CS3000: Algorithms & Data — Summer '23 — Laney Strange

Ford-Fulkerson Algorithm

The Ford-Fulkerson algorithm computes the value of the max flow in a flow network G. G is a directed, weighted graph with a source vertex s and sink vertex t. Instead of attaching attributes to vertices as we've done in some other graph algorithms, we attach an attribute f to each edge (u, v) of the graph. The f attribute for a given edge indicates the flow on that edge so far.

Each edge (u, v) also has a capacity $c(u, v)$. As long as there is an augmenting path from s to t, with available capacity on all edges in the path, we send flow along one of the paths. Then we find another path, and so on.

```
FORD-FULKERSON(G, s, t)
for each edge (u, v) ∈ G.E
    (u, v).f = 0
while there exists a path p from s to t in the residual network G_f
    c_f(p) = min{c_f(u, v) : (u, v) is in p}
    for each edge (u, v) in p
        if (u, v) ∈ G.E
            (u, v).f = (u, v).f + c_f(p)
        else
            (v, u).f = (v, u).f − c_f(p)
```

We typeset the Ford-Fulkerson procedure above with the following LaTeX:

```
\begin{codebox}
\Procname{$\proc{Ford-Fulkerson}(G, s, t)$}
\li \For each edge $(u, v) \in G.E$
\Do
\li $(u,v).f = 0$
\End
\li \While there exists a path $p$ from $s$ to $t$ in the residual network $G_f$
\Do
\li $c_f(p) =$ min$\{c_f(u,v) : (u,v)$ is in $p\}$
\li \For each edge $(u,v)$ in $p$
\Do
\li \If $(u,v) \in G.E$
\Then
\li $(u,v).f \gets (u,v).f + c_f(p)$
\li \Else
\li $(v,u).f \gets (v,u).f - c_f(p)$
\End
\End
\end{codebox}
```