Chapter 26: Maximum Flow.

Definition: a flow network is a directed graph $G = (V,E)$ with the following constraints.

1. $\exists c : E \rightarrow \mathbb{R}^+$; $c(u,v) \geq 0$ is the capacity of edge $(u,v)$; for $(u,v) \notin E$, we define $c(u,v) = 0$.

2. $(u,v) \in E \Rightarrow (v,u) \notin E$; no reverse edges.

3. $\forall v \in V, (v,v) \notin E$; no self-loops.

4. There exist two distinguished nodes, $s$ (source) and $t$ (sink); $v \in V \setminus \{s,t\} \Rightarrow \exists s \rightsquigarrow v \rightsquigarrow t$; every nondistinguished lies on a path from source to sink.

Observation: Since each node, except possibly $s$, has an incoming edge, we have $E \geq V - 1$. That is, $E = \Omega(V)$. 

Definition: a flow on flow network $[G = (V, E), s, t, c]$ is a function $f : V \times V \to \mathbb{R}^+$ with the following properties.

1. capacity constraint: $\forall u, v \in V, 0 \leq f(u, v) \leq c(u, v)$.

2. flow conservation constraint: $\forall u \in V\{s, t\}, \sum_{v \in V} f(v, u) = \sum_{v \in V} f(u, v)$; internal nodes cannot store flow material.

3. The value of flow $f$ is $|f| = \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s)$.

Observations:

$|f|$ is different from $|f(u, v)|$; the latter is a real number characterizing the local behavior of $f$ between vertices $u$ and $v$, while the former is a global property of the entire flow, specifically the difference between total flow leaving the source and total flow entering the source.

$|f|$ is the objective function that we wish to maximize over all possible flows.
$|f| = 19$

\[ \begin{align*}
  s & \quad 11/16 \quad v_1 \quad 12/12 \quad v_3 \quad 15/20 \quad t \\
  8/13 & \quad 1/4 \quad v_2 \quad 4/9 \quad v_4 \quad 7/7 \\
  & \quad 11/14 \quad v_4 \quad 4/4
\end{align*} \]

Anti-parallel edges

\[ \begin{align*}
  s & \quad 16 \quad v_1 \quad 12 \quad v_3 \quad 20 \quad t \\
  13 & \quad 10 \quad v_2 \quad 4 \quad 9 \\
  & \quad 14 \quad v_4 \quad 7 \quad 4
\end{align*} \]

\[ \begin{align*}
  s & \quad 16 \quad v_1 \quad 12 \quad v_3 \quad 20 \quad t \\
  13 & \quad 10 \quad v_2 \quad 4 \quad 9 \\
  & \quad 14 \quad v_4 \quad 7 \quad 4
\end{align*} \]
Multiple sources and sinks
Definitions:

1. Let \( G = (V, E) \) be a flow network with source \( s \) and sink \( t \). Let \( f \) be a flow in \( G \). Then, for every \((u, v) \in V \times V\), the residual capacity of \((u, v)\) is

\[
c_f(u, v) = \begin{cases} c(u, v) - f(u, v), & (u, v) \in E \\ f(v, u), & (v, u) \in E \\ 0, & \text{otherwise} \end{cases}
\]

2. Given a flow network \( G = (V, E) \) and a flow \( f \) in that network, the residual network of \( G \) induced by \( f \) is \( G_f = (V, E_f) \), where

\[
E_f = \{(u, v) \in V \times V : c_f(u, v) > 0\}.
\]

The edges in \( E_f \) are called residual edges.

3. A flow on the residual network is function \( f' : V \times V \to \mathbb{R}^+ \) that satisfies the capacity and flow conservation constraints of \( G_f \): (a) \( 0 \leq f'(u, v) \leq c_f(u, v) \) for all \((u, v) \in E_f\), and (b) \( \sum_{v \in V} f'(u, v) = \sum_{v \in V} f'(v, u) \) for all \( u \in V \).

