1 Cuts and Flows (50 points)

(a) (20 points) Suppose you are given a directed graph $G = (V, E)$, two vertices $s$ and $t$, a capacity function $c : E \rightarrow \mathbb{R}_+^+$, and a second function $f : E \rightarrow \mathbb{R}$. Give an $O(m + n)$-time algorithm to determine whether $f$ is a maximum feasible $(s, t)$-flow in $G$. As always, prove running time and correctness.

Cuts are sometimes defined as subsets of the edges of the graph, instead of as partitions of its vertices. We will sometimes go back and forth between these two definitions without making much of a distinction, so in this problem you will prove that these two definitions are essentially equivalent.

Let $G = (V, E)$ be a directed graph, and let $s, t \in V$. You may assume that $t$ is reachable from $s$ in $G$. We say that a set of edges $X \subseteq E$ separates $s$ and $t$ if every directed path from $s$ to $t$ contains at least one edge in $X$. For any subset $S$ of vertices, let $\delta(S)$ denote the set of edges leaving $S$, i.e., let $\delta(S) = \{(u, v) \in E : u \in S, v \notin S\}$.

(b) (15 points) Let $(S, \bar{S})$ be an $(s, t)$ cut (i.e., $s \in S$ and $t \in \bar{S}$, where $\bar{S} = V \setminus S$). Prove that $\delta(S)$ separates $s$ and $t$.

(c) (15 points) Let $X$ be an arbitrary subset of edges that separates $s$ and $t$. Prove that there is an $(s, t)$-cut $(S, \bar{S})$ such that $\delta(S) \subseteq X$.

It is easy to combine these two parts to prove that the minimum-weight cut has weight equal to the minimum weight separating set. We might use this later in the course!

2 Boruvka’s Algorithm (50 points)

(a) This was stated in class, but let’s prove it formally. Let $G = (V, E)$ be an undirected graph and let $w : E \rightarrow \mathbb{R}_{\geq 0}$ be edge weights. Prove that if all edge weights are distinct ($w(e) \neq w(e')$ for all $e \neq e' \in E$) then there is a unique minimum spanning tree (MST).

Let $G = (V, E)$ be a connected, undirected graph and let $w : E \rightarrow \mathbb{R}_{\geq 0}$ be distinct edge weights ($w$ is injective). We’re going to analyze yet another MST algorithm: Boruvka’s MST algorithm (from 1926), which is a bit like a distributed version of Kruskal.

We begin by having each vertex mark the lightest edge incident to it. (For instance, if the graph were a 4-cycle with edges of lengths 1, 3, 2, and 4 around the cycle, then two vertices would mark

1
the “1” edge and the other two vertices will mark the “2” edge). This creates a forest $F$ of marked edges. (Convince yourself that there won’t be any cycles!). In the next step, each tree in $F$ marks the shortest edge incident to it (the shortest edge having one endpoint in the tree and one endpoint not in the tree), creating a new forest $F'$. This process repeats until we have only one tree.

(b) Show correctness of this algorithm by proving that the set of edges in the current forest is always contained in the MST.

(c) Show how you can run each iteration of the algorithm in $O(m)$ time with just a few runs of DFS and no fancy data structures (heaps, union-find – remember, this algorithm was from 1926!). In other words, given a current forest $F$ of marked edges, show how to find the set of edges which consists of the shortest edge incident to each tree in $F$ in time $O(m)$.

(d) Prove an upper bound of $O(m \log n)$ on the running time of this algorithm.