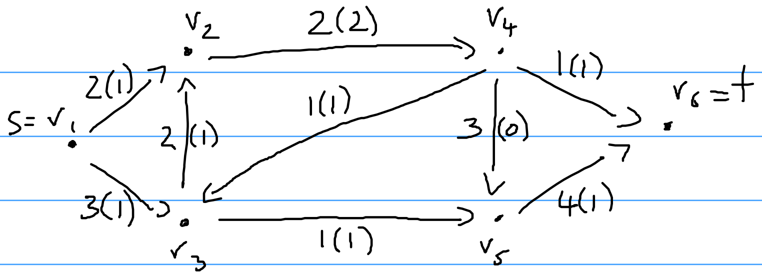
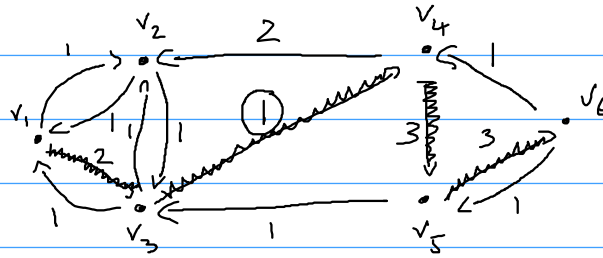


network
labels: capacity (flow)



residual network
labels: residual capacities

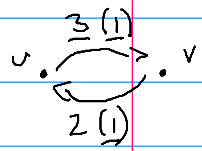
<u>Definition</u> Let (D,c) be a directed network, $s,t \in V(D)$. Let f be an s-t-flow of (D,c). Let $P = v_0 v_1 v_2 \dots v_m$ be a sequence of vertices in $V(D)$. Then P is an <u>f-augmenting s-t-path</u> if $v_0 = s$, $v_m = t$ and $c_f(v_{i-1}, v_i) > 0$ for all $i \in [m]$.
(in other words: it is a directed s-t-path in the residual network)

The residual capacity of P is $c_f(P) = \min_{i \in [m]} c_f(v_{i-1}, v_i)$.
An arc $e \in A(D)$ is a <u>forward arc</u> on P if it has tail v_{i-1} and head v_i for some $i \in [m]$ and a <u>backward arc</u> on P if it has tail v_i and head v_{i-1} for some $i \in [m]$.

<u>Lemma</u> Let (D,c) be a directed network, $s,t \in V(D)$, and f an s-t-flow in (D,c). Let P be an f-augmenting s-t-path. Let $F \subseteq A(D)$ and $B \subseteq A(D)$ be the forward and backward arcs on P. Let $g : A(D) \to \mathbb{R}$ such that

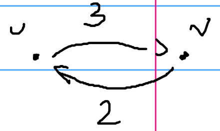
$$g(uv) = \begin{cases} \max\{f(uv) - c_f(P), 0\} & \text{if } uv \in B \\ f(uv) + \max\{c_f(P) - f(vu), 0\} & \text{if } uv \in F \text{ and } vu \in B \\ f(uv) + c_f(P) & \text{if } uv \in F \text{ and } vu \notin B \\ f(uv) & \text{if } uv \notin F \text{ and } vu \notin B \end{cases}$$

capacity (flow)



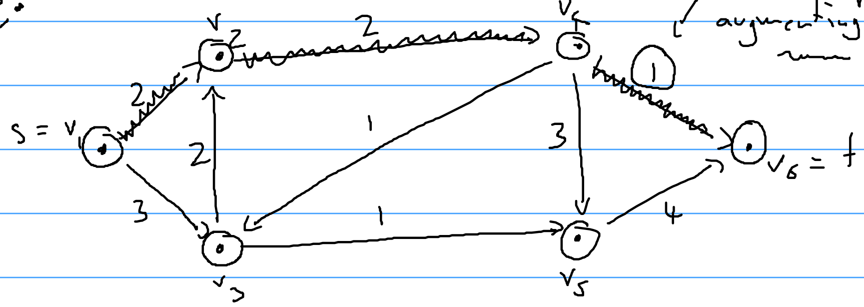
Then g is an s-t-flow of (D, c) and $|g| = |f| + c_f(P)$.

