CS 499 (Fall 2006)- Assignment 3
Due: Tuesday, 09/12/2006

(1) In assigning computers to wireless routers, one frequently needs to do some careful load balancing. Due to distance/signal strength constraints, not every computer can connect to every router. On the other hand, routers have capacity constraints.

Here is one way to model this problem: you are given the positions (x_i, y_i) for each router i. You are also given its radius r_i, and its capacity c_i. The radius says at what distance computers can connect to this router: any computer at distance r_i or less from router i can in principle connect to it. The capacity says how many computers at most can connect to this router at once. Of course, the capacities and radii can be different for different routers. For each computer, you are given its location as (x'_j, y'_j).

Your goal is to write a program that assigns as many computers as possible to routers. Notice that sometimes, not all computers can be assigned to routers simultaneously, for instance when all routers are too far from the computers, or the total capacity of the routers is not enough to serve everyone. But you are to find a solution that assigns as many computers as possible.

(2)

1. What aspects, if any, of this real-world problem have we ignored? How much more difficult would the problem get if we tried to incorporate them?
2. Suppose that instead of using the excellent program that you just wrote, we have the following process: the computer users arrive one after the other at their locations (x'_j, y'_j), in some order that we cannot influence. Each user, as soon as he/she arrives, connects to the closest router that (a) they can connect to, and (b) still has capacity available. If there is no such router, the user goes unserved.

   It won't be too difficult for you to construct examples and arrival orders where this approach serves fewer users than would be possible with your program. What's the worst example you can come up with? In other words, what is the worst *ratio* between the number of users served by your (optimal) program, and this FIFO-closest assignment rule? Show what the example looks like with a drawing or such.