4. Let \( f \) be a flow in flow network \( G \) and \( f' \) be a flow in the corresponding residual network \( G_f \). Then \( f \uparrow f' \), the augmentation of \( f \) by \( f' \), is a function \( f \uparrow f' : V \times V \to \mathbb{R} \) via defined by

\[
(f \uparrow f')(u, v) = \begin{cases} f(u, v) + f'(u, v) - f'(v, u), & (u, v) \in E \\ 0, & \text{otherwise} \end{cases}
\]

Observations:

1. For any \( u, v \in V \times V \), exactly one of the three conditions in the definition of \( c_f(u, v) \) holds.

2. \(|E_f| \leq 2|E|\), since the definition of \( c_f(u, v) \) introduces two residual capacities for each \((u, v) \in E\) and the definitions of \( E_f \) excludes edges for which \( c_f = 0 \).
Graph with flow $f$, $|f| = 19$

Residual graph $G_f$

Residual graph $G_f$ with flow $f'$

Graph $G_f$ with augmented flow $f \uparrow f'$

$f' = 0$ where not specified

$|f \uparrow f'| = 23$
Lemma 26.1: Given $G, f$ and corresponding $G_{f}, f'$, $f \uparrow f'$ is a flow on $G$ and $\| f \uparrow f' \| = \| f \| + \| f' \|$.

Proof:

Capacity constraint: For $(u, v) \in E$, we have $c_{f}(v, u) = f(u, v)$, implying $f'(v, u) \leq c_{f}(v, u) = f(u, v)$. Then

$$
(f \uparrow f')(u, v) = f(u, v) + f'(u, v) - f'(v, u) \geq f(u, v) + f'(u, v) - f(u, v) = f'(u, v) \geq 0.
$$

Also,

$$
(f \uparrow f')(u, v) = f(u, v) + f'(u, v) - f'(v, u) \leq f(u, v) + f'(u, v) \leq f(u, v) + c_{f}(u, v) = f(u, v) + c(u, v) - f(u, v) = c(u, v).
$$
Flow conservation constraint: for $u \in V \backslash \{s, t\}$,

$$
\sum_{v \in V} (f \uparrow f')(u, v) = \sum_{v \in V} [f(u, v) + f'(u, v) - f'(v, u)]
$$

$$
= \sum_{v \in V} f(u, v) + \sum_{v \in V} f'(u, v) - \sum_{v \in V} f'(v, u)
$$

$$
= \sum_{v \in V} f(v, u) + \sum_{v \in V} f'(v, u) - \sum_{v \in V} f'(u, v)
$$

$$
= \sum_{v \in V} [f(v, u) + f'(v, u) - f'(u, v)]
$$

$$
= \sum_{v \in V} (f \uparrow f')(v, u).
$$
Finally, to compute \(|f \uparrow f'|\), we recall that \(G\) has no antiparallel edges, so for each \(v \in V\), there may be an edge \((s, v)\) or an edge \((v, s)\), but not both. No edge at all is also possible. We let \(V_1 = \{v : (s, v) \in E\}\), \(V_2 = \{v : (v, s) \in E\}\). Then \(V_1 \cup V_2 \subset V\) and \(V_1 \cap V_2 = \phi\). Now, since vertices outside \(V_1 \cup V_2\) have no connection to \(s\),

\[
|f \uparrow f'| = \sum_{v \in V} (f \uparrow f')(s, v) - \sum_{v \in V} (f \uparrow f')(v, s)
= \sum_{v \in V_1} (f \uparrow f')(s, v) - \sum_{v \in V_2} (f \uparrow f')(v, s)
= \sum_{v \in V_1} [f(s, v) + f'(s, v) - f'(v, s)] - \sum_{v \in V_2} [f(v, s) + f'(v, s) - f'(s, v)]
= \left(\sum_{v \in V_1} f(s, v) - \sum_{v \in V_2} f(v, s)\right) + \left(\sum_{v \in V_1} f'(s, v)\right) - \left(\sum_{v \in V_1} f'(v, s)\right).
\]

Since \(G\) has no antiparallel edges, \(v \notin V_1\) implies \(f(s, v) = 0\). Likewise, \(v \notin V_2\) implies \(f(v, s) = 0\). So, in the leftmost group above, both sums may be extended to \(V\) without changing the expressions. For a \(v \notin V_1 \cup V_2\), there is no edge \((s, v)\) or \((v, s)\) \(\in E\). Hence \(c(s, v)\) and \(c(v, s)\) are both zero, which forces \(f'(s, v)\) and \(f'(v, s)\) to be zero also. Therefore the final two sums can also be extended to all of \(V\). Then

\[
|f \uparrow f'| = \left(\sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s)\right) + \left(\sum_{v \in V} f'(s, v) - \sum_{v \in V} f'(v, s)\right)
= |f| + |f'|. \qed
\]
Definitions: Let $G = (V, E)$ be a flow network, and let $f$ be a flow in $G$. Let $G_f$ be the corresponding residual network.

