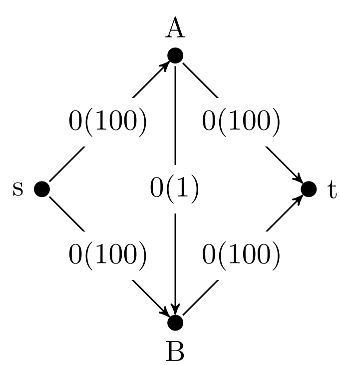
1 Flows, continued

1.1 The Ford-Fulkerson algorithm

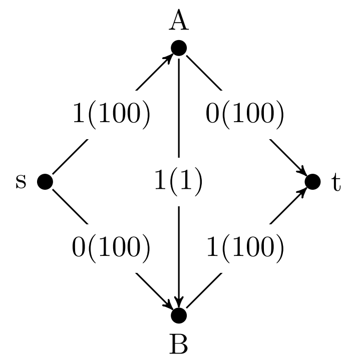
The proof above contains, as a subtle sidenote, the outline of an explicit algorithm for constructing a maximal flow. Note that there is only one place in the proof above where f's ostensible maximality is invoked: it is used to guarantee that the S-construction procedure does not include t, since if the procedure did include t, it would induce a path on which improvements to f could be made. Instead of phrasing this line of thought as a proof, we could phrase it as an algorithm: we probe outwards from s using the technique described in the min-flow max-cut theorem, and, if our probes reach t, we can slightly improve our flow and begin probing all over again. We can explicitly lay out this algorithm:

1. Let f be a flow (for simplicity and algorithmic explicitness, we might start with the zero flow in which $f(e) = 0$ for all edges e).
2. Let $s \in S$.
3. For any $x \in S$ from which we have not yet "probed", consider each neighbor $y \notin S$ of x. If either $x \to y$ with $f(x, y) < c(x, y)$ or if $y \to x$ with $f(y, x) > 0$, then put y in S, recording that its "parent" is x.
4. If $t \in S$, then skip to step 7.
5. If there are still unprobed vertices in S, go back to step 3.
6. There are no unprobed vertices in S and $t \notin S$, so f is a maximal flow!
7. Identify t as x_0, t's parent as x_1, the parent of x_1 as x_2, and so forth, until reaching $x_k = s$.
8. For each i, let $q_i = c(x_{i+1}, x_i) - f(x_{i+1}, x_i)$ or $f(x_i, x_{i+1})$ as appropriate. Let $q = \min q_i$.
9. For each i, either increment $f(x_{i+1}, x_i)$ by q or decrement $f(x_i, x_{i+1})$ by q, as appropriate.
10. With this improved flow f defined, forget which elements are in S and return to step 2.

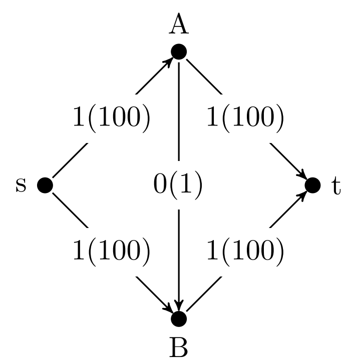
This algorithm, although guaranteed to find ever better flows until it reaches a maximum, is not without its flaws. The following weighted graph with initial zero choice of f represents a canonically problematic graph:



Following the procedure above, we might probe from s to A, from A to B, and B to t. We construct the path $s \to A \to B \to t$ with $q = 1$, leading to the slightly better flow



But now, supposing we probe from s to B, B to A,a nd A to t, we produce the path $s \to B \leftarrow A \to t$ with $q = 1$, which is a rather disappointingly tiny improvement:



And, now, if we once again quested along our original path, we would get another meager improvement of 1 in our value. Despite having obvious high-capacity paths, if our augmentation always makes use of the bottleneck, our algorithm will approach the maximal flow very slowly.

There are improvements to the Ford-Fulkerson algorithm to combat this problem, such as keeping a running total at all vertices of what their apparent "q" is, and prioritizing the probing and augmentation steps at vertices with high q-values. This is, however, a bit outside the scope of our studies, which seek mostly to determine that such an algorithm exists at all, not to provide its refinements.

A useful subordinate result to the algorithm is that its explicit construction makes use of augmentations of size q at each stage, and that q belongs to the same class of numbers as the capacities do.

