Data Structures and Algorithms
Pennsylvania State University
Professor Paul Medvedev

October 10, 2015
CMPSC 465

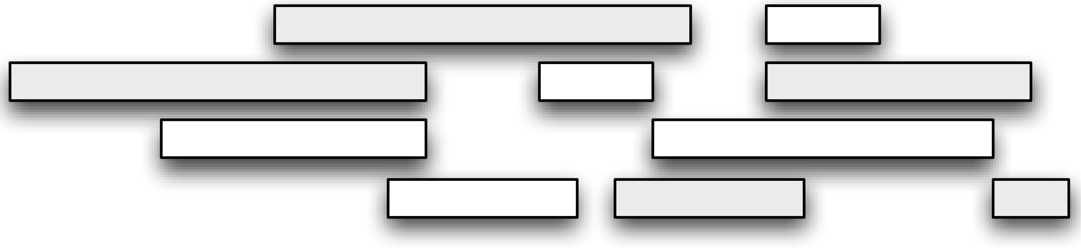
Homework 5 – due Oct 19th, 4:40pm before class

Reminders:

- You may work either alone or in teams of two. If working in pairs, you should hand-in a single assignment, with both of your names on it.
- The solution must be typed (Word is recommended). No handwritten work will be accepted or graded (though you are allowed to draw an illustration for Problem 2(a) if you feel it helps).
- Remember to use the header page provided on Angel.
- Late homework will not be accepted, but you are allowed to drop one lowest homework grade at the end of the semester.
- You are not allowed to discuss any aspects of the homework solutions problem with anyone except your partner. Finding answers to problems on the Web or from other outside sources (these include anyone not enrolled in the class) is strictly forbidden.

Problems:

1. Suppose that instead of always selecting the first activity to finish, we instead select the last activity to start that is compatible with all previously selected activities. Prove that this greedy algorithm always yields an optimal solution.
2. Given a family of intervals $[a_i, b_i]$, $i = 1, ..., n$, a *sub-cover* is subset of the intervals that covers the same area of the real line as the union of all the intervals. For example, in the following picture, the gray intervals form a sub-cover consisting of 5 intervals:



(Warm-up exercise, not to hand in: what is the smallest subcover for this family?)

You want to design an efficient algorithm that finds a sub-cover with as few intervals as possible. In the following, you may assume that (i) all the endpoints (a_i, b_i) are distinct, and (ii) the original set of intervals covers a contiguous segment of the real line.

(a) Your friend suggests the following greedy approach: at each stage, add the interval with the most length not covered by the intervals selected so far. Give an example showing that this approach will not find the smallest sub-cover.

(b) Give a polynomial-time algorithm to find a smallest sub-cover, given the a_i's and b_i's as input.

(c) What is the running time of *your* algorithm?

(d) Prove that the greedy algorithm is correct.