Flows in graphs

The min-cut max-flow theorem is a powerful culmination of various topics we have seen so far: Hall's, König's, and Menger's Theorems all are really special cases of the min-cut max-flow idea.
We will consider directed graphs for this topics: simple graphs in which every edge has a direction. We denote xy for an edge $x \to y$. The direction of the edge does not imply anything about which direction flows are possible, it is only a label to help distinguish the directions. Furthermore, suppose that our graph has two special vertices, a *source* s, and a *sink* t.

We define a *flow* as a function $f : E(G) \to \mathbb{N}$. We let $f(v, S) = \sum_{w \in S} f(v, w)$ for any $v \in V$ and $S \subseteq V$. Furthermore, we let the *capacity* be a function $c : E(G) \to \mathbb{N}_{\geq 0}$.

- $f(x, y) \leq c(x, y)$ for all $(x, y) \in E(G)$,
- $f(x, y) = -f(y, x)$ for all $(x, y) \in E(G)$,
- $f(V, s) = f(t, V) = 0$,
- $f(V, v) = f(v, V)$ for all $v \in V(G) \setminus \{s, t\}$.

Let the *value* of the flow be $f(s, V) = f(V, t)$. (It should be easy to see that this equality holds.)

We are interested in edge cuts $[S, \overline{S}]$ such that $s \in S$ and $t \in \overline{S}$, since those are cuts that stop all possible flow from the source to the sink. We let the capacity of a cut be defined as

$$c(S) = \sum_{vw \in [S, \overline{S}]} c(vw).$$

Note the directedness in this definition: we are only counting the capacity of edges from S to $\overline{S}$.

Theorem 1 (Ford-Fulkerson). *The maximum possible value of a flow is equal to the minimum capacity of a cut.* Thm 6.2.2 p.153

Exercise 1. *Show that the Ford-Fulkerson algorithm terminates if the capacities and flows take rational values.* 2 points

Exercise 2. *Suppose that all edges have capacity 1 (in both directions). What can you say about the run time of the FF algorithm?* 3 points

Exercise 3. *Can you generalize the min-cut max-flow result to networks that have multiple sources and sinks?* 2 points

Exercise 4. *Use min-cut max-flow to prove Hall's, König's, and Menger's Theorems.* 2 pts each

SageMath

Exercise 5. *Use Sage to find an example of a triangle-free graph that has edge connectivity greater than its vertex connectivity. Include the code that you used.* 1

Exercise 6. *Use Sage to form a conjecture on the edge-connectivity of the n-dimensional hypercube graph. These are built into sage as* `graphs.CubeGraph(n)`*. Include the code that you used.* 2