Corollary 1. *If every capacity in a weighted graph is an integer (rational number), then there exists a maximum flow in which each edge has integral (rational) flow.*

Proof. In the initial step of the Ford-Fulkerson algorithm, every flow is zero, and thus an integer. We shall inductively assume that at the beginning of a single runthrough of steps 2–10, every $f(x, y)$ is an integer (or a rational number). During a single augmentation stage, q is calculated as equal to either some $f(x, y)$ or some $c(x, y) - f(x, y)$, which, by our inductive assumption and constraints on $c(x, y)$, must be an integer (rational number). Thus, our new f, which results from adding and subtracting q from particular $f(x, y)$, must also be integral (rational) on each edge. □

1.2 Consequences of Max-Flow Min-Cut on Edges

Structurally, we are already aware that the Max-Flow Min-Cut Theorem closely resembles our other dual-concept exploring theorems. In fact, Max-Flow Min-Cut is in many ways a prototypical function, since all of the other results can be contextualized as flow results.

Theorem 1 (König-Egerváry, '31)**.** *A bipartite graph G contains a matching M and set-cover S such that $|M| = |S|$.*

Proof by way of Max-Flow Min-Cut. For a bipartite graph $G = (A, B)$, let D be a digraph such that $V(D) = V(G) \cup \{s, t\}$, and such that for $a \in A$ and $b \in B$, $a \sim b$ in G if and only if $a \to b$ in D. In addition, let $s \to a$ for all $a \in A$, and $b \to t$ for all $b \in B$. Let the edges all have capacity 1.

Let f be a maximal integral flow in D. For each edge (s, a), $f(s, a)$ must be 0 or 1. Since (s, a) is the only incoming edge to a, the outflow from a must be exactly 1, and since the flow is integral, that outflow must all be along a single edge (a, b). Likewise, since b has an outflow of at most 1, along the edge (b, t), it can only have one incoming edge carrying flow. Thus, we know that if $f(a, b) = 1$, then $f(a, b') = f(a', b) = 0$ for any $a' \neq a$ and $b' \neq b$, so the edges carrying flow between A and B have no endpoints in common, so they form a matching M in G; it then follows that $|f| = |M|$.

By the Max-Flow Min-Cut Theorem, there is also a cut (S, T) of capacity $|M|$, which we construct via the iterative development of S used in the Ford-Fulkerson method. Let $R = (T \cap A) \cup (S \cap B)$. We shall see that R is a vertex cover of G and that $|R| = c(S, T)$.

Let $e = \{a, b\}$ be an edge in G. If $a \in T$ or $b \in S$, then R covers e. However, if $a \in S$ and $b \in T$, then it must be the case that $f(a, b) = 1$, or the Ford-Fulkerson construction would have put b into S. Since $f(a, b) = 1$, it must be the case that a has inflow of at least 1, so $f(s, a) = 1$. The step of the Ford-Fulkerson process that chose a to be in S must have made use of some $u \in S$ and either $u \to a$ with flow zero, or $u \leftarrow a$ with flow one. We have determined that the only incoming edge to a has flow one, so there must be some $u \in S$ such that $f(a, u) = 1$. However, since a has inflow of 1 total and already has an outflow of 1 to b, this cannot happen. Thus, there are no edges $\{a, b\}$ in G where $a \in S$ and $b \in T$.

This same fact will help us see that $|R| = c(S, T)$. Since none of the edges (a, b) in D are from S to T, the only edges which contribute capacity towards $c(S, T)$ are edges of the form (s, a) where $a \in T$ and (b, t) there $b \in S$. Since each of these edges has capacity 1, the total capacity of the cut $c(S, T)$ is equal to the number of such edges, which is $|T \cap A| + |S \cap B| = |R|$. □

Theorem 2 (Menger, '27 (directed and undirected edge version))**.** *If s and t are vertices of a (di)graph G, then there is a (s, t)-separating set of edges S and collection of edge-disjoint directed paths $\mathcal{P}$ from s to t, such that $|S| = |\mathcal{P}|$.*

Proof. First, let us show that this question on an undirected graph can be reduced to the directed graph case. For an undirected graph G, let us build a directed graph D via the most straightforward procedure: let $V(D) = V(G)$, and for every edge $\{u, v\} \in E(G)$, let (u, v) and (v, u) be in $E(D)$.

