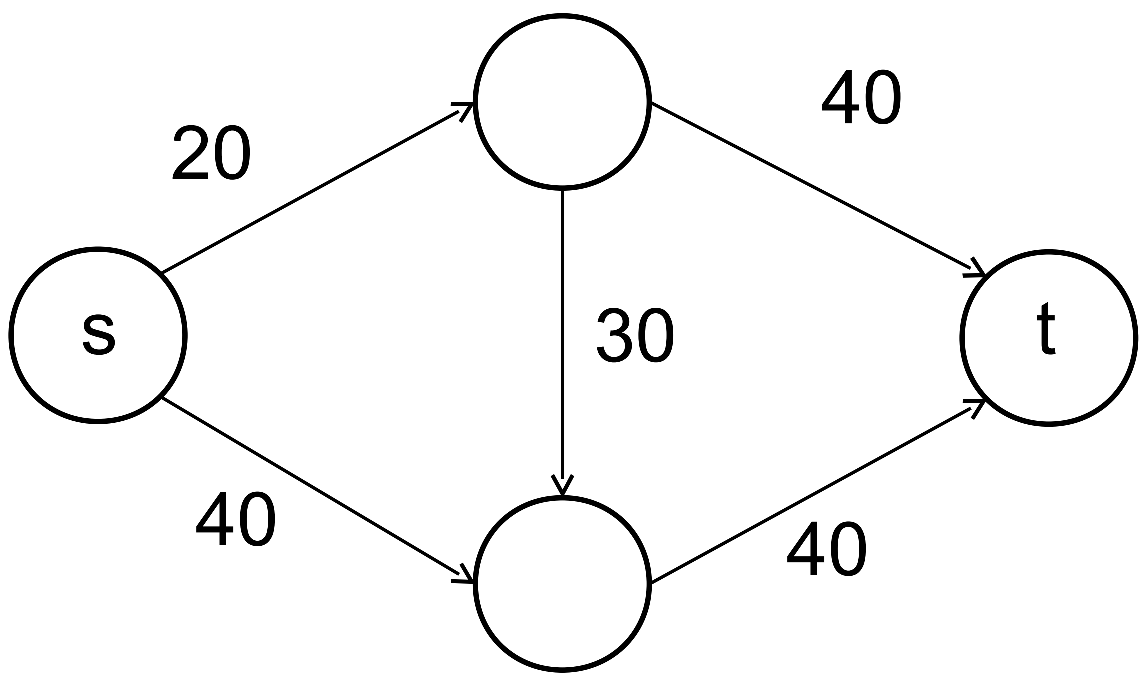
Network flow

Definition: A flow network is a directed graph $G = (V,E)$ with two nodes s and t, and a function $c(u,v) \geq 0$ on each directed edge (u,v)

- s is called the source
- t is called the sink
- $c: E \rightarrow R^+$ is called the capacity function

- Example



- A flow $f : V \times V \rightarrow R$ satisfies:

Skew symmetry: $f(u,v) = - f(v,u)$ for every pair (u,v)

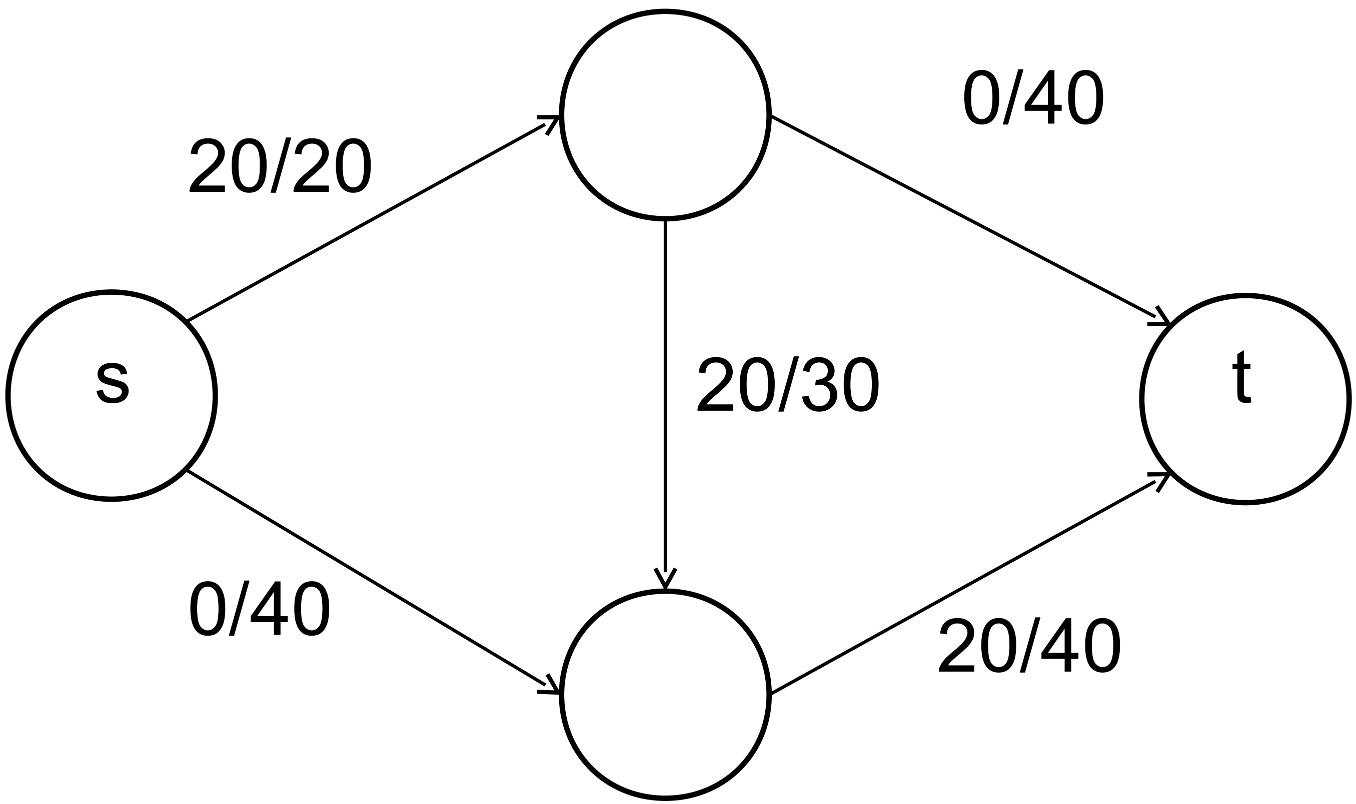
Capacity constraint $f(u,v) \le c(u,v)$ for each $(u,v) \in E$

Conservation of flows: $f(u,V) = 0$ for every $u \notin \{s,t\}$,
Where we define $f(X,Y) := \sum f(x,y)$ over $x \in X$ and $y \in Y$

- The value of flow f is $| f | = f(s,V)$

It represents the amount of flow passing from the source to the sink.

- Example



| f | = 20

Maximum flow problem

Input: A flow network G with s and t, a capacity function c
Output: A flow f so that | f | is maximum.

Applications: railway traffic, food supply, airline scheduling, image segmentation, baseball elimination...

Residual network

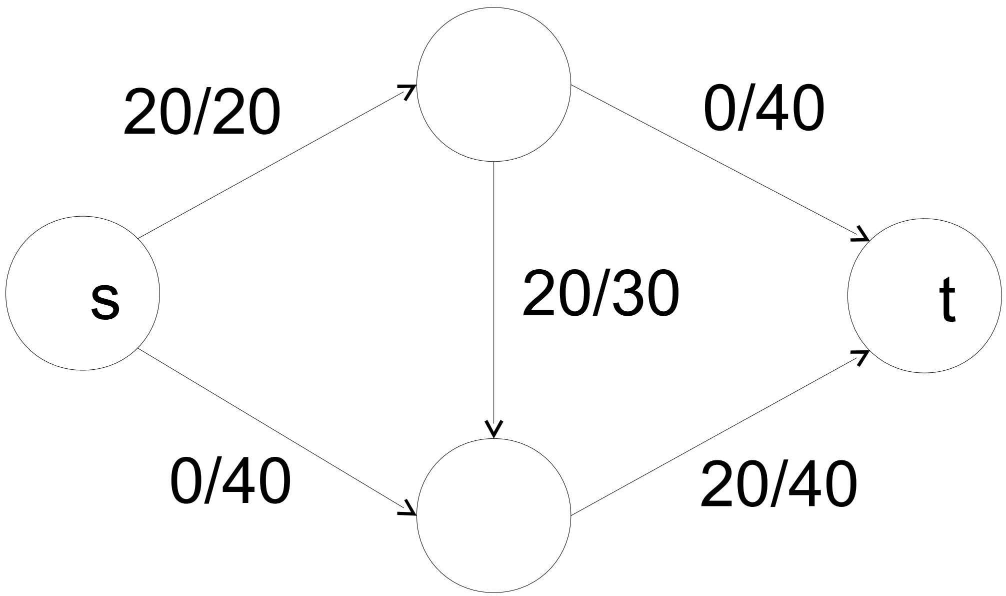
- A flow f induces a residual network G_f , consisting of the original graph G, and residual capacity function c_f :

For every (u,v) such that (u,v) or (v,u) $\in$ E we set $c_f(u,v) := c(u,v) – f(u,v) \geq 0$.

