1 The Ford-Fulkerson Algorithm

Let $G = (V, E)$ be an undirected graph with edge capacity $c\colon E \to \mathbb{R}^{\geq 0}$ and $s, t \in V$ be the source and the sink of the network.

The generic Ford-Fulkerson for the max-flow problem works as follows.

1. $f = 0$.

   $\text{resCap}(u, v) = \text{resCap}(v, u) = c_{u,v}$ for all $(u, v) \in E$.

2. Repeatedly call the procedure Aug-Path(s) until it returns false.

3. Output f.

The Procedure Aug-Path(s) attempts to compute an f-augmenting path starting from s. If the flow f is successfully augmented, it returns true. Otherwise it returns false. It works as follows.

1. Mark all the vertices as *unvisited*.

2. $k \longleftarrow$ Recursive-Aug-Path(s, ∞)

3. If $k > 0$, then set $f \longleftarrow f + k$ and return true.

   Otherwise, return false.

The recursive procedure Recursive-Aug-Path(u, Δ) uses simple DFS to find an f-augmenting path starting from u with value no more than Δ. If successfully found, it returns the value of the augmenting path. Otherwise, it returns false. It works as follows.

1. If u is the sink t,

   then return Δ.

2. Mark the vertex u as *visited*.

3. For each *unvisited* neighbor v of u with $\text{resCap}(u, v) > 0$, do

   - Let $k = \text{Recursive-Aug-Path}\,(\ v, \min\{\Delta, \text{resCap}(u, v)\}\ )$.

   - If $k > 0$, then // Successfully found the path, augment the path

     - $\text{resCap}(u, v) \longleftarrow \text{resCap}(u, v) - \Delta$.
     - $\text{resCap}(v, u) \longleftarrow \text{resCap}(v, u) + \Delta$.
     - Return k.

4. Return false.

2 The $O(m^2 \log f)$ Capacity Scaling Algorithm

The capacity scaling algorithm is almost identical to the Ford-Fulkerson algorithm. It works as follows.

1. $f = 0$.

   $\text{resCap}(u, v) = \text{resCap}(v, u) = c_{u,v}$ for all $(u, v) \in E$.

2. Let $\Delta = \max_{(u,v) \in E} c_{u,v}$.

3. Repeat while $\Delta > 0$, do the following.

   - Repeatedly call the procedure Aug-Path-of-Delta(s, Δ) until it returns false.

   - Set $\Delta \longleftarrow \Delta/2$.

4. Output f.

The Procedure Aug-Path-of-Delta(s, Δ) attempts to compute an augmenting path that starts from s and has value Δ. If the flow f is successfully augmented, it returns true. Otherwise it returns false. It works as follows.

1. Mark all the vertices as *unvisited*.

2. If Recursive-Aug-Path-of-Delta(s, Δ) returns true, then set $f \longleftarrow f + \Delta$ and return true.

   Otherwise, return false.

The recursive procedure Recursive-Aug-Path-of-Delta(u, Δ) uses simple DFS to find an f-augmenting path starting from u with value exactly Δ. If such a path is successfully found, it returns true. Otherwise, it returns false. It works as follows.

1. If u is the sink t,

   then return true.

2. Mark the vertex u as *visited*.

3. For each *unvisited* neighbor v of u with $\text{resCap}(u, v) \geq \Delta$, do

   - If Recursive-Aug-Path-of-Delta (v, Δ) returns true, then
     - $\text{resCap}(u, v) \longleftarrow \text{resCap}(u, v) - \Delta$.
     - $\text{resCap}(v, u) \longleftarrow \text{resCap}(v, u) + \Delta$.
     - Return true.

4. Return false.