Then it is clear that a collection of edge-disjoint paths in G can be easily mapped to a collection of edge-disjoint paths in D; the reverse is somewhat less obvious, since what if two distinct paths used (u, v) and (v, u), which are disjoint in D but not in G? Fortunately, it can be show that this is not particularly problematic: suppose $P_1 = s \Longrightarrow u \to v \Longrightarrow t$ (where $\Longrightarrow$ represents a directed path, consisting of zero or more directed edges), and $P_2 = s \Longrightarrow v \to u \Longrightarrow t$. The four constituent subpaths are from s to u, v to t, s to v, and u to t, and we can recouple them to form *directed walks* which do not use either (u, v) or (v, u): $W_1 = s \Longrightarrow u \Longrightarrow t$ and $W_2 = s \Longrightarrow v \Longrightarrow t$. Using the techniques presented last semester to create a path from a walk, we can prune out any self-intersections: since this only makes the walks shorter, it will preserve the edge-disjointness properties, so we can get new paths P_1' and P_2' which do not use (u, v) or (v, u). Repeating for every pair of opposite-orientation edges, we can modify any collection of edge-disjoint paths in D to not use opposite-orientation edges, and thus remain edge-disjoint even on projection into D.

Showing that minimal (s, t)-separation in D uses the same number of edges as minimal $\{s, t\}$-separation in G is somewhat easier. Any minimal separating set of edges in G separates G into components S and T (if it created more components, it would not be minimal). To project this set onto D, all that is necessary is to use the edges of orientation $S \to T$. Thus, the smallest separating set in D is the same size as the smallest separating set in G.

Thus, we may prove Menger's Theorem on edges in the undirected case as a special case of Menger's Theorem on *arbitrary* directed graphs. Having drawn this correspondence, the directed case becomes absurdly easy: let s be a source, t a sink, and give every edge capacity 1. Let f be a maximal integer flow, with $|f| = k$. Each unit of flow describes a walk along disjoint edges of flow 1 from s to t; thus there are k edge-disjoint directed walks from s to t. Paring off cyclic sections as mentioned above, these become k edge-disjoint paths.

The Max-Flow Min-Cut Theorem tells us that, in addition, there is a partition (S, T) such that $c(S, T) = k$. Thus, by the definition of the capacity and the fact that $c(e) = 1$ for all edges, there must be k edges from S to T. Removal of these k edges will separate S from T, and thus s from t. □

1.3 Consequences of Max-Flow Min-Cut on Vertices

At first glance, the Max-Flow Min-Cut Theorem seems ill-suited to vertex-based results, since its statements all concern the utilization and removal of edges. However, with some cleverness, we can actually make vertices "masquerade" as edges:

Theorem 3 (Menger, '27 (directed and undirected vertex version))**.** *If s and t are nonadjacent vertices of a (di)graph G, then there is a (s, t)-separating set of vertices S and collection of independent directed paths $\mathcal{P}$ from s to t, such that $|S| = |\mathcal{P}|$.*

Proof. As in the edge case, we can trivially convert a graph G to a digraph D by replacing every undirected edge with a pair of directed edges. Our previous modifications to justify this are not even necessary, as a collection of vertex-disjoint paths could not use both edges between vertices.

Now, given a digraph D and pair of vertices s and t, let us construct a digraph D' via splitting every vertex except s and t: For $v \in V(D)$, let v' and v'' be the identities of vertices in D'; let us simply include s and t undoubled in D', although for brevity of presentation we will use the names s, s', and s'' to refer to the same point (likewise for t, t', and t''). We now shall add several edges to D': for every vertex $v \neq s, t$, we add the edge $e_v = (v', v'')$, and for every edge $(u, v) \in E(D)$, we add to D' the edge (u'', v').

The result of this construction is that removing the edge e_v has the same connectivity effect that removing v would have had in the original graph: any path $u \to v \to w$ in D would be represented in D' as $u'' \to v' \to v'' \to w'$, so removal of the edge e_v would have the same effect on this path that removal of v had on the path from which it was mapped.

Let s and t be the source and sink respectively in D', and give every edge capacity of 1. If f is a maximal integer flow with $|f| = k$, we know, as seen in the edge-version above, that there are k edge-disjoint paths on D', and that there is a set of k edges in D' separating s from t.