residual capacity

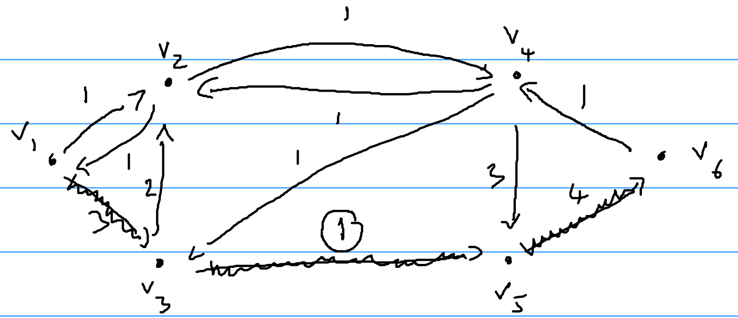


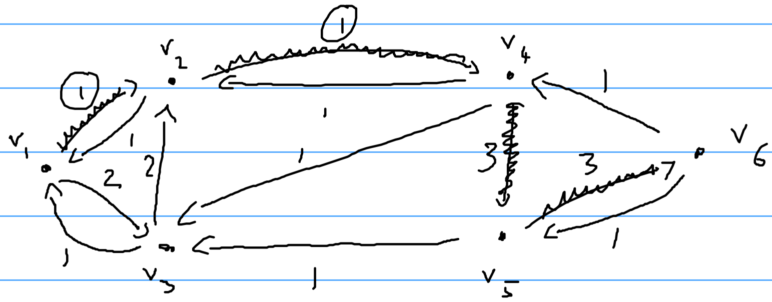
(5)

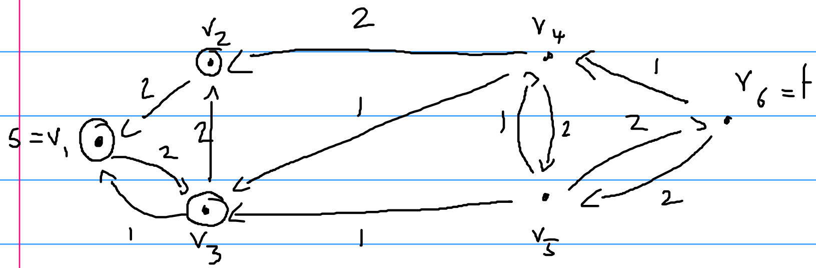
residual capacity of augmenting path



network with capacities (residual capacities for all-zero flow)







no more augmenting paths

Lemma Let (D,c) be a directed network, $s,t \in V(D)$ and f an s-t-flow of (D,c). Then exactly one of the following is true: (i) there exists an f-augmenting s-t-path in (D,c); (ii) there exists an s-t-cut S of (D,c) with $c(S) = |f|$.

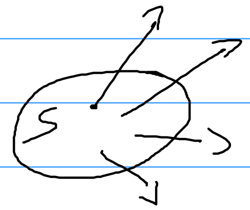
Proof. Let S be the set of all vertices v such that there exist an f-augmenting s-v-path in (D,c).

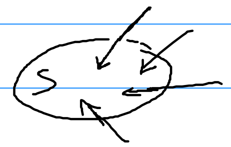
If $t \in S$, there exists an f-augmenting s-t-path in (D,c).

If $t \notin S$, then S is an s-t-cut. As before, let $F(U,W)$ denote the overall amount of flow along arcs with tail in U and head in W. Since f is an s-t-flow and S an s-t-cut, we know from Lemma 6.4 that

$$|f| = F(S, V(D)\setminus S) - F(V(D)\setminus S, S)$$
$$\leq F(S, V(D)\setminus S) \leq c(S).$$

We will now show that both inequalities hold with equality.

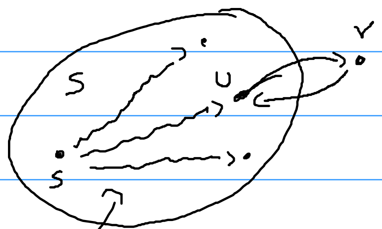




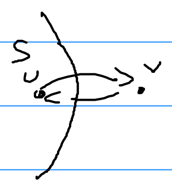
by definition of S

no augmenting path to v, so

$c_f(u,v) = 0$



f– augmenting paths



$$c_f(u,v) = \underbrace{c(u,v) - f(u,v)} + \underbrace{f(v,u)}$$

for $c_f(u,v)$ to be zero, both $c(u,v) - f(u,v)$ and $f(v,u)$ have to be zero

By definition of S for any arc e with tail in S and head in $V(D) \setminus S$, $f(e) = c(e)$. Therefore
$F(S, V(D) \setminus S) = C(S)$.

Also by definition of S, for any arc e with head in S and tail in $V(D) \setminus S$, $f(e) = 0$. Therefore
$F(V(D) \setminus S, S) = 0$.

Therefore $|f| = C(S)$, as claimed. $\square$

__Theorem__ (max-flow min-cut) Let (D,c) be directed network, $s,t \in V(D)$. Let f be a maximum s-t-flow of (D,c) and S a minimum s-t-cut of (D,c). Then $|f| = C(S)$.