1. In this context, an augmenting path in $G_f$ is a simple path from source $s$ to sink $t$.

2. If $p : s = u_0, u_1, \ldots, u_n = t$ is an augmenting path, the residual capacity of $p$ is given by
   \[
   c_f(p) = \min\{c_f(u_i, u_{i-1}) : 1 \leq i \leq n\}.
   \]

3. Given an augmenting path $p$, let $f_p : V \times V \to \mathcal{R}$ via
   \[
   f_p(u, v) = \begin{cases}
   c_f(p), & \text{if } (u, v) \text{ is on } p \\
   0, & \text{otherwise}.
   \end{cases}
   \]

   Note that $f_p(u, v)$ assumes only two values: $c_f(p)$ if $(u, v)$ is on $p$ or zero if $(u, v)$ is not on $p$.

Observation: Since $c_f(u, v) > 0$ for any $(u, v) \in E_f$, it follows that $c_f(p)$ is the minimum over a finite number of positive numbers. Therefore $c_f(p) > 0$. 
Lemma 26.2: Let $G = (V, E)$ be a flow network, let $f$ be a flow in $G$, and let $p$ be an augmenting path in the residual graph $G_f$. Then $f_p$ is a flow in $G_f$ with value $|f_p| = c_f(p) > 0$.

Proof: In view of the observation above, we need only prove that $f_p$ is a flow on $G_f$.

Suppose $(u, v)$ is in $p$. Then $f_p(u, v) = c_f(p) > 0$. Also, $f_p(u, v) = c_f(p) = \min\{c_f(x, y) : (x, y) \in p\} \leq c_f(u, v)$. Thus, $f_p$ satisfies the capacity constraint in $G_f$. For all vertices $u$ not on $p$, function $f_p$ assigns zero to all incoming and outgoing edges, thereby maintaining flow conservation. Each vertex $u$ on $p$ has one incoming edge with assignment $c_f(p)$ and one outgoing edge with the same assignment, which also maintains flow conservation. Consequently, $f_p$ is a flow on $G_f$.  \[\square\]
Corollary 26.3: Let \( G = (V,E) \) be a flow network, let \( f \) be a flow in \( G \), and let \( p \) be an augmenting path in residual network \( G_f \). Finally, let

\[
f_p(u,v) = \begin{cases} 
c_f(p), & (u,v) \text{ is on } p \\ 
0, & \text{otherwise.} 
\end{cases}
\]

Then \( f \uparrow f_p \) is a flow on \( G \) with value \(|f \uparrow f_p| = |f| + |f_p| > |f|\).

Ford-Fulkerson strategy for finding the maximum flow \( f \) in \( G \) starts with \( f = 0 \) on all edges and updates via \( f \uparrow f_p \) as long as augmenting paths \( p \) are available. Need to prove that when no more augmenting paths remain, we have achieved the maximum flow.
Definitions:

1. A cut \((S, T)\) of the flow network \(G = (V, E)\) is a partition of \(V\) into two sets, \(S\) and \(T = V \setminus S\) such that \(s \in S\) and \(t \in T\).

2. If \(f\) is a flow on \(G\), then the net flow across the cut \((S, T)\) is
   \[
   f(S, T) = \sum_{u \in S} \sum_{v \in T} f(u, v) - \sum_{u \in S} \sum_{v \in T} f(v, u).
   \]

3. The capacity of cut \((S, T)\) is
   \[
   c(S, T) = \sum_{u \in S} \sum_{v \in T} c(u, v).
   \]

4. A minimum cut of a flow network is a cut whose capacity is minimum over all cuts of the network.

Observation: The definitions of net flow across a cut and capacity of a cut are asymmetrical. Net flow across a cut is a algebraic sum: flow from \(S\) to \(T\) minus flow in the reverse direction. Cut capacity is capacity from \(S\) to \(T\), without any reference to capacity in the reverse direction. Intuitively, the capacity of any cut should be an upper bound on the value of any flow. The flow from \(s\) that does not return to \(s \in S\) must be absorbed by \(t \in T\) and must therefore not exceed the capacity of any cut.
Lemma 26.4: Let $f$ be a flow in flow network $G = (V, E)$, and let $(S, T)$ be a cut of $G$. Then $f(S, T) = |f|$.