Edge-distinct paths in D' must correspond to independent paths in D: a path in D' must, by the extremely limited number of outflows from v' vertices and limited inflows to v'' vertices, take on the form

$$s \to v_0' \to v_0'' \to v_1' \to v_1'' \to \cdots \to v_\ell' \to v_\ell'' \to t$$

which corresponds to the D-path $s \to v_0 \to v_1 \to \cdots \to v_\ell \to t$.Since in a collection of edge-disjoint paths in D', only one path can use any particular e_v, it follows that in the associated collection of paths in D, only one path visits v. Thus there are k independent paths in D.

Similarly, if we map our k separating edges in D' up to separating structures in D, we can map the edges (v', v'') to the vertices v, and (u'', v') to the edge (u, v), so we have a set of k objects, some of which are vertices and some of which are edges, separating s from t. □

Just to round out our collection of theorem's, here's a max-flow min-cut proof of Dilworth's theorem. Note that this would normally be proven by appeal to König-Egerváry rather than Max-Flow Min-Cut:

Theorem 4 (Dilworth '48)**.** *If A is a largest antichain in a finite poset $(X, \preceq)$, then there is a partition of X into chains $C_1 \cup C_2 \cup \cdots C_n$ such that $n = |A|$. Furthermore, each C_i contains exactly one element of A, and there is no partition of X into fewer than n chains.*

Proof. From the poset $(X, \preceq)$, we construct the digraph D whose vertices are a source vertex s, a sink vertex t, and for each $a \in X$, two vertices a' and a''. We add edges in the following manner: for every $a \in X$, we have the edge (s, a') and (a'', t), and for $a \preceq b$ and $a \neq b$, we have the edge (a', b''). Each edge will be assigned weight 1.

Let f be a maximal flow in D, and (S, T) a minimal cut, so that $|f| = c(S, T)$. Every path from s to t is, by construction, of the form $s \to a' \to b'' \to t$, so $|f|$ is equal to the number of vertices a' such that $f(s, a') = 1$ and some $f(a', b'') = 1$. Let B be the set of $a \in X$ such that $f(s, a') = 0$; clearly $|B| = |X| - |f|$. We shall see that X is partitioned into $|B|$ chains. For each value a outside of B in X, there is a unique b such that $f(a', b'') = 1$. Thus each $a \in X \setminus B$ is associated with a unique distinguished successor; percolating upwards from any given a_0 along this sequence of succession, we produce a chain $a_0 \preceq a_1 \preceq a_2 \preceq \cdots \preceq a_n$, where $a_n \in B$, since only elements of B do not have successors defined by this flow. Since no two distinct elements can flow into a single b'', successors are guaranteed to be unique, so this procedure decomposes X into disjoint chains, each of whose maximal elements is in B, so X has a chain decomposition of size $|B|$.

Now we will build an antichain using the cut (S, T). Specifically, we let A consist of those values $a \in X$ such that $a' \in S$ and $a'' \in T$. Using the Ford-Fulkerson-crafted properties of (S, T), we shall show that $|A| = |X| - c(S, T)$ and that elements of A are incomparable. For the second assertion, let us consider $a, b \in A$. Since $a' \in S$ and $b'' \in T$, it cannot be the case that (a', b'') is an edge with zero flow. Thus either (a', b'') is not an edge, in which case $a \not\leq b$, or $f(a', b'') = 1$. If $f(a', b'') = 1$, then it would necessarily follow that $f(s, a') = 1$, so in order for a' to be in S, there must be a $c'' \in S$ such that $f(a', c'') = 1$, which would necessarily unbalance flow at a'. Thus, it is impossible that $f(a', b'') = 1$, so there must be no edge (a', b''). A similar argument serves to show that there is no edge (b', a''), so a and b are incomparable.