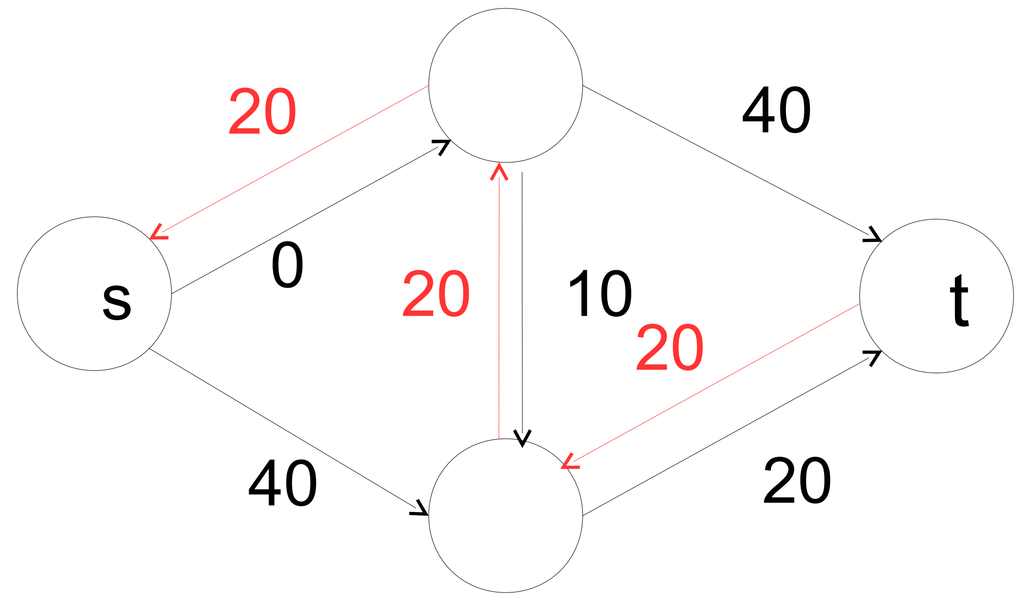
Note: the residual network may put non-zero capacity on edges which were non-existing or had zero capacity.

- An augmenting path is a path from s to t in the residual network

- Example

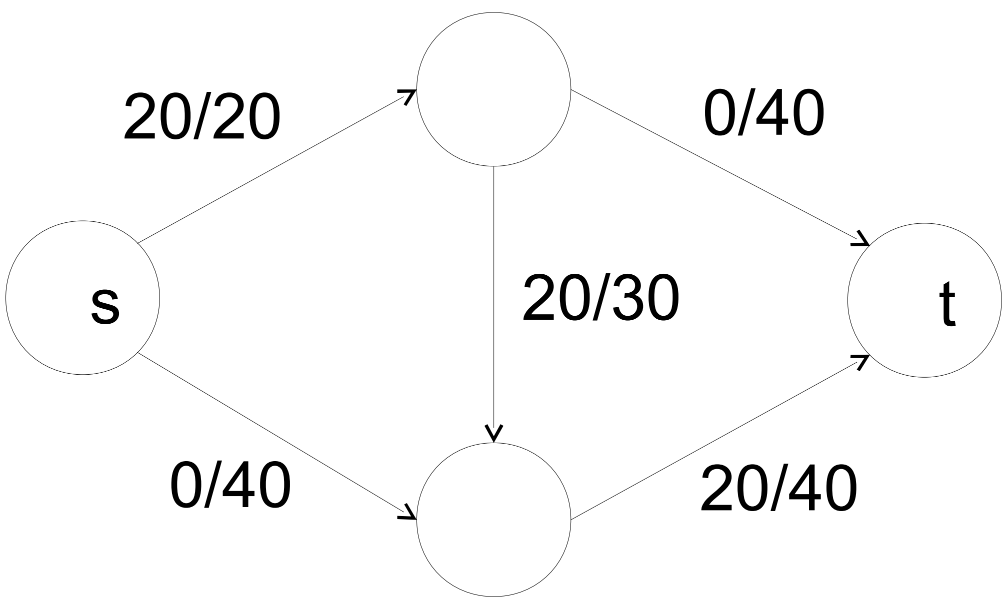


f

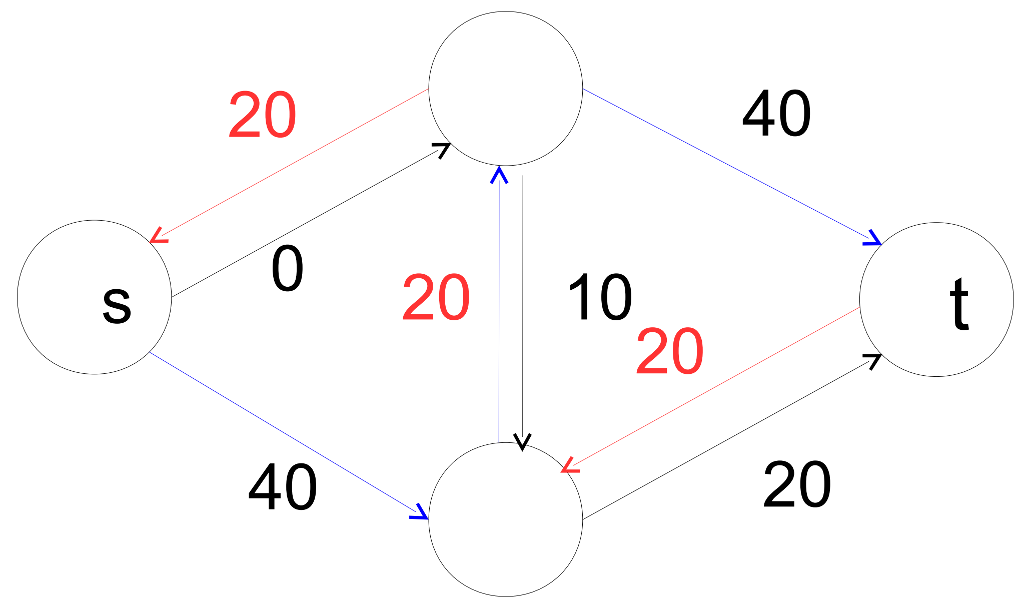


G_f

- Example



f



G_f

An augmenting path

Ford—Fulkerson Algorithm

Given G, s, t, $c(\cdot,\cdot)$. Start with $f \equiv 0$

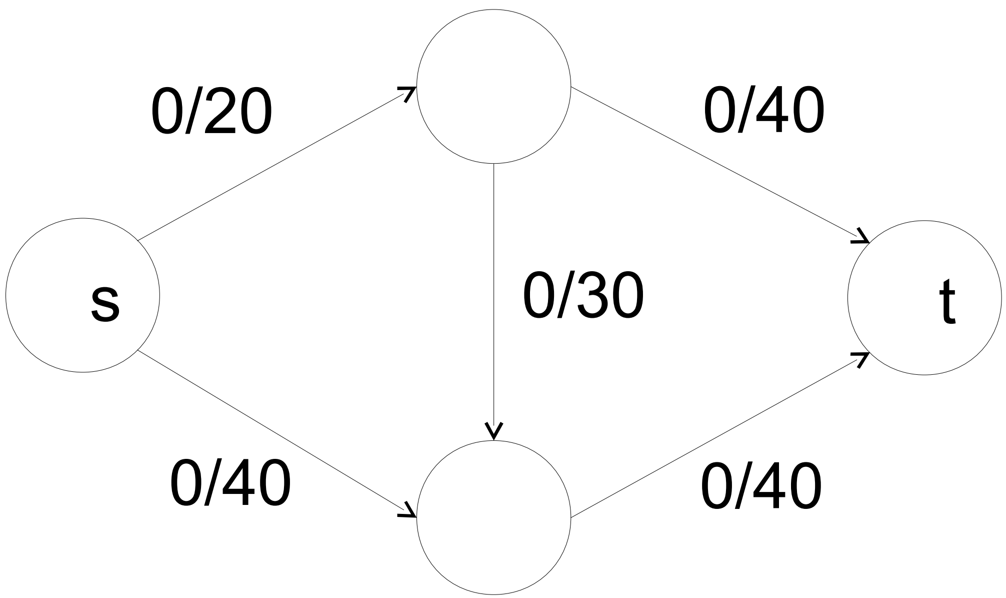
Repeat while there is an augmenting path P in G_f

Let $m = \min_{(u,v)\in P} c_f(u,v)$.

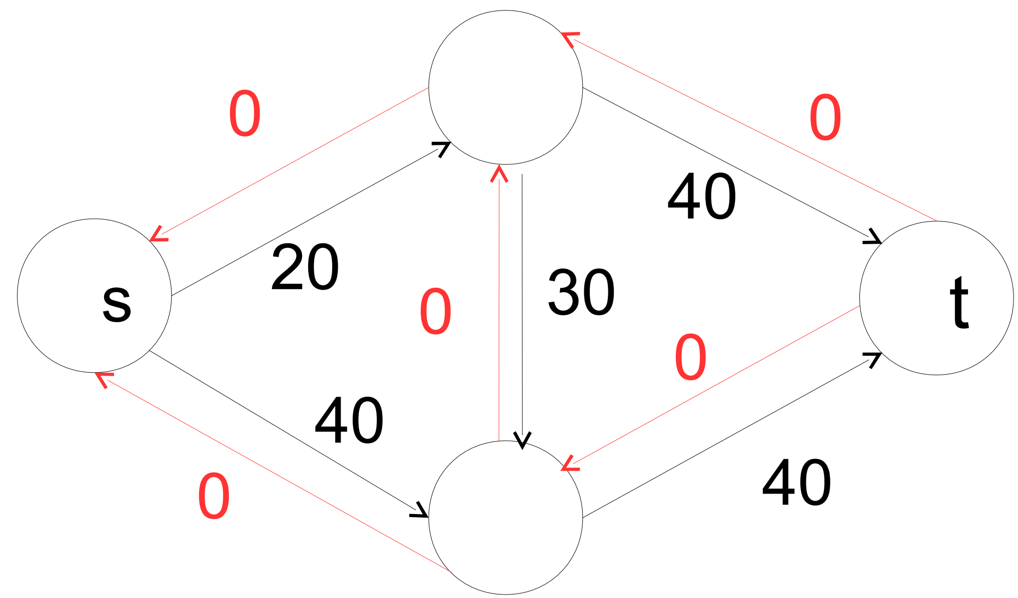
Define $f'(u,v) = m$ if (u,v) in P, $f'(u,v) = 0$ otherwise.