6.4 The Ford Fulkerson Algorithm

__Algorithm__ Let (D,c) be a directed network, $s,t \in V(D)$. The Ford-Fulkerson algorithm starts from $f: A(D) \to \mathbb{R}$ with $f(e) = 0$ for all $e \in A(D)$ and then repeats the following steps:

1. Let P be an f-augmenting s-t-path of (D,c). If no such path exists, then stop: f is a maximum s-t-flow of (D,c).
2. Augment f by sending $c_f(P)$ along P. (Lemma 6.10)

__Theorem__ Let (D,c) be a directed network such that $c(e) \in \mathbb{N}_0$ for all $e \in A(D)$. Let $s,t \in V(D)$. Then there exists a maximum s-t-flow f such that $f(e) \in \mathbb{N}_0$ for all $e \in A(D)$.

(We will call such a flow an "integral" flow)

Proof. Initial flows and residual capacities are integral. When the residual capacities are integral, so is the residual capacity of any augmenting path. If the residual capacity of the augmenting path and the current flow are integral, then the new flow is also integral. Repeating the argument, we see that throughout the algorithm residual capacities and flows remain integral.
While the algorithm is running, in each iteration the size of the flow increases by at least 1 (the smallest positive integer). The size of the maximum flow is finite, so the algorithm stops after a finite number of iterations. When it stops, it does so with a maximum flow that is also an integral flow. □

If all capacities are rational numbers, the algorithm is also guaranteed to finish after a finite number of iterations.

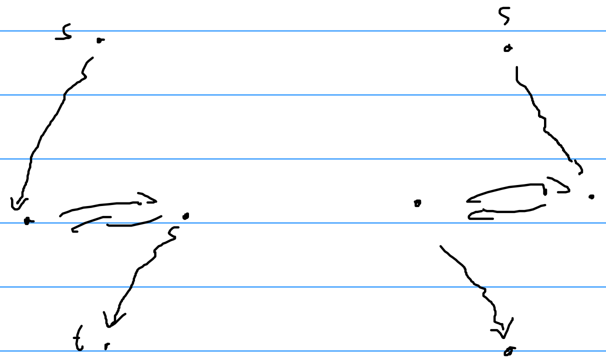
There exists a network with irrational capacities on which the algorithm runs forever (and increases the size of the flow by smaller and smaller amounts).

When capacities are integral, the running time of the algorithm is

$$O(\underbrace{|V(D)| \cdot |A(D)|}_{\text{tree search to find an augmenting path}} \cdot \underbrace{|f|}_{\text{upper bound on number of iterations}}),$$ where f is a maximum flow

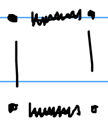
This is not necessarily polynomial in the size of the input because $|f|$ may be exponential in the amount of space we need to write down the capacities.

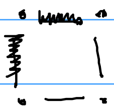
The running time can be shown to be polynomial, even when capacities are irrational, if we always choose shortest augmenting path (where length is measured in terms of the number of arcs).



7 Matchings

a matching



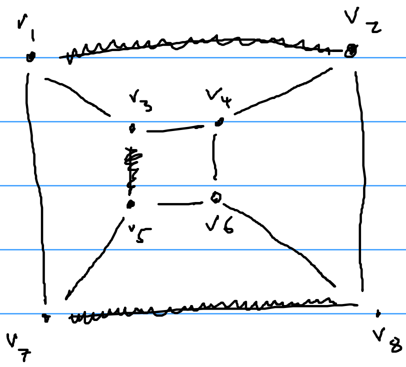


not a matching

Definition Let G be a graph. A set $M \subseteq E(G)$ is a <u>matching</u> of G if every vertex of G appears at most once as an endpoint of an edge in M. A matching M of G is a <u>maximum matching</u> of G if it has maximum cardinality among all matchings of G. A matching M <u>saturates</u> $X \subseteq V(G)$ if every $x \in X$ is the endpoint of an edge in M. A matching M is a <u>perfect matching</u> if it saturates $V(G)$.

Example

G



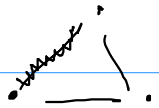
$M = \{v_1v_2, v_3v_5, v_7v_8\}$ is a matching of G

$\{v_1v_2, v_1v_3\}$ is <u>not</u> a matching of G

$\{v_1v_2, v_3v_7\}$ is <u>not</u> a matching of G

$\{v_1v_2, v_3v_4, v_5v_6, v_7v_8\}$ is a perfect matching, and therefore a maximum matching.

It also shows that M is not a maximum matching.



is a maximum matching, a perfect matching does not exist