A modified version of the argument seen in the proof of the König-Egerváry Theorem will show that $|A| = |X| - c(S, T)$. We saw there that edges of the form (a', b'') cannot be from S to T, so the only edges which contribute capacity towards $c(S, T)$ are edges of the form (s, a') where $a \in T$ and (a'', t) where $a \in S$. This specifically excludes exactly those a where $a' \in S$ and $a'' \in T$; in other words, it excludes the members of A. Thus, $c(S, T) = |X| - |A|$, which can be algebraically reformed into our desired value for $|A|$. □

2 Directed Graphs

In studying flows, we introduced, nearly incidentally, the concept of a directed graph. As mentioned previously, directed graphs have a number of the same concepts as undirected graphs have. We have encountered directed walks, paths, and degrees, and we can easily derive from those concepts directed notions of trails, tours, cycles, Eulerian tours, Hamiltonian cycles, subgraphs, and suchlike. Many of these ideas can be fully described and characterized with only minor variations on their undirected counterparts. There are a few concepts which have unusual and different representations in a directed regime:

Definition 1. A digraph is acyclic if it does not have a directed cycle as a subgraph.

An acyclic digraph is a structure somewhat different from an acyclic graph; while we have a good visual conception of what a forest must look like, an acyclic digraph might have what appear to be cycles on a visual inspection, but in which the edges are not oriented so as to give a directed cycle. We can, however, develop somewhat analogous rules for acyclicity in digraphs to our tree properties. For instance, akin to our criterion that every tree has a leaf, we can see that every acyclic digraph has a node with no outgoing edges and a node with no incoming edges.

Proposition 1. *Let D be a finite acyclic digraph with at least one vertex. There are some $u, v \in V(D)$ such that $d^-(u) = 0$ and $d^+(v) = 0$.*

Proof. Suppose to the contrary there is no such v, so that every vertex has outdegree 1. Then, choosing an arbitrary v_0, since $d^+(v_0) \geq 1$, there is a vertex v_1 such that $v_0 \to v_1$. Likewise, there must be a v_2 such that $v_1 \to v_2$. This procedure can be continued indefinitely to get an arbitrarily long (or even infinite) sequence $v_0 \to v_1 \to v_2 \to \cdots$. Since D is finite, this sequence must have a repetition. Let j be the least index such that for some $i < j$, $v_i = v_j$. Then $v_i \to v_{i+1} \to \cdots \to v_{j-1} \to v_j = v_i$ is clearly a closed walk, and by minimality of j, all the $v_i, v_{i+1}, \ldots, v_{j-1}$ must be distinct, so this closed directed walk is a directed cycle, contradicting the acyclicity of D.

A similar proof serves to show the existence of u with indegree zero. □

Much like a tree can be crafted by adding leaves to a basic case, we can structurally assemble an acyclic digraph by successively adding vertices without inflows:

Theorem 5. *Every acyclic digraph D can be assembled via a sequence $D_1 \subset \cdots \subset D_n = D$, where D_1 is an isolated vertex and D_i is produced by adding to D_{i-1} a vertex v_i and zero or more edges from v_i to the vertices of D_{i-1}.*

Proof. It is fairly easy to show by induction that the graph D_i produced in the above procedure is acyclic. Clearly this is true for the base case D_1. For our inductive step, let us inductively assume D_{i-1} is acyclic; thus, if D_i contains a cycle, it must use one of the newly introduced edges. However, since $d^-(v_i) = 0$, it is impossible for a directed cycle to pass through v_i, since such a cycle would need a segment $u \to v_i \to v$, and no such u exists in D_i. Thus, D_i contains no directed cycle.

We can likewise show by induction on the number of vertices that any acyclic D can be produced by the above procedure, using a number of steps equal to the number of vertices in D. Clearly this procedure generates all acyclic digraphs with $|D| = 1$. Now for arbitrary acyclic D where $|D| = n$, we know from the previous proposition that D contains a vertex v_n with $d^-(v_n) = 0$. Clearly $D - v_n$ is itself acyclic, since an acyclic graph will have acyclic subgraphs, so by our inductive hypothesis, there is a sequence $D_1 \subset \cdots \subset D_{n-1} = D - v_n$. Now, note that D can be formed by adding the vertex v_n to $D - v_n$, and, since v_n has indegree zero, also adding some number of edges from v_n to $D - v_n$. □

This structural theorem is rarely used, but the underlying conceit of it frequently is. Using the enumerative labeling of vertices as v_i above, we see that any edge between v_i and v_j is oriented from higher indices to lower. The most straightforward characterization of an acyclic graph is, in practice, to apply an order-induced directionality on each edge to an undirected graph.

Definition 2. An *orientation* of an undirected graph G is a graph D such that $V(G) = V(D)$, and for each $\{u, v\} \in E(G)$, exactly one of (u, v) or (v, u) is in $E(D)$.