Augment the flow by setting $f = f + f'$

- Demo

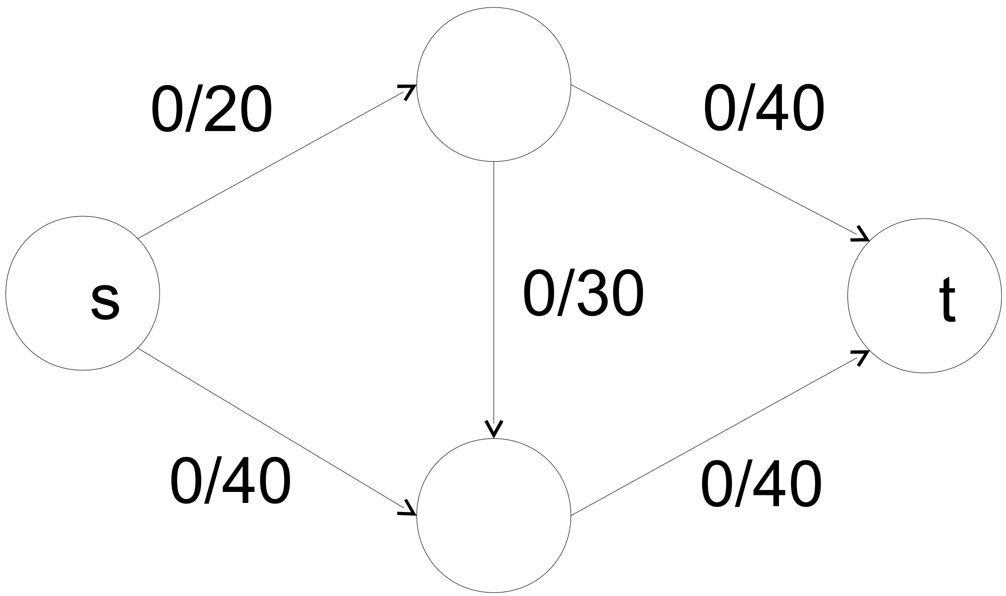


| f | = 0

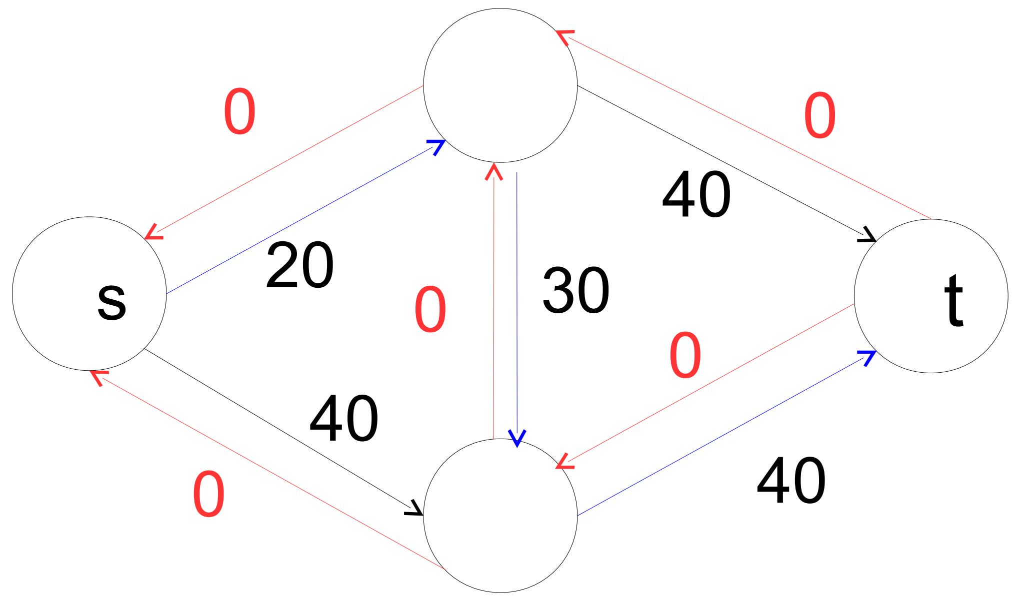


G_f

- Demo



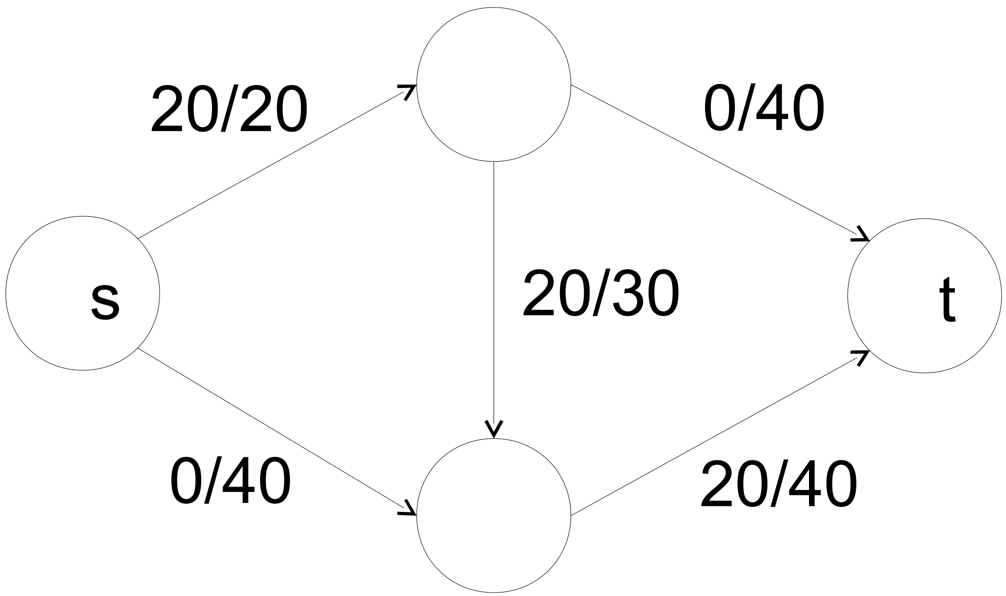
| f | = 0



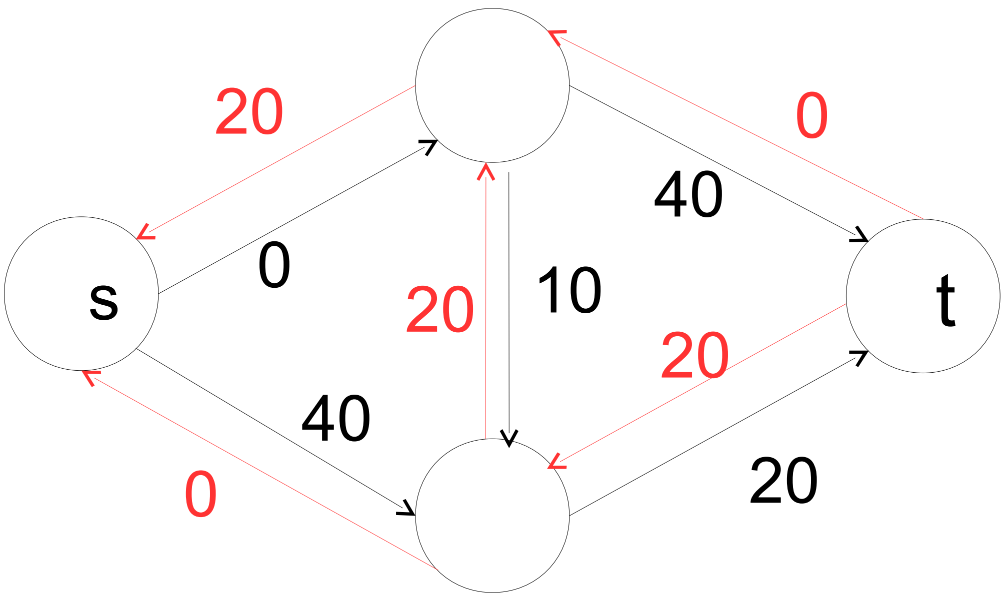
G_f

$\min_{(u,v)\in P} c_f(u,v) = 20$

- Demo

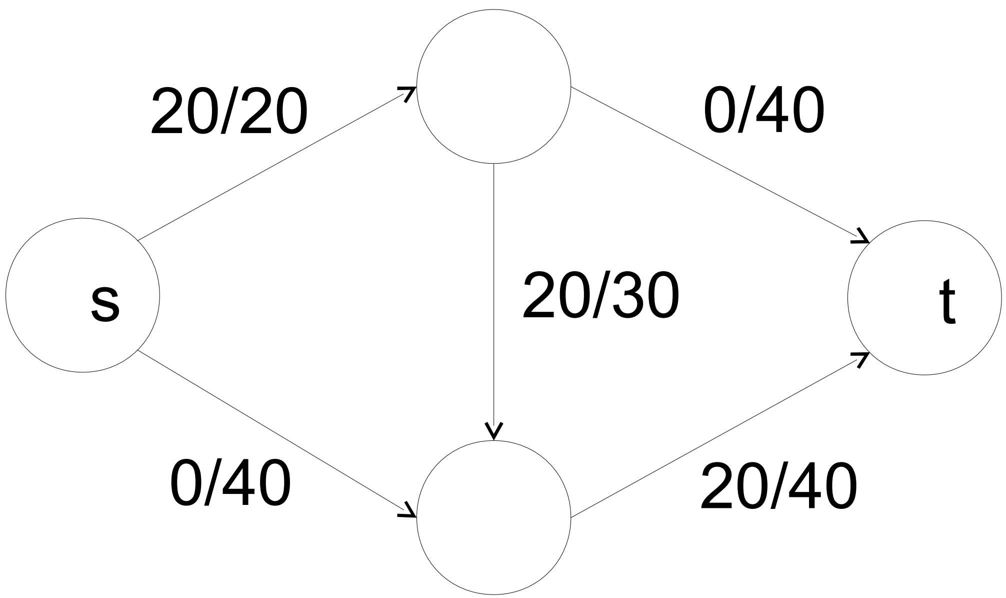


$|f| = 20$

