Basic Algorithms, Assignment 8

Due, Thursday, Nov 8

SINKING SHIP? Monday, November 5, 2007 is the LAST DAY for undergraduate students to withdraw from a course.
HONORS PROJECT: To receive Honors Credit students must complete an honors project. Please contact Prof. Spencer for possible projects.

1. Page 189, Problem 2
   Solution: For (a) this is true. MST has the amazing property that the tree itself depends only on the order of the costs of the edges. (In both Prim and Kruskal one never calculates with the actual costs themselves.) Squaring the costs doesn't change the order of the cost and so we get the same tree. But, turning to (b), that is not the case with Dijkstra's Algorithm. Indeed, (b) is false. Here is a simple counterexample. Consider a sourse s, a vertex (sometimes called the sink) t and internal vertex a. Suppose we have a path sat and an edge st. We have no other edges and so these are the only choice for an $s-t$-path. Suppose $c[s,t] = 100$ and $c[s,a] = 51 = c[a,t]$. The path sat has cost 101 so the minimimal $s-t$ path is the edge st with cost 100. When we square everything the costs are now $10000, 2601$. But now it is cheaper to take the two edges (cost 5202) than the one at cost 10000.

2. `Computer Experiment:` Implement either Dijkstra's Algorithm or Prim's Algorithm or Kruskal's Algorithm. You cannot use a canned program but must write it from scratch in the language of your choice. (It is OK if stacks and arrays are built in.) Apply it to the following random data: The graph is the complete graph on n vertices (for Dijkstra we assume an undirected graph) and the costs $c[i,j]$ are independent uniformly random (use standard code for this) real numbers in $[0,1]$. (For Kruskal you can use a canned sort to order the costs.) For Kruskal or Prim let T be the total length of the tree created. For Dijkstra, let vertex 1 be the source and let T be the number of times $d[2]$ changes during the algorithm. Run the algorithm several times for $n = 50, 100, 200, 500$. Plot the average value of T as a function of n. From the data (taking more data if you wish) come up with a conjecture of how T behaves as a function of n. `Notes:` This can be done in groups of at most three students and handed in jointly. You **MUST** give copious explanations, either in comments on the program itself or on a side sheet, of what you are doing.

Comment: The study of T for MST has been quite extensive. The surprising result is that T is tightly concentrated around $\pi^2/6$ when n is large.

Among his co-workers in an Indian named Ganapathy. Ganapathy often arrives late to work; on some days he does not come at all. When he does come, he does not appear to be working very hard; he sits in his cubicle with his feet on the desk, apparently dreaming. For his absences he has only the most cursory of excused ("I was not well") Nevertheless he is not chided. Ganapathy, it emerges, is a particularly valuable aquisition for International Computers. He has studied in America, holds an American degree in computer science.
J.M. Coetzee, *Youth*