Homework 7

Due date: 3/3/2021 right before class

Problem 1. Q8 page 418

Statistically, the arrival of spring typically results in increased accidents and increased need for emergency medical treatment, which often requires blood transfusions. Consider the problem faced by a hospital that is trying to evaluate whether its blood supply is sufficient. The basic rule for blood donation is the following. A person's own blood supply has certain antigens present (we can think of antigens as a kind of molecular signature); and a person cannot receive blood with a particular antigen if their own blood does not have this antigen present. Concretely, this principle underpins the division of blood into four types: A, B, AB, and O. Blood of type A has the A antigen, blood of type B has the B antigen, blood of type AB has both, and blood of type O has neither. Thus, patients with type A can receive only blood types A or O in a transfusion, patients with type B can receive only B or O, patients with type O can receive only O, and patients with type AB can receive any of the four types.

1. Let s_O, s_A, s_B, and s_{AB} denote the supply in whole units of the different blood types on hand. Assume that the hospital knows the projected demand for each blood type d_O, d_A, d_B, and d_{AB} for the coming week. Give a polynomial-time algorithm to evaluate if the blood on hand would suffice for the projected need.

2. Consider the following example. Over the next week, they expect to need at most 100 units of blood. The typical distribution of blood types in U.S. patients is roughly 45 percent type O, 42 percent type A, 10 percent type B, and 3 percent type AB. The hospital wants to know if the blood supply it has on hand would be enough if 100 patients arrive with the expected type distribution. There is a total of 105 units of blood on hand. The table below gives these demands, and the supply on hand.

blood type	supply	demand
O	50	45
A	36	42
B	11	8
AB	8	3

Is the 105 units of blood on hand enough to satisfy the 100 units of demand? Find an allocation that satisfies the maximum possible number of patients. Use an argument based on a minimum-capacity cut to show why not all patients can receive blood. Also, provide an explanation for this fact that would be understandable to the clinic administrators, who have not taken a course on algorithms. (So, for example, this explanation should not involve the words flow, cut, or graph in the sense we use them in this book.)

Problem 2. Q10 page 419

Suppose you are given a directed graph $G = (V, E)$, with a positive integer capacity c_e on each edge e, a source $s \in V$, and a sink $t \in V$. You are also given a maximum $s-t$ flow in G, defined by a flow value f_e on each edge e. The flow f is acyclic: There is no cycle in G on which all edges carry positive flow. The flow f is also integer-valued. Now suppose we pick a specific edge $e^* \in E$ and reduce its capacity by 1 unit. Show how to find a maximum flow in the resulting capacitated graph in time $O(m+n)$, where m is the number of edges in G and n is the number of nodes.

Problem 3. Q12 page 419

Consider the following problem. You are given a flow network with unit capacity edges: It consists of a directed graph $G = (V, E)$, a source $s \in V$, and a sink $t \in V$; and $c_e = 1$ for every $e \in E$. You are also given a parameter k.

The goal is to delete k edges so as to reduce the maximum $s-t$ flow in G by as much as possible. In other words, you should find a set of edges $F \subseteq E$ so that $|F| = k$ and the maximum $s-t$ flow in $G^{'} = (V, E-F)$ is as small as possible subject to this. Give a polynomial-time algorithm to solve this problem.

Problem 4. Q14 page 421

We define the Escape Problem as follows. We are given a directed graph $G = (V, E)$ (picture a network of roads). A certain collection of nodes $X \subset V$ are designated as populated nodes, and a certain other collection $S \subset V$ are designated as safe nodes. (Assume that X and S are disjoint.) In case of an emergency, we want evacuation routes from the populated nodes to the safe nodes. A set of evacuation routes is defined as a set of paths in G so that (i) each node in X is the tail of one path, (ii) the last node on each path lies in S, and (iii) the paths do not share any edges. Such a set of paths gives a way for the occupants of the populated nodes to "escape" to S, without overly congesting any edge in G.

1. Given G, X, and S, show how to decide in polynomial time whether such a set of evacuation routes exists.

2. Suppose we have exactly the same problem as in (a), but we want to enforce an even stronger version of the "no congestion" condition (iii). Thus we change (iii) to say "the paths do not share any nodes." With this new condition, show how to decide in polynomial time whether such a set of evacuation routes exists. Also, provide an example with the same G, X, and S, in which the answer is yes to the question in (a) but no to the question in (b).

To solve this problem, construct a graph $G = (V, E)$ as follows: Let the set of vertices consist of a super-source node, four supply nodes (one for each blood type) adjacent to the source, four demand nodes and a super sink node that is adjacent to the demand nodes. For each supply node u and demand node v, construct an edge (u, v) if type v can receive blood from type u and set the capacity to ∞ or the demand for type v. Construct an edge (s, u) between the source s and each supply node u with the capacity set to the available supply of type u. Similarly, for each demand node v and the sink t, construct an edge (v, t) with the capacity set to the demand for type v.