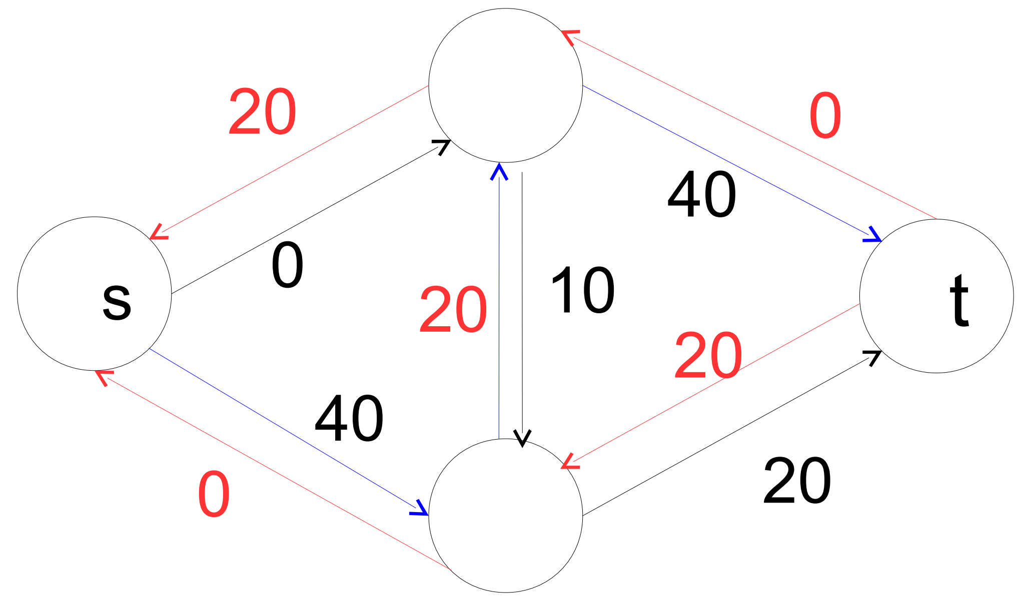


G_f

- Demo



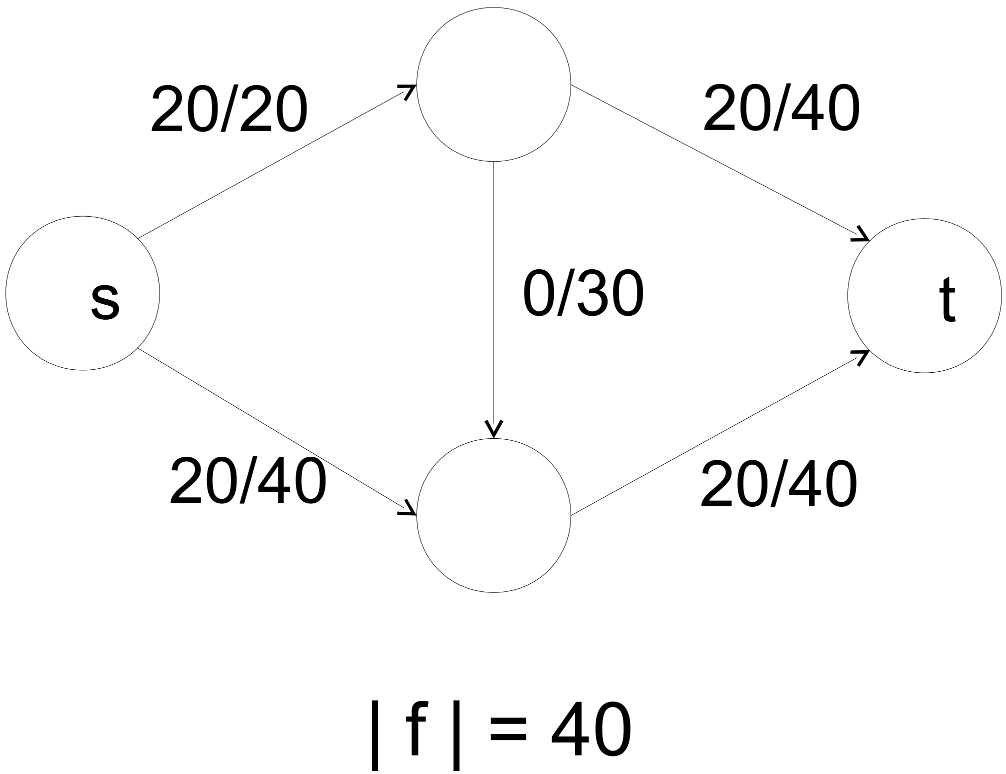
$|f| = 20$

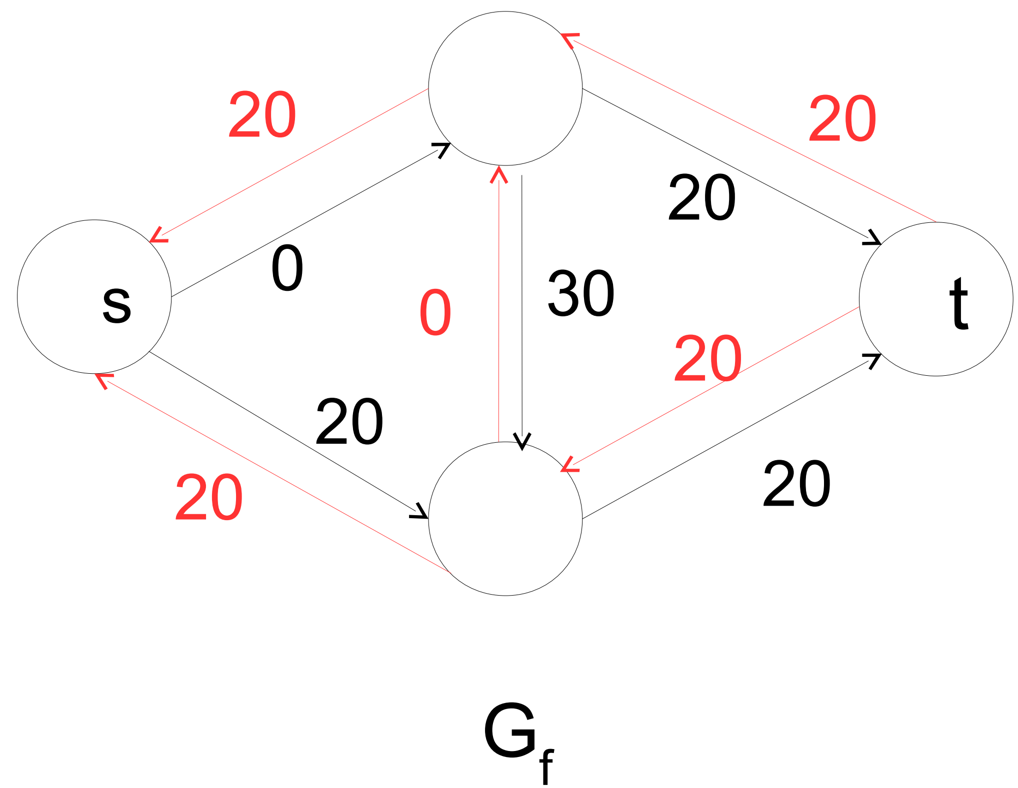


G_f

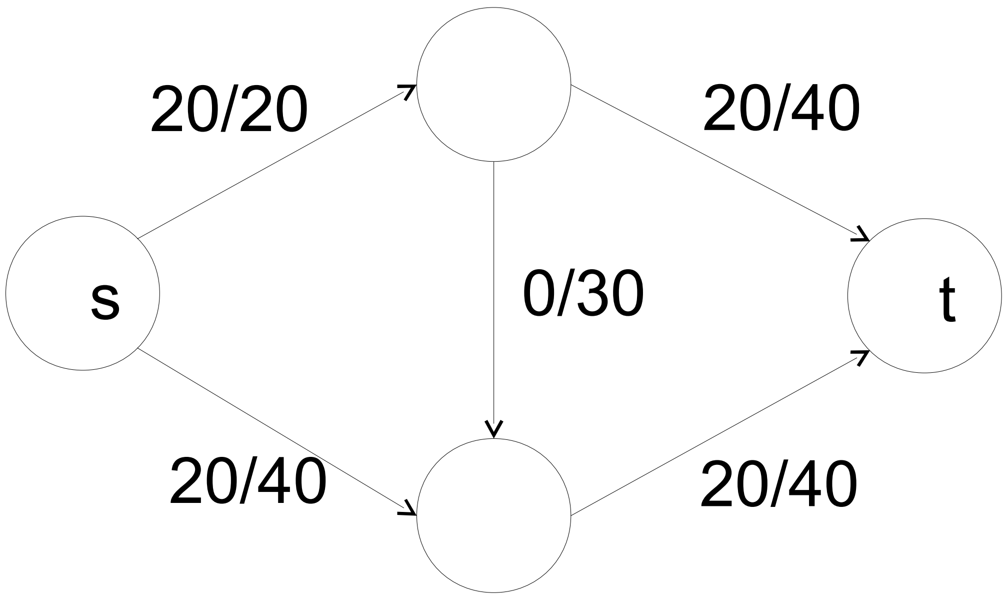
$\min_{(u,v)\in P} c_f(u,v) = 20$

- Demo

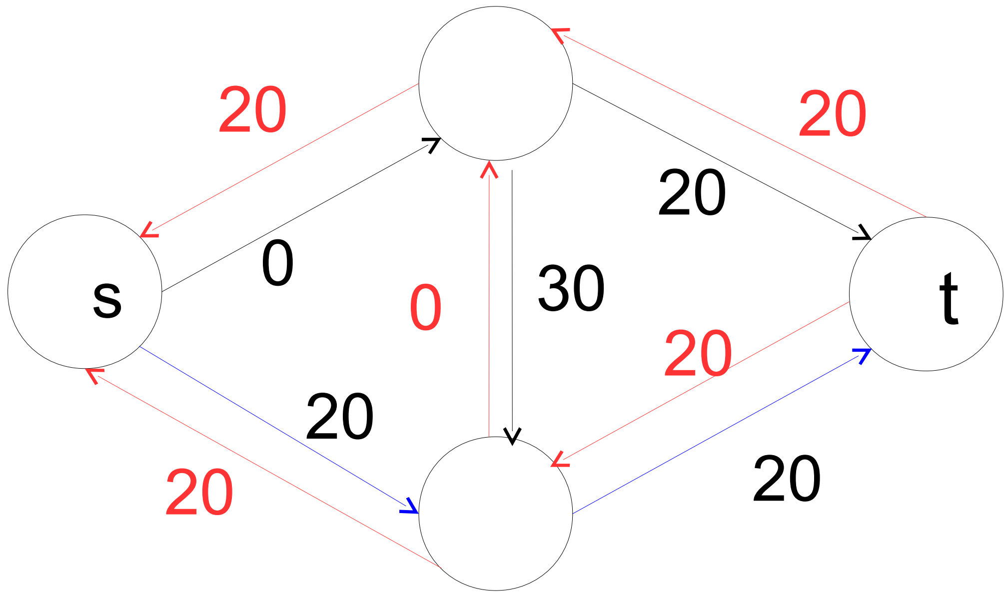




- Demo



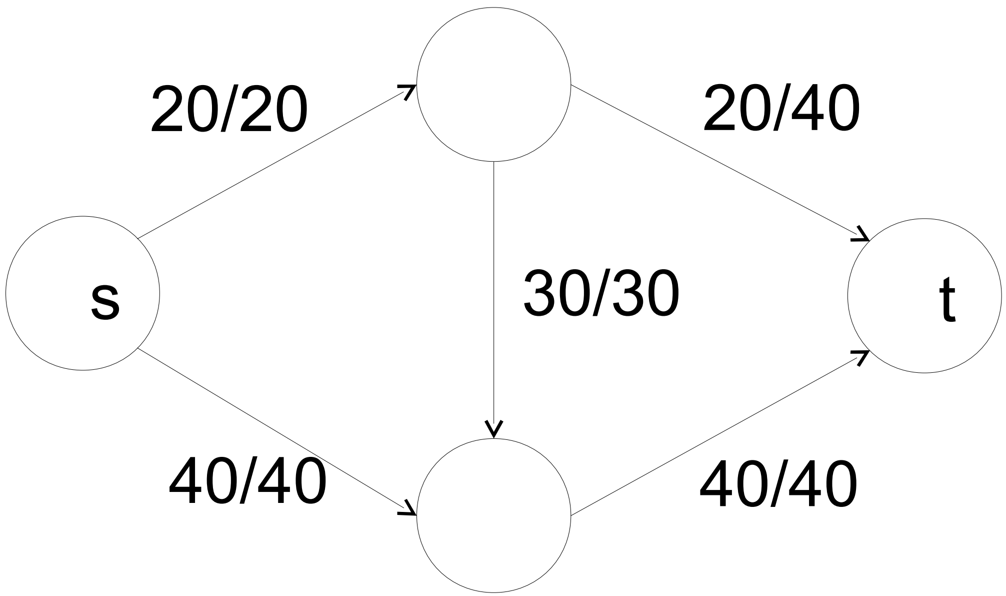
$|f| = 40$



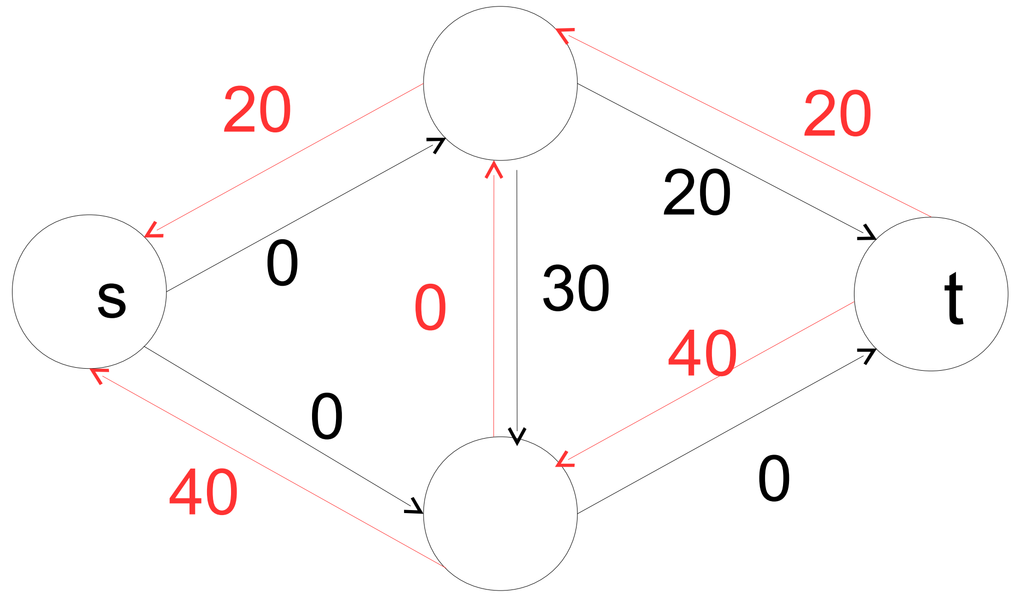
