Network Optimization (Fall 2008) – Project

- You should work on the project by yourself.
- Do **NOT** include your MATLAB codes in the printed report.
- The total points (given in parentheses) add up to 150.
- **This project is due in class on Thursday, November 13**.

1. (20) Show that the running time of Dial's implementation of Dijkstra's algorithm (with buckets for storing nodes with same distance label values) can be improved to $O(m+nC/c)$ where $c = \min_{(i,j)\in A} c_{ij}$.

2. **Average running times of shortest-path algorithms.**
   We have seen the following algorithms for solving the shortest path problem (also listed are their worst-case running times):

   - Dijkstra's algorithm, default implementation – $O(n^2)$;
   - Dial's implementation of Dijkstra's algorithm – $O(m + nC/c)$
   - FIFO label correcting algorithm – $O(nm)$;

   Our goal here is to compare the running times of these algorithms in practice to their worst-case running times. You should have already implemented two of the three algorithms as part of previous homeworks (if not, then do so now!).

   (a) (30) Write a code for the Dial's implementation of Dijkstra's algorithm.

   (b) (100) Describe a mechanism to generate random networks having n nodes, m arcs, and each arc having integer cost randomly chosen from the interval $[1, C]$ for a given parameter C. (*Hint:* One idea to assign the network structure randomly is to choose the entries of the node-arc or the node-node incidence matrix "randomly").

   You need to get average running times taken by the three algorithms for solving random instances. For each set of (n, m, c, C) values, generate at least five different (random) networks, and record the running time taken by each algorithm. To ensure uniformity, run the three algorithms on the same instances (i.e., do not generate *new* networks for each algorithm, even if using the same parameters). Vary each parameter – n, m, c, and C – across appropriate ranges, and tabulate the observed average running times for each algorithm, showing a comparison with the worst-case running time. If the running time is $O(f(n, m, c, C))$, in order to estimate the worst-case running time, just calculate $f(n, m, c, C)$. Briefly summarize your findings. Would you prefer using one of the algorithms over the other two in most cases?

   Once you are sure that your codes for each algorithm work properly, you need not print (or display) the solution for each network instance considered. To record the running time in MATLAB, you can use the commands `tic` and `toc`, or `etime` and `clock`, or using `cputime` – see help on these commands in MATLAB. In fact, you may want to generate plots of running times against various parameters considered.