Christine Chung
47-835 Graph Theory
Homework 3 **Due:** Wednesday, Sept 20, 2006

1. Suppose that we run the Floyd-Warshall Algorithm on a weighted directed graph G with arbitrary weights $w : E(G) \to \mathbb{R}$. Prove that L_{ii}^p remains non negative for all p if and only if w is conservative.

$\Longrightarrow$: If L_{ii}^p remains non negative for all p, but w is *not* conservative, then (by definition of conservative) there is a negative weight cycle somewhere in G. Let m be the maximum index on this negative weight cycle. Then $L_{ii}^m < 0$ for each vertex i on that cycle. This contradicts our initial assumption.

$\Longleftarrow$: If w is conservative but L_{ii}^p becomes negative for some p, then that means there is a negative-cost shortest path starting from vertex i back to vertex i using only vertices from the set $\{1...p\}$. A path from i to i is a cycle. But if there is a negative cycle, then w can't be conservative (by definition of conservative). We have thus contradicted our initial assumption.

2. Let G be a directed graph on n vertices and with conservative weights $w : E(G) \to \mathbb{R}$. Show how to find a directed circuit of minimum weight in polynomial time. Can you achieve $O(n^3)$ running time?

Assuming a "directed circuit" is a directed cycle, to find a directed circuit of minimum weight in G, start by running the Floyd-Warshall Algorithm to get the matrix of shortest paths between all pairs of vertices. In accordance with our class notes, call this matrix L^n. Let the weight of the min-weight directed circuit be called w.

Initially set w to infinity.

For $i = 1...n$

for $j = 1...n$

if $i \neq j$

if $L^n(i,j) + L^n(j,i) < w$

set w to $L^n(i,j) + L^n(j,i)$,

record the start and end point of circut i_0 to be i,

record $j_0 = j$ as an intermediate point of the circuit.

Then to get the actual list of the vertices along the circuit, just use L^n's corresponding predecessor matrix and look up the reverse sequence of the vertices along the shortest path from i_0 to j_0, then do the same for the shortest path from j_0 back to i_0. Concatenate the two sequences (being careful to reverse them first) to form the list of vertices in the circuit. The predecessor matrix P^n can be computed as the L^n matrix is computed. For the matrix P^0, set $P^0(i,j) = i$ if $i \neq j$ and there is an edge from i to j. All other entries should be left

blank. Then recursively define the rest of the matrices P^k for $1 \leq k \leq n$ based on the entry for the corresponding L^k matrix as follows:

$$P^k(i,j) = \begin{cases} P^{k-1}(i,j) & \text{if } L^{k-1}(i,j) \leq L^{k-1}(i,k) + L^{k-1}(k,j) \\ P^{k-1}(k,j) & \text{otherwise} \end{cases}$$

The pseudocode above has complexity $O(n^2)$, and to compute the predecessor matrix using dynamic programming takes $O(n^3)$ for the same reason computing the L^n matrix does: each predecessor matrix has n^2 entries to fill, each done in constant time based on already calculated entries, and there are n predecessor matrices to build all together. After adding it all together, the solution still runs in $O(n^3)$ time.

3. **Problem 4.1.8** (page 158) Determine $\kappa(G)$, $\kappa'(G)$, and $\delta(G)$ for each graph G below. For the graph on the left, $\kappa(G) = 2$ (example shown in red), $\kappa'(G) = 4$ (example show in green), and $\delta(G) = 4$. For the graph on the right, $\kappa(G) = 4$ (example shown in red), $\kappa'(G) = 4$ (example shown in red again), and $\delta(G) = 4$.

4. **Problem 4.1.14** (page 159) Let G be a connected graph in which for every edge e, there are cycles C_1 and C_2 containing e whose only common edge is e. Prove that G is 3-edge-connected. Use this to show that the Peterson graph is 3-edge-connected.

   Suppose by way of contradiction that G is not 3-edge-connected. That means there is at least one two-edge-cut or one one-edge-cut somewhere.

   Consider any one edge $e = uv$ that is cut. This edge (by definition of G in the problem statement) is part of two different cycles C_1 and C_2 that share no other edges. Which means even after this edge is removed, u and v are still connected via the new bigger cycle $C_1 \cup C_2 - e$. (In order to make this cut a separating cut, at least two more edges must be cut from the new super cycle.) Thus we've shown there can't be any one-edge separating cut in G.

   Now consider any two edges that are cut, e and f. Edge e is in two different cycles C_1^e and C_2^e. Since C_1^e and C_2^e have no other edges in common besides e, after e's removal (before f's removal), there is a bigger cycle left (as in the previous case), so everything is still connected. While f may in fact be an edge in this bigger cycle, removing it alone is not sufficient. We still need to remove one more edge in order to disconnect the vertices in the cycle. (If f is not an edge in this bigger cycle, since it also is part of two distinct cycles that share no other edges, it 's removal also does not disconnect the graph.) So we've also shown there can't be any two-edge separating cut in G.

Since we have shown there are no $2-$ or $1-$ edge separating cuts in G, and G is connected, so there is no 0-edge separating cut, G must be 3-edge-connected.

Since every edge e in the Peterson graph is part of at least two distinct cycles whose only common edge is e, by application of the statement we just proved, the Peterson graph is 3-edge-connected.

5. Problem 4.1.28 (page 160) Prove that the symmetric difference of two different edge cuts is an edge cut.

Let the two edge cuts be called $X = [S, \overline{S}]$ and $Y = [T, \overline{T}]$.

Then $Z = X \triangle Y$ is the edge cut $[S \cup T - (S \cap T), \overline{S \cup T} \cup (S \cap T)]$.

If S and T are disjoint, then it is easy to see why Z is an edge cut. (When they are disjoint, Z is simply the edge cut $[S \cup T, \overline{S \cup T}]$.) However, if there is a vertex in both S and T, we move it to the other set set since $S \cup T$ has some vertices from $\overline{S}$ and/or $\overline{T}$, and if the vertex in $S \cap T$ is adjacent to and in the same set as a vertex in $\overline{S}$ or $\overline{T}$, then an edge in Z will be invalid (on the same side of the cut). Moving the vertex in $S \cap T$ to the other set is safe since it is not adjacent to any of the vertices in the other set.

6. True or False?

(a) It is possible to find conservative edge weights for the complete directed graph on 3 vertices such that applying Dijkstra's algorithm from vertex s will return an incorrect number for the shortest sv-path for some $v \in V(G)$.

True.

(b) Let a, b, and c be three distinct vertices of a 2-connected graph G. Then there exists a bc-path using a and there exists a bc-path not using a in G.

True. If both paths from b to c needed a, then we could remove a and disconnect b from c, i.e. disconnect G, which is impossible since G has no cut vertex. If no bc-path has a, then a vertex along the path from b to a would be a cut vertex, but again G has no cut vertex.

(c) Every 4-connected graph has connectivity 4.

False. A 4-connected graph has connectivity at least 4, not necessarily exactly 4. (By definition of a 4-connected graph.)

(d) For any $1 \leq a \leq b \leq c$, there exists a graph G with $\kappa(G) = a$, $\kappa'(G) = b$, and $\delta(G) = c$.

True. As long as there is no limit set on the size of n relative to a, b, and c, we can construct G accordingly.

(e) A 3-regular connected graph is 2-edge connected.
False.