G_f

$\min_{(u,v)\in P} c_f(u,v) = 20$

- Demo

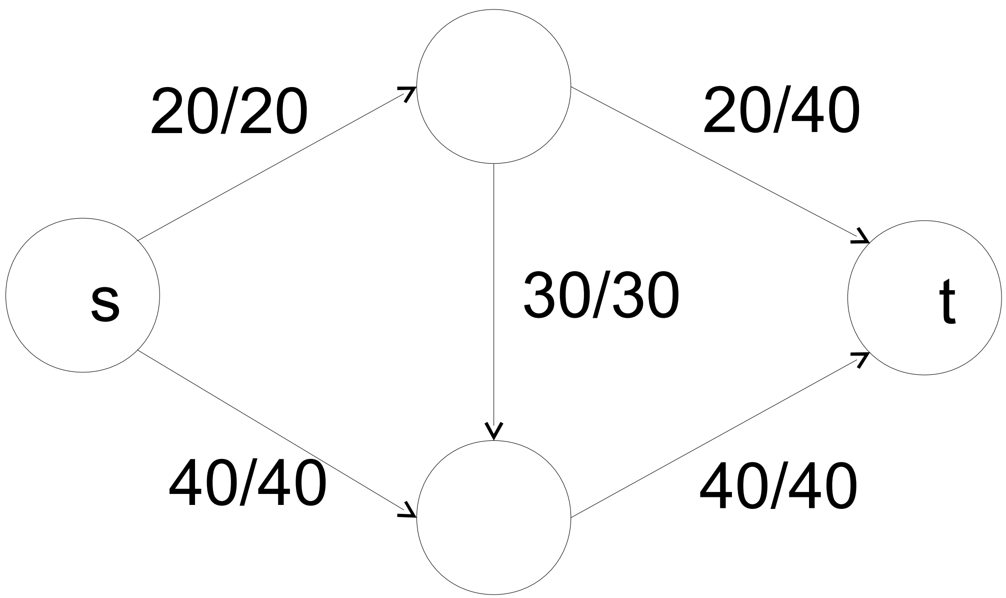


| f | = 60

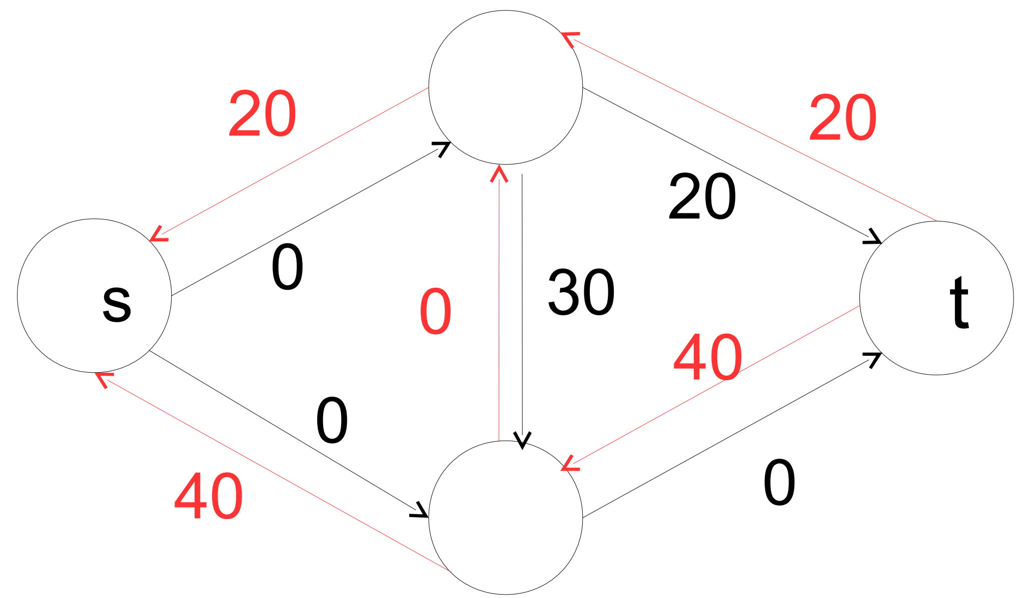


G_f

- Demo



| f | = 60



G_f

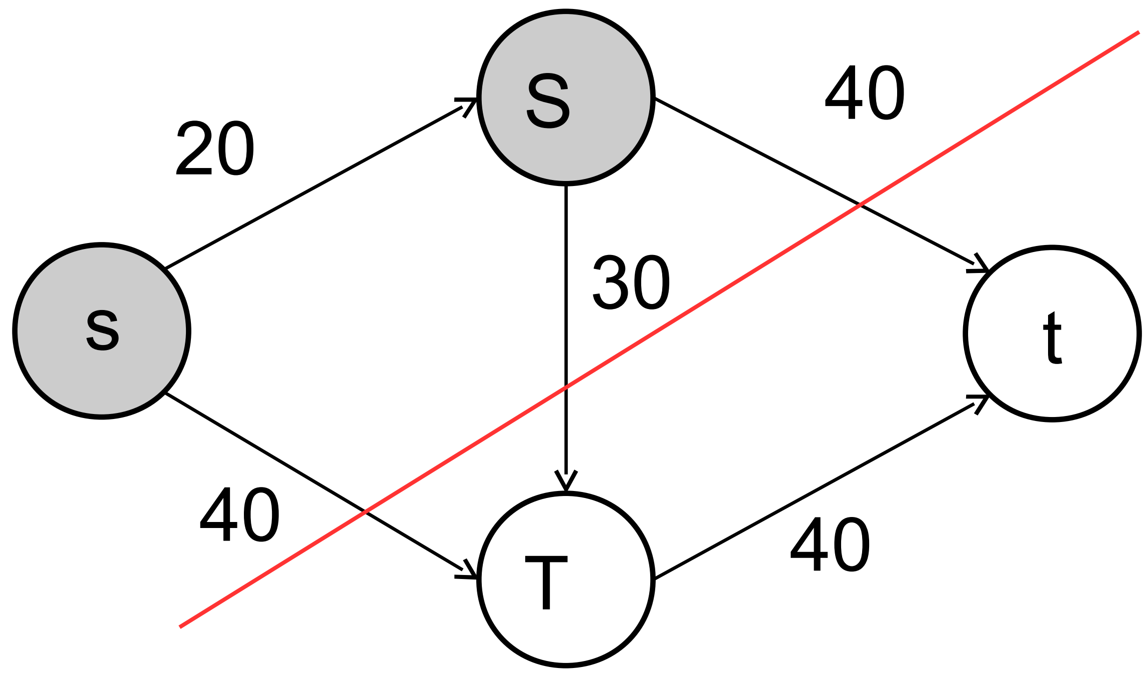
No augmenting path

max | f | = 60

Definition: An s-t cut (S,T) is a partition S, T = V – S such that s in S and t in T.

Meaning: removing the edges between S and T disconnects s and t