Proof. For any $u \in V \setminus \{s, t\}$, flow conservation $\sum_{v \in V} f(u, v) - \sum_{v \in V} f(v, u) = 0$ gives

$$\sum_{u \in S \setminus s} \sum_{v \in V} f(u, v) = \sum_{v \in V} f(v, u) = 0,$$

because $t \not\in S$. Noting that $V = S \cup T, S \cap T = \phi$,

$$|f| = \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s)$$

$$= \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s) + \sum_{u \in S \setminus s} \left( \sum_{v \in V} f(u, v) - \sum_{v \in V} f(v, u) \right)$$

$$= \sum_{v \in V} \left( f(s, v) + \sum_{u \in S \setminus s} f(u, v) \right) - \sum_{v \in V} \left( f(v, s) + \sum_{u \in S \setminus s} f(v, u) \right)$$

$$= \sum_{v \in V} \sum_{u \in S} f(u, v) - \sum_{v \in V} \sum_{u \in S} f(v, u)$$

$$= \sum_{v \in S} \sum_{u \in S} f(u, v) + \sum_{v \in T} \sum_{u \in S} f(u, v) - \sum_{v \in S} \sum_{u \in S} f(v, u) - \sum_{v \in T} \sum_{u \in S} f(v, u)$$

$$= \sum_{v \in T} \sum_{u \in S} f(u, v) - \sum_{v \in T} \sum_{u \in S} f(v, u) + \left( \sum_{v \in S} \sum_{u \in S} f(u, v) - \sum_{v \in S} \sum_{u \in S} f(v, u) \right)$$

$$= \sum_{v \in T} \sum_{u \in S} f(u, v) - \sum_{v \in T} \sum_{u \in S} f(v, u) = f(S, T).$$

The last parenthesized term is zero because each double sum contains $f(x, y)$ exactly once for each $(x, y) \in S \times S$. \qed
Corollary 26.5: Flow $f$ in flow network $G$ has $|f|$ bounded above by the capacity of any cut of $G$.

Proof: Let $(S, T)$ be a cut with capacity constraint $c(S, T)$.

$$|f| = f(S, T) = \sum_{u \in S} \sum_{v \in T} f(u, v) - \sum_{u \in S} \sum_{v \in T} f(v, u) \leq \sum_{u \in S} \sum_{v \in T} f(u, v) \leq \sum_{u \in S} \sum_{v \in T} c(u, v) = c(S, T).$$
Theorem 26.6 (Max-flow min-cut theorem): If \( f \) is a flow in flow network \( G = (V, E) \) with source \( s \) and sink \( t \), then the following conditions are equivalent.

1. \( f \) is a maximum flow in \( G \).
2. The residual network \( G_f \) contains no augmenting paths.
3. \( |f| = c(S, T) \) for some cut \((S, T)\) of \( G \).

Proof. (1) \( \Rightarrow \) (2) by contradiction. Suppose \( f \) is a maximum flow and \( G_f \) contains an augmenting path \( p \). By Corollary 26.3, \( |f ↑ f_p| > |f| \), a contradiction.

(2) \( \Rightarrow \) (3): Suppose \( G_f \) has no augmenting path. That is, \( G_f \) has no path from \( s \) to \( t \). Let

\[
S = \{ v \in V : \text{there exists a path from } s \text{ to } v \text{ in } G_f \} \\
T = V \setminus S.
\]

Since \( s \in S, t \notin S \), \((S, T)\) is a cut. Consider vertices \( u \in S, v \in T \). We have three mutually exclusive possibilities: (a) \((u, v) \in E\), (b) \((v, u) \in E\), or (c) neither \((u, v)\) nor \((v, u)\) is in \( E \).

If \((u, v) \in E\), then \( f(u, v) = c(u, v) \), since otherwise \((u, v) \in E_f \) placing \( v \in S \).

If \((v, u) \in E\), then \( f(v, u) = 0 \), since otherwise, \( c_f(u, v) = f(v, u) > 0 \) placing \((u, v) \in E_f \) and consequently \( v \in S \).

