COMP 3711 – Design and Analysis of Algorithms
2022 Fall Semester – Written Assignment 4
Distributed: November 14,2022
Due: November 30, 2022, 23:59

Question 1 (20%): Cycle Detection

Let $G(V,E)$ be a undirected graph, not necessarily connected. The degree of a vertex is equal to the number of neighbors of that vertex in G. Assume that there are no nodes with degree 0. Prove that if every vertex of G has an even degree, every vertex of G appears in some cycle.

Solution

Take any vertex v. Move from v along any edge to another vertex w. Mark the edge (v,w). Move from w along any unmarked edge to another vertex. Keep doing this until we return to v. Note that we cannot get stuck in an intermediate vertex w because w has a positive even degree, and we mark two edges every time we visit w. After we return to v, we discover a cycle of marked edges. Remove all marked edges. If any vertex has zero degree afterwards, remove it as well; such a vertex must appear on the cycle of marked edges that we just traversed. Notice that every remaining vertex still has a positive even degree. Therefore, we can repeat the above argument again until all vertices are removed. This is similar to Hierholzer's algorithm for finding Euler paths, discussed in Graph Intro (Lecture 18).

Question 2 (20%): All Pairs Shortest Paths

Assume an undirected connected graph $G(V,E)$, where all edges have the same weight w where $w > 0$. Give an $O(VE)$ algorithm for computing the shortest distance between all pairs of nodes.

Solution

This is the same as finding the path with minimum number of edges between all pairs of nodes. This can be done by applying BFS on G for $|V|$ times, where each vertex is the source once. For any two nodes u, v, the shortest distance is w times the number of edges in the path from u to v. The running time is $O(V(V + E)) = O(VE)$ since the graph is connected.

Question 3 (30%): Currency Exchange

Let $x_1,..x_n$ a set of n currencies. You are given a nxn matrix M that contains the exchange rates for pairs of currencies (x_i,x_j), e.g., (HK$,US$) = 0.13 means that we can exchange HK$ 1 for US$ 0.13. Similarly, (US$, HK$) = 7.85 means that we can exchange US$ 1 for HK$ 7.85. The exchange rates for some pairs of currencies may be missing. By taking advantage of exchange rates, a trader can earn profit. For instance, in addition to the above assume the following exchange rates: (US$, EUR) = 1 and (EUR, $HK) = 7.9. One can start with HK$ 1,000, exchange to US$ 130, then exchange to EUR 130, and finally exchange back to HK$ 1,027 (130x7.9), for a profit of HK$ 27. Design an algorithm, which given M, it detects if there are any profit opportunities. The output should be TRUE or FALSE. TRUE implies that there is an opportunity, and FALSE that there is not. In case of TRUE, you do not have to output any of the profit opportunities.

Hint: Convert the problem to a problem of detecting whether there is a negative cycle in a directed graph.

Solution

Convert M to a directed graph where each currency is a node, and each exchange rate (x_i,x_j) corresponds to an edge. A profit opportunity corresponds to a cycle $x_i, x_i, x_j.. , x_k,x_i$ in the graph, such that (x_i, x_j)x..x(x_k, x_i)>1. Equivalently, log((x_i, x_j)x..x(x_k, x_i)) > 0, which implies that log(x_i, x_j)+.. log (x_k, x_i) > 0. If I use the negative log as weight edges, the problem becomes equivalent to finding a negative cycle in the graph. Assume for instance the profit opportunity HK$, US$, EUR, HK$ in the previous example. The logarithms of the conversion rates are: log(HK$, US$) = log0.13 = -2.94, log(US$, EUR) = log1 = 0, log(EUR, HK$) = log7.9 = 2.98. After negating the logarithms, I have a negative cycle 2.94+0-2.98 = -0.04. I use Floyd-Warshall for negative cycle detection, as discussed in the lecture slides.

Alternatively, I can change the concept of edge relaxation in order to apply Floyd-Warshall using the original exchange rates. Specifically, initially I set (a) $e(x_i,x_i)$=1, (b) $e(x_i,x_j)$=the input exchange rate between (x_i,x_j) if provided, and (c) $e(x_i,x_j)$ = -inf, otherwise. Then, I define $e^k(x_i,x_j)$ as the maximum value that I can obtain starting from x_i and reaching x_j, if I am allowed to exchange using the k first currencies:

$e^k(x_i,x_j)$=max($e^k(x_i,x_j)$, $e^{k-1}(x_i,x_k)$ x $e^{k-1}(x_k,x_j)$)

If there are profit opportunities, after applying Floyd-Warshall, for some element(s) in the main diagonal it will be $e(x_i,x_i)$>1.

Question 4 (30%): Maximum Flow

Let A be a $n \times m$ matrix of non-negative *real* numbers such that the sum of the entries in every row and in every column is an integer. Prove that there is an $n \times m$ matrix B of non-negative *integers* with the same sums as in A, in every row and every column.

Solution

Let $a_1, \ldots, a_n$ and $b_1, \ldots, b_m$ be the sums of the entries in the rows and columns of A, respectively. Consider the complete bipartite graph $G = (A \cup B, E)$, where $A = \{A_1, \ldots, A_n\}$ and $B = \{B_1, \ldots, B_m\}$. Orient all edges from A to B, and set their capacity to infinity. Add a source s, connect it to all vertices of A by oriented edges of capacity a_i for the vertex A_i. Add a sink t and connect all vertices of B to it by oriented edges of capacity b_j for vertex B_j. The minimum cut capacity is $a_1 + \ldots + a_n = b_1, \ldots, b_m$. Since all capacities are integral, the Ford-Furkerson outputs an integer flow with maximum value. We define $B[i,j]$ to be the value of this flow on the edge (A_i, B_j) .