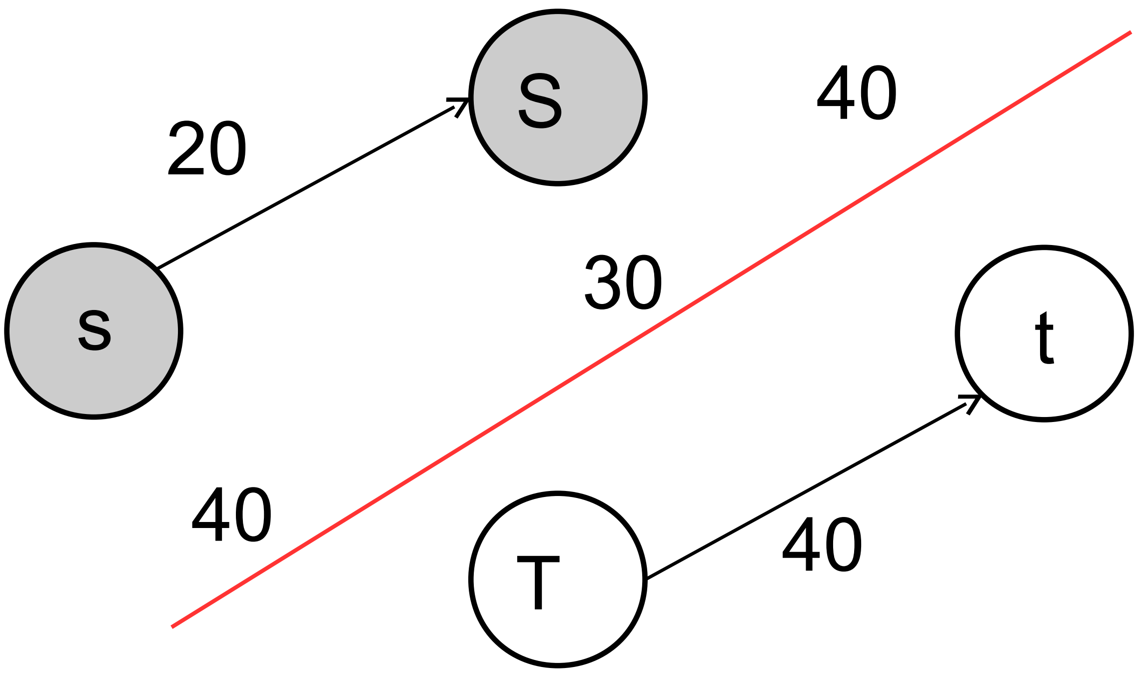
Example:



Definition: An s-t cut (S,T) is a partition S, T = V – S such that s in S and t in T.

Meaning: removing the edges between S and T disconnects s and t

Example:

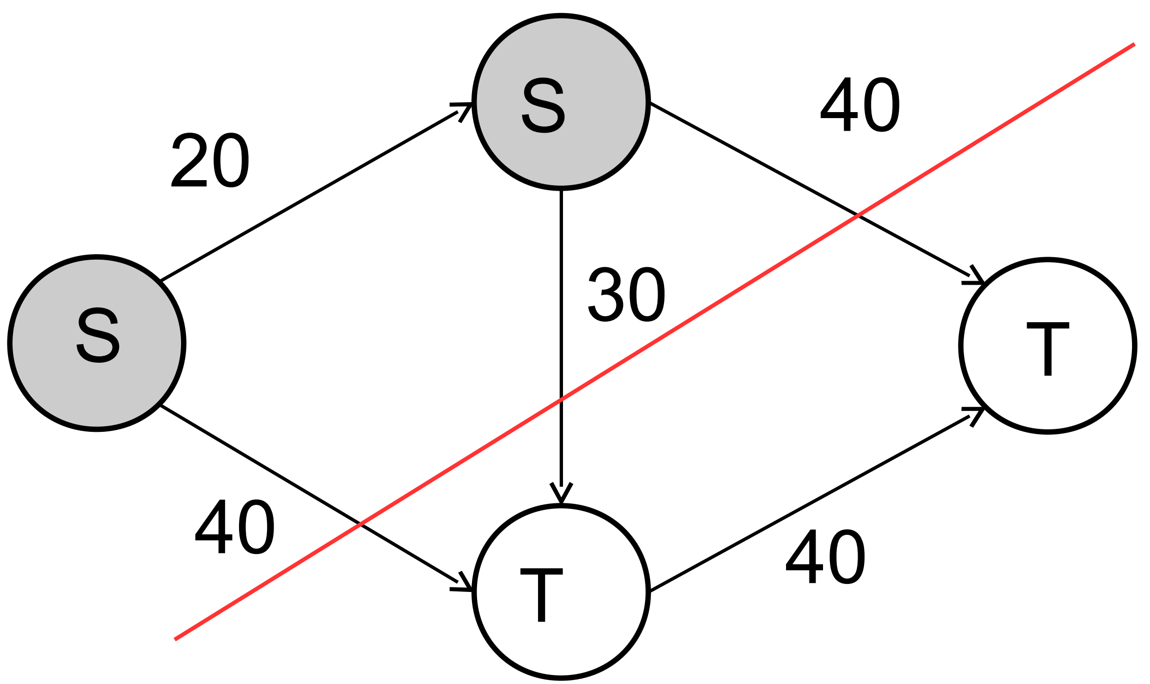


Definition: An s-t cut (S,T) is a partition S, T = V – S such that s in S and t in T.