If \((u, v) \notin E \) and \((v, u) \notin E \), then \( f(u, v) = 0 \).

\[
f(S, T) = \sum_{u \in S} \sum_{v \in T} f(u, v) - \sum_{v \in T} \sum_{u \in S} f(v, u) = \sum_{u \in S} \sum_{v \in T} f(u, v) = \sum_{u \in S} \sum_{v \in T} c(u, v) = C(S, T).
\]

(3) \( \Rightarrow \) (1): Suppose \( |f| = c(S, T) \) for some cut \((S, T)\) By Corollary 26.5, we have \( |g| \leq c(S, T) = |f| \), for any flow \( g \). Therefore, \( f \) is a maximum flow. \( \blacksquare \)
Example.

$G = G_f$, $|f| = 0$

$|f| = 4$

$|f| = 8$

$|f| = 12$

$|f| = 12$

$|f| = 23$

no augmenting paths
The Ford-Fulkerson Algorithm.

Design outline.

1. Let $E' = \{(u, v) \in V \times V : (u, v) \in E \text{ or } (v, u) \in E\}$. Maintain an adjacency list representation of $G' = (V, E')$. $|E'| = 2|E|$. Adding transpose edges to the adjacency lists is linear ($\Theta(V + E)$).

2. Attributes for edges in $E'$.

   $$(u, v).\text{isEdge} = \begin{cases} \text{true, } (u, v) \in E \\ \text{false, } (u, v) \notin E \end{cases}$$

   $$(u, v).f = \begin{cases} \text{flow, } (u, v) \in E \\ 0, \quad (u, v) \notin E \end{cases}$$

   $$(u, v).c = \begin{cases} \text{capacity constraint, } (u, v) \in E, \quad \text{(available from input graph)} \\ 0, \quad (u, v) \notin E \end{cases}$$

   $$(u, v).\hat{c} = \text{residual capacity} = \begin{cases} (u, v).c - (u, v).f, \quad (u, v) \in E \\ (v, u).f, \quad (u, v) \notin E \end{cases}$$

3. Structure of $(G, E')$ adjacency list.

4. Recall $(u, v) \in G_f$ if and only if $(u, v).\hat{c} > 0$. We ignore such edges in searching for a path in $G_f$. 
Ford-Fulkerson($G, s, t$) {
    Establish ($G' = (V, E')$);
    for each edge $(u, v) \in G.E'$ {
        $(u, v).f = 0$;
    }
    Update-Residual($G'$);
    while $\exists p: s \leadsto t$ in $G_f$ {
        $x = \min\{c_f(u, v) : (u, v) \in p\}$
        for each edge $(u, v) \in p$
            if $(u, v) \in E$
                $(u, v).f = (u, v).f + x$;
            else
                $(v, u).f = (v, u).f - x$;
            Update-Residual($G'$);
    }
}

Update-Residual($G, E'$) {
    for $v \in V$
        for $u \in v.\text{adj}$
            if $(u, v).\text{isEdge}$
                $(u, v).\hat{c} = (u, v).c - (u, v).f$;
            else
                $(u, v).\hat{c} = (v, u).f$;
}

Observations.

1. If all capacity constraints are integers, then since the initial flow is zero (an integer), all edge attributes will remain integers throughout the execution of the program. If the capacity constraints are all rational numbers, they are all integer multiples of some minimal rational. We need only work with the multiples as capacity constraints in the algorithm, and consequently all edge attributes again remain integral. Each iteration then increases the flow by at least one unit, and therefore the algorithm must converge.

2. We achieve $O(E)$ in the while-loop body since we expend $\Theta(1)$ time on each edge of $E'$, which is of size $2E$. With integer capacity constraints, the loop increases $|f|$ by at least one in each iteration, which implies $T(n) = O(|f^*| \cdot E)$, where $f^*$ is a maximum flow. This conclusion assumes that we can choose an augmenting path in $O(|f^*| \cdot E)$ time.
The above poor choice of an augmenting path takes 2,000,000 iterations to achieve the maximum flow $|f^*| = 2,000,000$. 
The Edmonds-Karp Algorithm.

The Edmonds-Karp Algorithm improves Ford-Fulkerson by using a breadth-first search to find an augmenting path with the shortest number of links. We will show that it achieves $O(VE^2)$ time complexity.