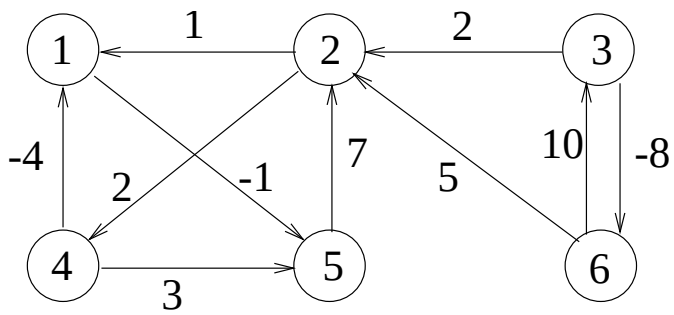
COMP 3711H – Fall 2014
Tutorial 9

1. Run the Floyd-Warshall algorithm on the weighted, directed graph shown in the figure. Show the matrix $D^{(k)}$ that results for each iteration of the outer loop.



2. Let $G = (V, E)$ be Directed Acyclic Graph and s a vertex from which it's possible to get to all vertices. Show how to build a shortest path tree routed at s in $O(|V| + |E|)$ time.

3. Give an $O(nW)$ dynamic programming algorithm for the 0-1 knapsack problem where n is the number of items and W is the max weight that can fit into the knapack. Recall that the input is i items with given weights $w_1, w_2, \ldots, w_n$ and associated values v_1, $v_2, \ldots, v_n$ and the objective is to choose a set of items with weight $\leq W$ with maximum value.

   Now suppose that you are given *two* knapsacks with the same max weight. Give an $O(nW^2)$ dynamic programming algorithm for finding the maximum value of items that can be carried by the two knapsacks.

4. (CLRS) Give an algorithm that takes as input a directed graph with positive edge weights, and returns the cost of the shortest cycle in the graph (if the graph is acyclic, it should say so). Your algorithm should take time at most $O(n^3)$, where n is the number of vertices in the graph.

5. The subset sum problem is: Given a set of n positive integers, $S = \{x_1, x_2, \ldots, x_n\}$ and an integer W determine whether there is a subset $S' \subseteq S$, such that the sum of the elements in S' is equal to W. For example, if $S = \{4, 2, 8, 9\}$ and $W = 11$, then the answer is "yes" because there is a subset $S' = \{2, 9\}$ whose elements sum to 11. Give a dynamic programming solution to the subset sum problem that runs in $O(nW)$ time. Justify the correctness and running time of your algorithm.