Meaning: removing the edges between S and T disconnects s and t

The capacity of an s-t cut (S,T) is $c(S,T) := \sum_{u \in S,\, v \in T} c(u,v)$

Example:



$c(S,T) := 40 + 30 + 40 = 110$

Analysis of Ford—Fulkerson algorithm:

Lemma: Let f be a flow. For any cut (S,T), f(S,T) = | f |

Proof:

Let's move x from S to T.

We lose f(x,T), and we gain f(S,x).

But f(x,T) = - f(x,S) because f(x,V) = 0. qed

Theorem (Max flow-min cut): The following are equivalent:

1. |f| is maximum
2. the residual network has no augmenting paths
3. | f | = c(S,T) for some cut (S,T)

Proof:

1 → 2: otherwise could increment the flow as said before.

2 → 3: define S := vertices reachable from s on residual network. Note $t \notin S$. By previous lemma, | f | = f(S,T).

Now note for each edge (u,v) in S×T, f(u,v) = c(u,v), otherwise v would be in S.

3 → 1: if f is not maximum, could have a better flow. But by lemma it would augment the flow on this cut, thus violate capacity constraints. □

Analysis of running time

Fact: Let f be a flow in G. Let f' be a flow on residual network G_f. Then f + f' is a flow on G_f with $|f + f'| = |f| + |f'| > |f|$.

Analysis of running time

Fact: Let f be a flow in G. Let f' be a flow on residual network G_f. Then f + f' is a flow on G_f with $|f + f'| = |f| + |f'| > |f|$.

Assume the capacities are all integers.

In each iteration,
finding an augmentation path takes time ???

Analysis of running time

Fact: Let f be a flow in G. Let f' be a flow on residual network G_f. Then f + f' is a flow on G_f with $|f + f'| = |f| + |f'| > |f|$.

Assume the capacities are all integers.

In each iteration,
    finding an augmentation path takes time O(|E|)
    | f | increments by at least 1

Running time ???

Analysis of running time

Fact: Let f be a flow in G. Let f' be a flow on residual network G_f. Then f + f' is a flow on G_f with $|f + f'| = |f| + |f'| > |f|$.

Assume the capacities are all integers.

In each iteration,

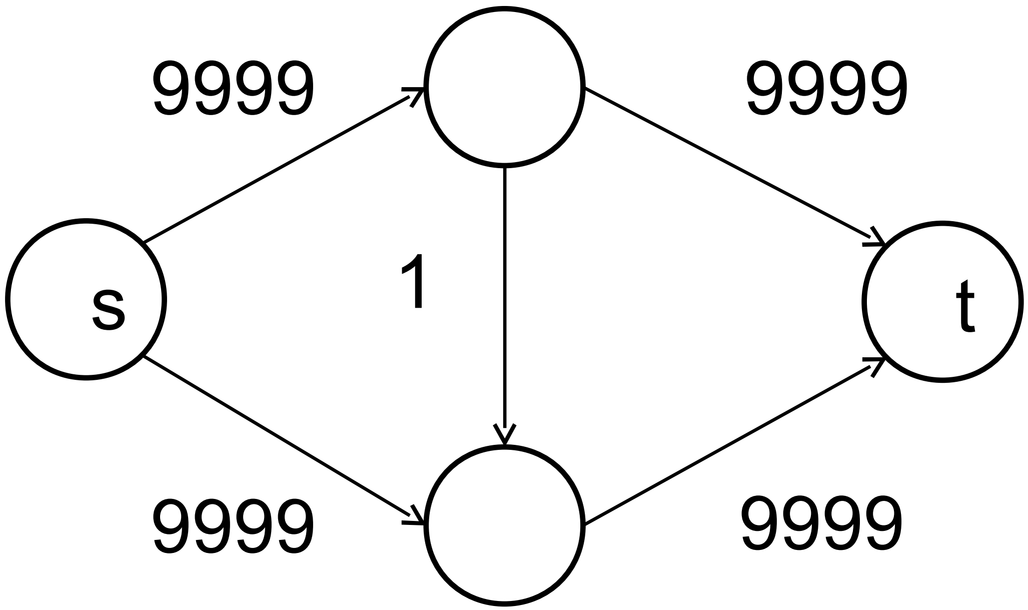
- finding an augmentation path takes time $O(|E|)$
- $| f |$ increments by at least 1

Running time $O(|E| \max |f|)$

Question: Is $O(|E| \max |f|)$ tight?

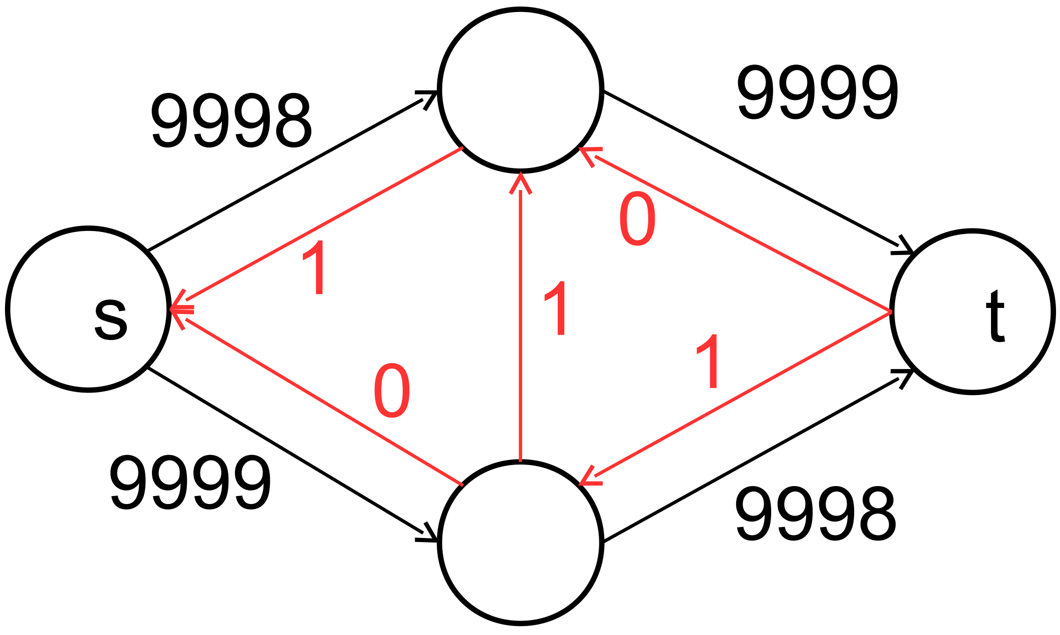
Analysis of running time

Question: Is O(|E| max |f|) tight? Yes.



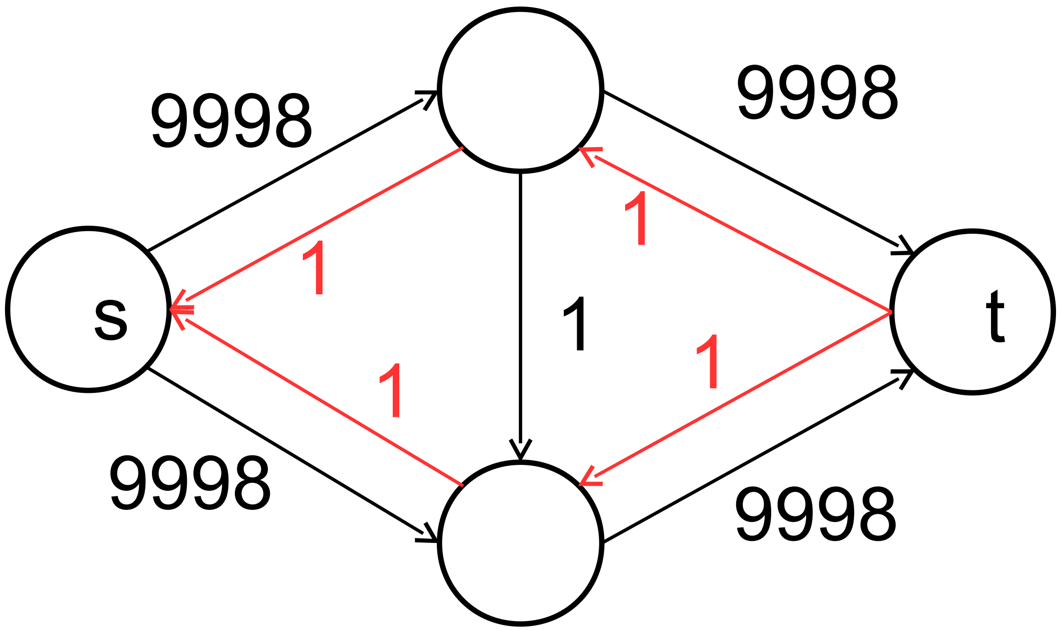
Analysis of running time

Question: Is O(|E| max |f|) tight? Yes.



