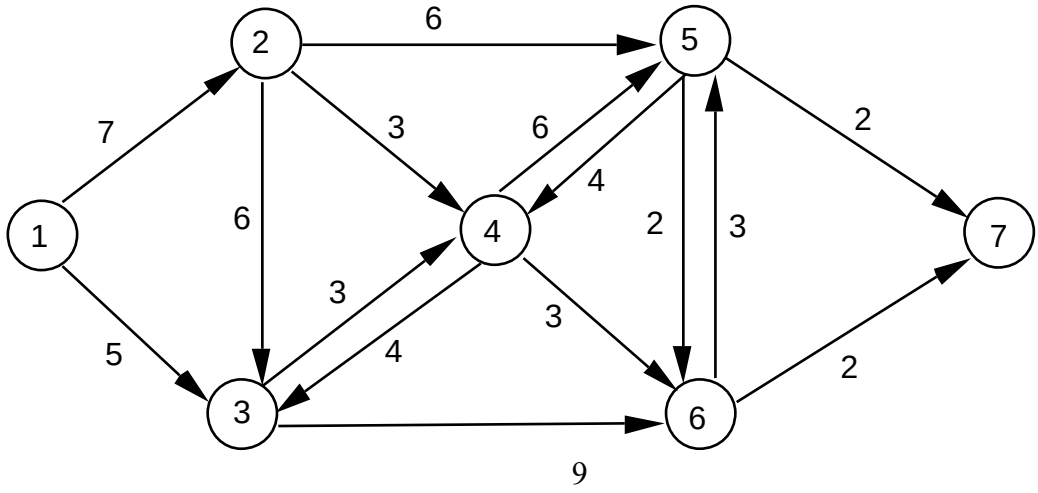
Dept. of ECE **KRP**
Univ. of Conn. **Spring 2016**

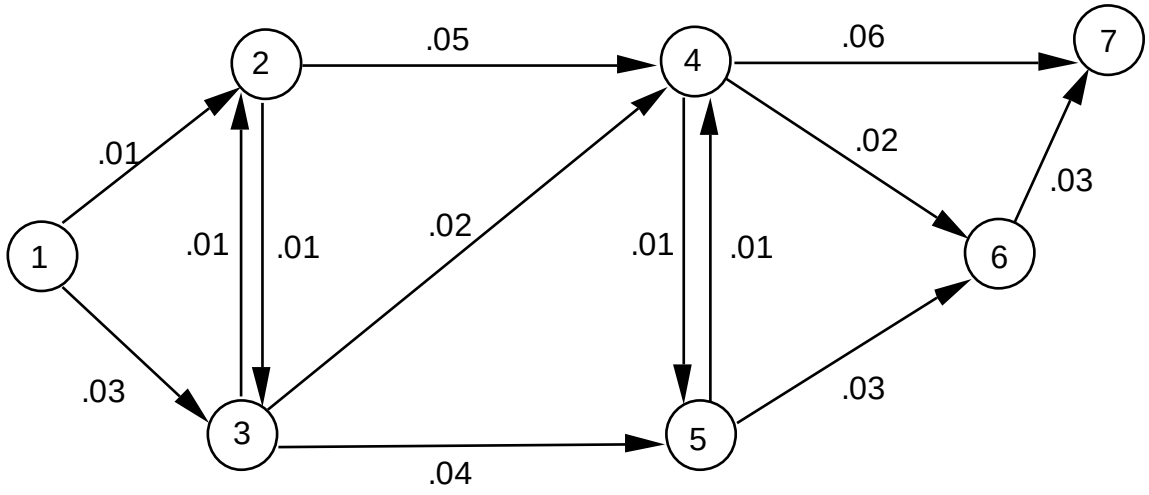
Home Work Set # 6

A. Analytical (Due March 31, 2016)

1. Find the shortest path tree from node 1 to every other node for the graph of Fig. 1 using Dijkstra and Bellman-Moore-d'Esopo-Pape algorithms. Do Bi-directional Dijkstra when we desire a shortest path from node 1 to node 7.