Now compute an (integer-valued) maximum flow on this graph. Since the graph has constant size, the scaling max-flow algorithm takes time $O(\log C)$, where C is the total supply, and the Preflow-Push algorithm takes constant time.

We claim that there is sufficient supply for the projected need. if and only if the edges from the demand nodes to the sink are all saturated in the resulting max-flow. Indeed, if there is sufficient supply, in which s_{ST} of type S are used for type T, then we can send a flow of s_{ST} from the supply node of type S to the demand node of type T, and respect all capacity conditions. Conversely, if there is a flow saturating all edges from demand nodes to the sink, then there is an integer flow with this property; if it sends f_{ST} units of flow from the supply node for type S to the demand node for type T, then we can use f_{ST} nodes of type S for patients of type T.

(b) Consider a cut containing the source, and the supply and demand nodes for B and A. The capacity of this cut is $50 + 36 + 10 + 3 - 99$, and hence all 100 units of demand cannot be satisfied.

An explanation for the clinic administrators: There are 87 people with demand for blood types O and A; these can only be satisfied by donors with blood types O and A; and there are only 86 such donors.

Note. We observe that part (a) can also be solved by a greedy algorithm; basically, it works as follows. The O group can only receive blood from O donors; so if the O group is not satisfied, there is no solution. Otherwise, satisfy the A and B groups using the leftovers from the O group; if this is not possible, there is no solution. Finally, satisfy the AB group using any remaining leftovers. A short explanation of correctness (basically following the above reasoning) is necessary for this algorithm, as it was with the flow algorithm.

[1] ex717.885.42

We will assume that the flow f is integer-valued. Let $e^* = (v, w)$. If the edge e^* is not saturated with flow, then reducing its capacity by one unit does not cause a problem. So assume it is saturated.

We first reduce the flow on e^* to satisfy the capacity conditions. We now have to restore the capacity conditions. We construct a path from w to t such that all edges carry flow, and we reduce the flow by one unit on each of these edges. We then do the same thing from v back to s. Note that because the flow f is acyclic, we do not encounter any edge twice in this process, so all edges we traverse have their flow reduced by exactly one unit, and the capacity condition is restored.

Let f' be the current flow. We have to decide whether f' is a maximum flow, or whether the flow value can be increased. Since f was a maximum flow, and the value of f' is only one unit less than f, we attempt to find a single augmenting path from s to t in the residual graph $G_{f'}$. If we fail to find one, then f' is maximum. Else, the flow is augmented to have a value at least that of f; since the current flow network cannot have a larger maximum flow value than the original one, this is a maximum flow.

[1] ex257.178.863

If the minimum s-t cut has size $\leq k$, then we can reduce the flow to 0. Otherwise, let $f > k$ be the value of the maximum s-t flow. We identify a minimum s-t cut (A, B), and delete k of the edges out of A. The resulting subgraph has a maximum flow value of at most $f - k$.

But we claim that for any set of edges F of size k, the subgraph $G' = (V, E - F)$ has an s-t flow of value at least $f - k$. Indeed, consider any cut (A, B) of G'. There are at least f edges out of A in G, and at most k have been deleted, so there are at least $f - k$ edges out of A in G'. Thus, the minimum cut in G' has value at least $f - k$, and so there is a flow of at least this value.

[1] ex225.750.725

(a) We turn G into a flow network as follows. Let $k = |X|$. We give each edge a capacity of 1, we add a super-source s with capacity-1 edges to each node in X, and we add a super-sink t with a capacity-k edge from each node in S. We then check whether the maximum s-t flow has a value of k.

If so, then it has an integer-valued one, and by deleting s and t, we get the desired set of disjoint paths from X to S. Conversely, if there is a solution to the escape problem, then by sending one unit of flow to each node in X, one unit of flow along each of the paths in the escape solution, and c units of flow to t from each node in S that is at the head of c paths, we get a flow of value k.

(b) We use the result of the previous problem on node-capacitated graphs. We again attach a source s to each node in X, and connect each node in S to a sink t. We now give each node in G a capacity of 1, and we give s and t each a capacity of k. By an argument strictly analogous to the one in (a), there is a flow of value k respecting the node capacities if and only if there is a solution to the escape problem with the stronger restrictions on nodes.

A simple example of an instance where there is a solution under the constraints in (a) but not under the constraints in (b) is as follows. Let $X = \{x_1, x_2\}$ and $S = \{s_1, s_2\}$; suppose G has nodes $X \cup S \cup \{v\}$ for an additional node v, and edges (x_i, v) and (v, s_j) for $i = 1, 2$ and $j = 1, 2$.

[1] ex672.975.278