the activity-selection problem in CLRS 16.1.
   The input is to be read from standard input and will be entered either from the terminal or by using a shell redirect. The first line of the input will be k and n (with $k < n$), the number of available colors and the number of (integer) intervals in the remaining n input lines. Each of the intervals $[x, y)$ will have $0 < x < y$.
   The first line of the output will be the number of intervals successfully colored and each of the remaining n lines will contain an interval and a color (in the range $1 \ldots k$ if successfully colored and 0 otherwise). The output ordering of intervals is not required to be the same as the input ordering.
   Your output must follow this format since the results will be checked by a program.
   Your program must compile and execute on OMEGA. There should be a comment near the beginning of your code indicating how to compile on OMEGA.

2. Email your code (as attachments) to `spal@uta.edu` before 10:15 am on July 18. The subject should include your name as recorded by the University.

Getting Started:

1. Only the sorts should take $O(n \log n)$ time (you may use `qsort`). The rest of the processing, including the loop to initialize `adjacent[]`, should take $O(n)$ time (under the assumption that *union* and *find* take $O(1)$ amortized time by the use of an efficient *union* along with path compression on *find*s).

2. The paper treats the "leader" (e.g. the root) and "name" of a disjoint set as being the same thing. Your code will be cleaner (and theoretically faster) if you distinguish these concepts, i.e. a disjoint set may have a name other than its leader.

3. Arrays should be dynamically allocated.