2. The number shown next to each link of the network in Fig. 2 is the probability of the link failing during the lifetime of a virtual circuit from node A to node B. It is assumed that links fail independently of each other. Find the most reliable path from A to B, i.e., the path for which the probability that all its links stay intact during the virtual circuit's lifetime is maximal. What is this probability?



3. Consider the single source shortest path problem. Assume that a positive lower bound is known for all arc lengths (This situation is quite common in communication networks, since one knows the minimal delay on each link, e.g., transmit time with no waiting). Modify Dijkstra's algorithm by allowing more than one node to enter the set of permanently labeled nodes at each iteration. Show that the modified algorithm is correct.

4. Consider a network in which each node knows the network topology and a positive length for each directed arc. Each node has already calculated the shortest path from itself to every other node of the network (assume that the graph is strongly connected). Assume that one arc, (i,k), *increases* in length. Show how to modify Dijkstra's algorithm for a given origin node, to recalculate as efficiently as possible the shortest paths from itself to all other nodes.

5. *Transitive Closure* R = $[r_{ij}]$ (also called the *reachability matrix*) of a digraph is defined as follows:

$$r_{ij} = \begin{cases} 1, \text{ if there is a path from node i to node j with one or more edg} \\ 0, \text{ otherwise} \end{cases}$$

Write a procedure for finding the reachability matrix starting from the adjacency matrix of a given digraph using: (a) Dijkstra's method, and (b) Floyd-Warshall method.

6. In some applications (e.g., in critical path analysis), we need the longest paths (rather than the shortest) from s to all other nodes in a directed network G. Will the maximization procedure analogous to Dijkstra, BMDP or Floyd-Warshall work? What if the network G is acyclic?

7. The objective of the following algorithm (based on Bellman-Ford method) is to compute in a distributed way a shortest path from a single origin (node 1) to all others *and* to notify node 1 that the computation has terminated. Assume that all links (i,j) are bidirectional, have nonnegative lengths c_{ij}, maintain the order of messages sent to them, and operate with no errors or failures. Each node i maintains an estimate λ_i of the shortest distance from node 1 to itself. Initially, $\lambda_i = \infty$ for all nodes $i \neq 1$, and $\lambda_1 = 0$. Node 1 initiates the computation by sending the estimate $\lambda_1 + c_{1j}$ to all neighbor nodes j. The algorithmic rules for each node $j \neq 1$are as follows:

(i) When node j receives an estimate $\lambda_i + c_{ij}$ from some other node i, after some unspecified finite delay (but before processing a subsequently received estimate), it does the following: If $\lambda_j \leq \lambda_i + c_{ij}$, node j sends ACK to node i. Otherwise, node j sends $\lambda_j = \lambda_i + c_{ij}$, marks node i as its best current predecessor on a shortest path, sends ACK to its previous best predecessor (if any), and sends the estimate $\lambda_j + c_{jk}$to each neighbor k.

(ii) Node j sends an ACK to its best current predecessor once it receives an ACK for each of the latest estimates sent to its neighbors.

We assume that each ACK is uniquely associated with a previously sent estimate, and that node 1 responds with an ACK to any length estimate it receives.

(a) Show that eventually node 1 will receive an ACK from each of its neighbors, and at that time λ_i will be the correct shortest distance for each node i.

(b) What are the advantages and disadvantages of this algorithm compared with distributed, asynchronous Bellman-Ford algorithm discussed in class.

(**References:** Read papers by Chandy and Misra, CACM, 1982, pp. 833-837 and Bertsekas, IEEE T-AC, 1982, pp. 60-74, and 610-616.)

B. Computational (Due April 14, 2016)

Combining Dijkstra's best-first search method and the best-first search procedure of Bellman-Moore-d'Esopo-Pape method, write/modify a general shortest path algorithm that can be used for graphs with negative edge weights. The algorithm must detect cycles of negative length, and use d-heaps for algorithmic efficiency. The algorithm must have options for finding only single source shortest paths or all pairs shortest paths. Explore the possibilities of inserting a labeled node at the tail of the queue, at the head of the queue, or leave it at its old position. **Extensively test the algorithm on sparse and dense graphs.**