Corollary 2. *A directed graph D is acyclic if and only if there is an ordering of its vertices $v_1, v_2, \ldots, v_n$ such that it is an orientation of an undirected simple graph G given by $v_i \to v_j$ only if $i > j$.*

Proof. First we shall show that every acyclic graph has such a vertex-ordering. Let us use the vertex-ordering induced by the structural theorem and show inductively that every step of the structural theorem preserves the above orientation property. Clearly D_1 has no edges and satisfies the above orientation. For an arbitrary D_{i-1} satisfying the above orientation, D_i is constructed by adding a vertex v_i and edges from v_i to various v_j in D_{i-1}. Since $v_j \in D_{i-1}$, $j \leq i - 1$, so every edge newly introduced in D_i is an edge of the form $v_i \to v_j$ where $j < i$, so D_i satisfies the above orientation.

Conversely, consider an undirected G and an ordering of vertices $v_1, \ldots, v_n$; this ordering induces an orientation D of G. Suppose D has a directed cycle of length k; since we don't know the indices of various elements of the cycle, we shall denote them $v_{i_1} \to v_{i_2} \to v_{i_3} \to \cdots \to v_{i_k} \to v_{i_1}$. Our orientation would thus require that $i_1, \ldots, i_k$ satisfy the inequality $i_1 > i_2 > i_3 > \cdots > i_k > i_1$; the transitivity-induced result $i_1 > i_1$ is clearly impossible, so no such cycle can exist. □

The easiest way to construct an acyclic graph, then, is to consider it as an arbitrary undirected graph with a ranking-induced orientation imposed on the vertices. Note that while some ranking induces every acyclic orientation of a given graph, there are very few orientations which have *unique* rankings associated with them. For instance, if we consider a three-vertex graph with the edges

$u \to v$ and $u \to w$, u would have to have rank 3, but v and w could have ranks 1 and 2 in any order. Here, we can consider a slight curiosity:

Proposition 2. *An acyclic digraph D has a unique vertex-ordering inducing its orientation if and only if it has a directed Hamiltonian path.*

Proof. If D has a directed Hamiltonian path, then there is an ordering of its vertices along said path: $v_n \to v_{n-1} \to \cdots \to v_2 \to v_1$. If our orientation-inducing ordering function is denoted $f : V(D) \to \{1, 2, 3, \ldots, n\}$, then the orientation of edges in the above path requires that $f(v_n) > f(v_{n-1}) > \cdots > f(v_2) > f(v_1)$; since this is a strictly decreasing sequence of n numbers from $\{1, 2, 3, \ldots, n\}$, it is uniquely determined that $f(v_i) = i$.

Conversely, if D is induced by a unique ordering $f(v_1) < f(v_2) < f(v_3) < \cdots < f(v_n)$, it follows that the slight modification of the ordering $f'(v_1) < f'(v_2) < \cdots < f'(v_{k-1}) < f'(v_{k+1}) < f'(v_k) < f'(v_{k+2}) < \cdots < f'(v_n)$ does *not* induce D. However, on any edge $\{v_i, v_j\}$ except $\{v_k, v_{k+1}$, it induces the exact same orientation. Thus, it must be the case that $(v_{k+1}, v_k) \in D$. Since this is true for arbitrary k, it follows that the edges $(v_n, v_{n-1}), (v_{n-1}, v_{n-2}), \ldots, (v_2, v_1)$ are in D, forming a Hamiltonian path. □

One concept closely related to acyclicity is transitivity, which is quite similar to the relation property of the same name.

Definition 3. A digraph D is transitive if, for any (u, v) and (v, w) in $E(D)$, $(u, w) \in E(D)$.

Proposition 3. *Any transitive simple digraph D is acyclic.*

Proof. Suppose D has a cycle $v_1 \to v_2 \to \cdots v_k \to v_1$. By transitivity, since $v_1 \to v_2$ and $v_2 \to v_3$, $v_1 \to v_3$. Since $v_1 \to v_3$ and $v_3 \to v_4$, $v_1 \to v_4$. Proceeding in this manner as long as is necessary, it will follow that $v_1 \to v_k$. But since $v_k \to v_1$, it must be the case that $v_1 \to v_1$, which violates simplicity. □