

Homework 5

Released 4/2/2020

Due 4/15/2020 11:59pm in Gradescope

Instructions. You may work in groups, but you must write solutions yourself. List collaborators on your submission. Also list any sources of help (including online sources) other than the textbook and course staff.

If you are asked to design an algorithm, please provide: (a) the pseudocode or precise description in words of the algorithm, (b) an explanation of the intuition for the algorithm, (c) a proof of correctness, (d) the running time of your algorithm and (e) justification for your running time analysis.

Submissions. Please submit a PDF file. You may submit a scanned handwritten document, but a typed submission is preferred. Please assign pages to questions in Gradescope.

1. (10 points) **Increasing Capacities** (K&T 7.5) Prove the following claim or provide a counterexample: Let (A, B) be a minimum $s - t$ cut in a flow network. Add 1 to the capacity of every edge in the network. Then (A, B) is still a minimum cut for the new network.

2. (10 points) **Greedy Flow** (Erickson 10.17) An eager student of network flow proposes the following change to the Ford-Fulkerson algorithm: instead of computing a residual graph, it looks for an augmenting path in the flow network, decreases edge capacities on the found path by the flow on this path, and repeats the process. Saturated edges just get deleted. Show that GREEDYFLOW does not compute a maximum flow.

3. (20 points) **Cold War Flow** Nation A is preparing for a potential armed conflict with Nation B. They are concerned with how many military units B could move at one time from their home base to the battlefield. They have modeled B's transportation network as a directed graph G , where *every edge has the same capacity*, there is one source node at the base, and one target node at the battlefield. They have computed that the maximum flow through this network is F .

A has a network of spies and saboteurs throughout B's territory. In the event of war, these agents could disable a fixed number k of the edges in B's network. A's planners would like to compute which edges they should choose to reduce the capacity of B's network the most. Present an algorithm to let them do this, that takes as input G and k and gives a set of k edges as output. Prove that your algorithm is correct, and find its big-O running time, which should be polynomial in m (the number of edges in G).

4. (20 points) **Acyclic Flows** (Erickson 10.6) An (s, t) -flow in a network G is *acyclic* if there are no directed cycles with positive flow value on every edge; i.e., the subgraph of edges with positive flow value is a DAG.

(a) Describe and analyze an algorithm to compute an acyclic maximum (s, t) -flow in a given flow network. Your algorithm should have the same asymptotic running time as Ford-Fulkerson.

(b) Describe and analyze an algorithm to determine whether every maximum (s, t) -flow in a given flow network is acyclic.

5. (20 points) **Separating Edges** (Erickson 10.11) This problem explores defining a cut using a set of edges rather than a set of nodes. We say that a subset X of directed edges *separates* s and t if every directed path from s to t contains at least one directed edge in X . For any subset S of vertices, let δS denote the set of directed edges leaving S ; that is, $\delta S := \{u \rightarrow v \mid u \in S, v \notin S\}$.

(a) Prove that if (S, T) is an (s, t) -cut, then δS separates s and t .

(b) Let X be any subset of edges that separates s and t . Prove there is an (s, t) -cut (S, T) such that $\delta S \subseteq X$.

(c) Let X be a *minimal* subset of edges that separates s and t . (Such an edge set is sometimes called a *bond*.) Prove that there is an (s, t) -cut (S, T) such that $\delta S = X$.

6. (20 points) **Incremental Change** (Erickson 10.14) You are given a flow network G with integer edge capacities and an integer maximum flow f^* in G . Design algorithms for the following operations:

(a) INCREMENT(e): Increase the capacity of edge e by 1 and update the maximum flow.

(b) DECREMENT(e): Decrease the capacity of edge e by 1 and update the maximum flow.

Both algorithms should update the maximum flow in time $O(m + n)$.

7. (0 points). How long did it take you to complete this assignment?