Design and Analysis of Algorithms
(CSCE 423/823, Spring 2005)
URL: `http://www.cse.unl.edu/~vinod/823S05/index.html`
23rd Feb, 2005

Homework 3

Due on 4th Mar, 2005

Name :

Student ID :

Course No : **423** **823**

Instructions

- Write your name and student ID on the space provided. Circle the course number in which you are registered. Return this sheet with your solutions.
- Clarity of presentation is utmost important. You should give a clear description of all your algorithms with an analysis of their time complexity.
- No collaboration with any one is permitted for CORE problems!. See the Instructor or the TA if you need additional guidance. You may discuss approaches to solving ADVANCED problems with your classmates. However, you must write up your own solutions in your own words and acknowledge any one with whom you collaborated.

Core Problems

1. (5 Points) Problem 23.2-2 from the text book (page number 573). Suppose that the graph $G = (V, E)$ is represented as an adjacency matrix. Give a simple implementation of Prim's algorithm for this case that runs in $O(V^2)$ time.

2. (5 Points) Problem 24.1-1 from the text book (page number 591). Run Bellman-Ford algorithm on the directed graph of figure 24.4 (page number 589), using vertex z as the source. In each pass, relax edges in the same order as in the figure, and show the d and Π values after each pass.

3. (2.5 Points) Problem 24.2-1 from the text book (page number 594). Run DAG-SHORTEST-PATHS on the directed graph of Figure 24.5 (page number 593) using vertex r as the source.

4. (2.5 Points) Problem 24.3-1 from the text book (page number 600). Run Dijkstra's algorithm on the directed graph of Figure 24.2 (page number 585), first using vertex s as the source and then using vertex z as the source. In the style of Figure 24.6 (page number 596), show the d and Π values and the vertices in set S after each iteration of the **while** loop.

5. (10 Points) Problem 24.3-4 from the text book (page number 600). We are given a directed graph $G = (V, E)$ on which each edge $(u, v) \in E$ has an associated value $r(u, v)$, which is a real number in the range $0 \leq r(u, v) \leq 1$ that represents the reliability of a communication channel from vertex u to vertex v. We interpret $r(u, v)$ as the probability

that the channel from u to v will not fail, and we assume that these probabilities are independent. Give an efficient algorithm to find the most reliable path between two given vertices.

6. (10 Points) Problem 25.2-7 from the text book (page number 635). Another way to reconstruct shortest paths in Floyd-Warshall algorithm uses values ϕ_{ij}^k for $i, j, k = 1, 2, \ldots, n$, where ϕ_{ij}^k is the highest-numbered intermediate vertex of a shortest path from i to j in which all intermediate vertices are in the set $\{1, 2, \ldots, k\}$. Give a recursive formulation for ϕ_{ij}^k, modify the FLOYD-WARSHALL procedure to compute the ϕ_{ij}^k values.

Advanced Problems

7. **Arbitrage** (5 Points; Problem 24-3 of the text; page 615). Arbitrage is the use of discrepancies in currency exchange rates to transform one unit of a currency into more than one unit of the same currency. For example, suppose that 1 U.S. dollar buys 46.4 Indian Rupees, 1 Indian rupee buys 2.5 Japanese yen, and 1 Japanese yen buys 0.0091 U.S. dollars. Then by converting currencies, a trader can start with 1 U.S. dollar and buy $46.4 \times 2.5 \times 0.0091 = 1.0556$ U.S. dollars, thus turning a profit of 5.56 percent.

   Suppose that we are given n currencies $c_1, c_2, \ldots c_n$ and $n \times n$ table R of exchange rates, such that one unit of currency c_i buys $R[i, j]$ units of currency c_j.

   Give an efficient algorithm to determine whether or not there exists a sequence of currencies $\langle c_{i_1}, c_{i_2}, \ldots, c_{i_k} \rangle$ such that

   $R[i_1, i_2] \cdot R[i_2, i_3] \cdots R[i_{k-1}, i_k] \cdot R[i_k, i_1] > 1.$

   Analyze the running time of your algorithm.

8. **Second-best minimum spanning tree** (10 Points; Problem 23-1 of the text; page 575). Let $G = (V, E)$ be an undirected, connected graph with weight function $w : E \rightarrow \mathbf{R}$, and suppose $|E| \geq |V|$ and all edges weights are distinct.

   A second-best minimum spanning tree is defined as follows. Let τ be the set of all spanning trees of G, and let T' be a minimum spanning tree of G. Then a second-best minimum spanning tree is a spanning tree T such that $w(T)$ =$\min_{T^n \in \tau - \{T'\}}\{w(T'')\}$.

   (a) Show that the minimum spanning tree is unique, but that the second-best minimum spanning tree need not be unique.

   (b) Let T be a minimum spanning tree of G. Prove that there exists edges $(u, v) \in T$ and (x, y) does not belong to T such that $T - \{(u, v)\} \bigcup \{(x, y)\}$ is a second-best minimum spanning tree of G.

   (c) Let T be a spanning tree of G and, for any two vertices $u, v \in V$, let $max[u, v]$ be an edge of maximum weight on the unique path between u and v in T. Describe as $O(V^2)$-time algorithm that, given T, computes $max[u, v]$ for all $u, v \in V$.

   (d) Give an efficient algorithm to compute the second-best minimum spanning tree of G.