# Homework 11

2019.05.16

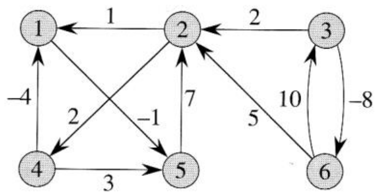
Note: When the exercise asks you to "design an algorithm for…," it always means that "designs an EFFICIENT algorithm for … and ANALYZES your algorithm". You should keep this in mind when writing solutions.

**1. Exercises 25.2-1**

Run the Floyd-Warshall algorithm on the weighted, directed graph of Figure 25.2. and answer the following questions:

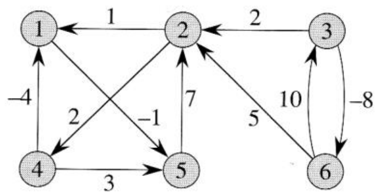
***a.*** Show the matrix $D^{(k)}$ that results for each iteration of the outer loop.

***b.*** List the vertices of one such shortest path from $v_6$ to $v_1$.




**2. Exercises 25.3-1**

Use Johnson's algorithm to find the shortest paths between all pairs of vertices in the graph of Figure 25.2. Show the values of $h$ and $\hat{w}$ computed by the algorithm.




**3. Problem 24-3** ***Arbitrage***

***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 49 Indian rupees, 1 Indian rupee buys 2 Japanese yen, and 1 Japanese yen buys 0.0107 U.S. dollars. Then, by converting currencies, a trader can start with 1 U.S. dollar and buy $49 \times 2 \times 0.0107 = 1.0486$ U.S. dollars, thus turning a profit of 4.86 percent.

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

***a***. Give an efficient algorithm to determine whether or not there exists a sequence of currencies <$c_{i1}$, $c_{i2}$, …, $c_{ik}$> such that $R[i_1, i_2] \times R[i_2, i_3] \times \ldots \times R[i_{k-1}, i_k] \times R[i_k, i_1] > 1$ . Analyze the running time of your algorithm.

***b***. Give an efficient algorithm to print out such a sequence if one exists. Analyze the running time of your algorithm.

**4. Problem 25-1: Transitive closure of a dynamic graph** (textbook, page 705)

Suppose that we wish to maintain the transitive closure of a directed graph $G = (V, E)$ as we insert edges into $E$. That is, after each edge has been inserted, we want to update the transitive closure of the edges inserted so far. Assume that the graph $G$ has no edges initially and that we represent the transitive closure as a Boolean matrix.

***a***. Show how to update the transitive closure $G^* = (V, E^*)$ of a graph $G = (V, E)$ in $O(V^2)$ time when a new edge is added to $G$.

***b***. Give an example of a graph $G$ and an edge $e$ such that $O(V^2)$ time is required to update the transitive closure after the insertion of $e$ into $G$, no matter what algorithm is used.

***c***. Describe an efficient algorithm for updating the transitive closure as edges are inserted into the graph. For any sequence of $n$ insertions, your algorithm should run in total time $\sum_{i=1}^{n} t_i = O(V^3)$, where $t_i$ is the time to update the transitive closure upon inserting the $i$th edge. Prove that your algorithm attains this time bound.

**5. Ext. 9-1**

Give an efficient algorithm to count the total number of shortest paths between any pair $(u, v)$ of vertices in a directed graph. (There may be the negative edge or a cycle but definitely no negative-cycle)

**6.** 給定三個整數 $m, n, p$，設計一個有效率的演算法，計算 $m^n \bmod p$（$\bmod P$ 可忽略，主要為探討如何快速計算出$m^n$）。

**7.** How to use Floyd-Warshall algorithm to determine whether there is a negative cycle in graph $G = (V, E)$ or not?