CS420/520 HW 4

Problem 1: Ford-Fulkerson. (*10 points*) In this problem, we will trace the partial execution of the Ford-Fulkerson algorithm on a network.

a) Consider the network G shown in Fig. 1 a), and consider the initial flow f in Fig. 1 b). Show the residual network G_f for this flow.

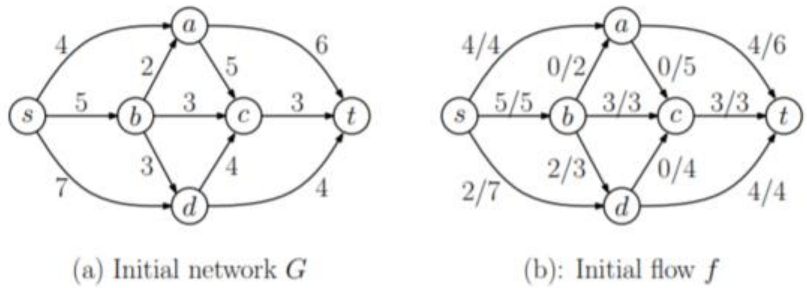

Figure 1: Problem 1: Ford-Fulkerson.

b) Find any augmenting path in G_f. How much flow can you push along this path? Show the updated flow (in the same manner as Fig. 1(b)).

c) Show the residual network that results for your flow from b).

d) Is this the final maximum flow in this network? (If not, keep running Ford-Fulkerson until you get the maximum flow, and show the final flow.) What is the value of the maximum flow?

e) Show the residual network for your maximum flow from d). (If the flow from c) was already maximum, then state this.)

f) Show the cut that results by partitioning the network into two subsets of vertices, the vertices S that are reachable from s and the remaining vertices T = V -S. What is the capacity of this cut? (It should match your flow value, if you did everything correctly.)

Problem 2: Disjoint Roads (*10 points*)

A number k of trucking companies, $c_1, \ldots, c_k$, want to use a common road system, which is modeled as a directed graph, for delivering goods from source locations to a common target location. Each trucking company c_i has its own source location, modeled as a vertex s_i in the graph, and the common target location is another vertex t. (All these k + 1 vertices are distinct.) The trucking companies want to share the road system for delivering their goods, but they want to avoid getting in each other's way while driving. Thus, they want to find k edge-disjoint paths in the graph, one connecting each source s_i to the target t. We assume that there is no problem if trucks of different companies pass through a common vertex. Design an algorithm for the companies to use to determine k such paths, if possible, and otherwise return "impossible". What is the running time of your algorithm.

Problem 3: Food Truck Orders (*10 points*)

The Acme food truck produces a large variety of different lunch menu items. Unfortunately, they can only produce their foods in limited quantities, so they often run out of popular items, making customers sad. To minimize sadness, Acme Foods is implementing a sophisticated lunch-ordering system. Customers text in their acceptable choices before lunch time. Then they can use an algorithm to preassign lunches to customers. Customers who do not get one of their choices should receive a $10 voucher. Acme Foods would like to minimize the number of vouchers they give out. Give an efficient algorithm for Acme Foods to assign lunches to customers. In general, suppose that, on a given day, Acme Foods has produced m types of food items $b_1, \ldots, b_m$, and the quantity of each type of food item b_j is exactly q_j. Suppose that n customers $a_1, \ldots, a_n$ text in their preferences, where each customer a_i submits a set A_i of one or more acceptable lunch choices. The algorithm should assign each customer either one of his/her choices or a $10 voucher. It should minimize the number of vouchers. What is the running time of your algorithm?

CS 520 Problem:

An edge of a flow network is called critical if decreasing the capacity of this edge results in a decrease in the maximum flow value. Design an efficient algorithm that, given a network G with source s and sink t finds all critical edges in a network (assuming one exists). What is the running time of the algorithm.