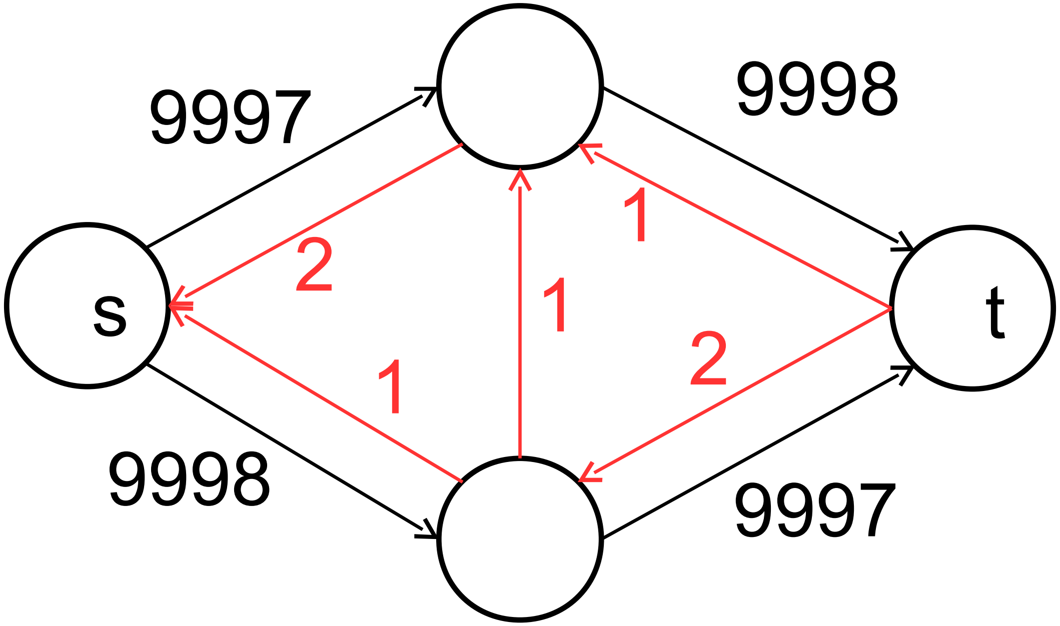
Analysis of running time

Question: Is O(|E| max |f|) tight? Yes.



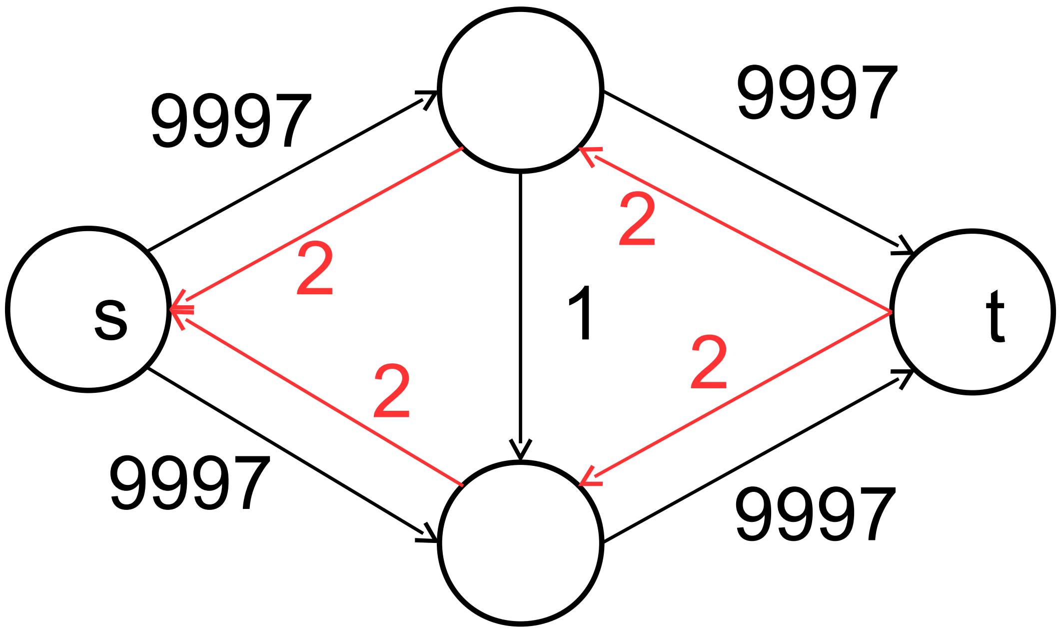
Analysis of running time

Question: Is O(|E| max |f|) tight? Yes.



Analysis of running time

Question: Is O(|E| max |f|) tight? Yes.



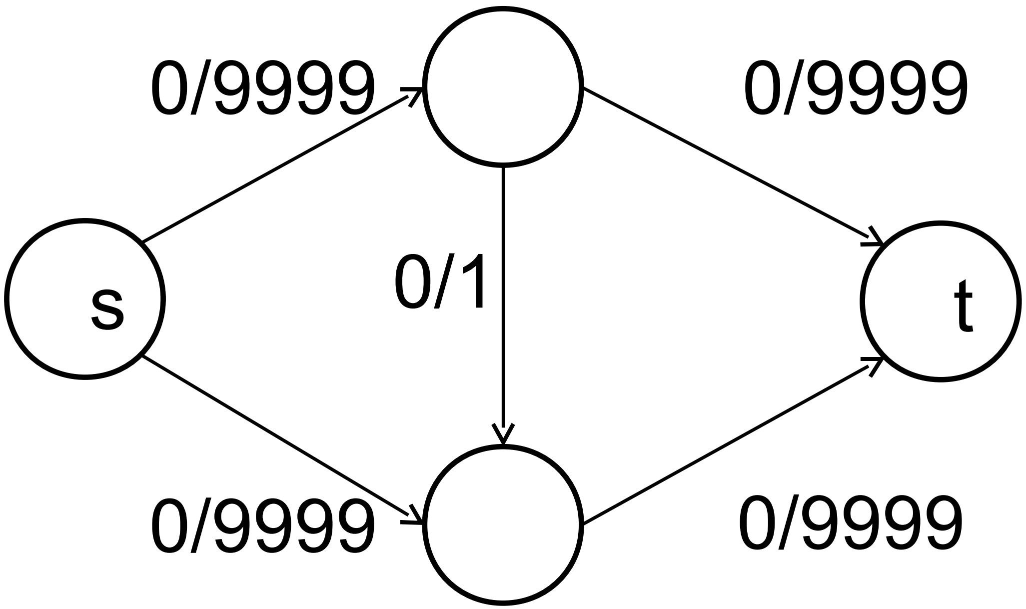
Edmonds—Karp algorithm

- Same as Ford-Fulkerson, but each time use a shortest path in residual network

Let's run it on the previous example.

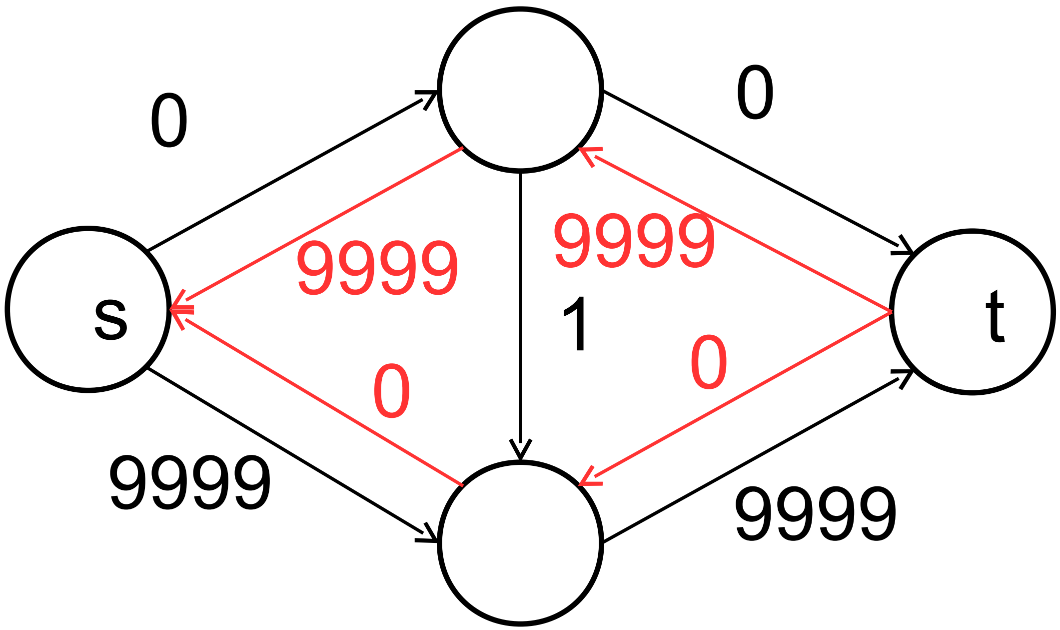
Edmonds—Karp algorithm

Edmonds—Karp on previous example:



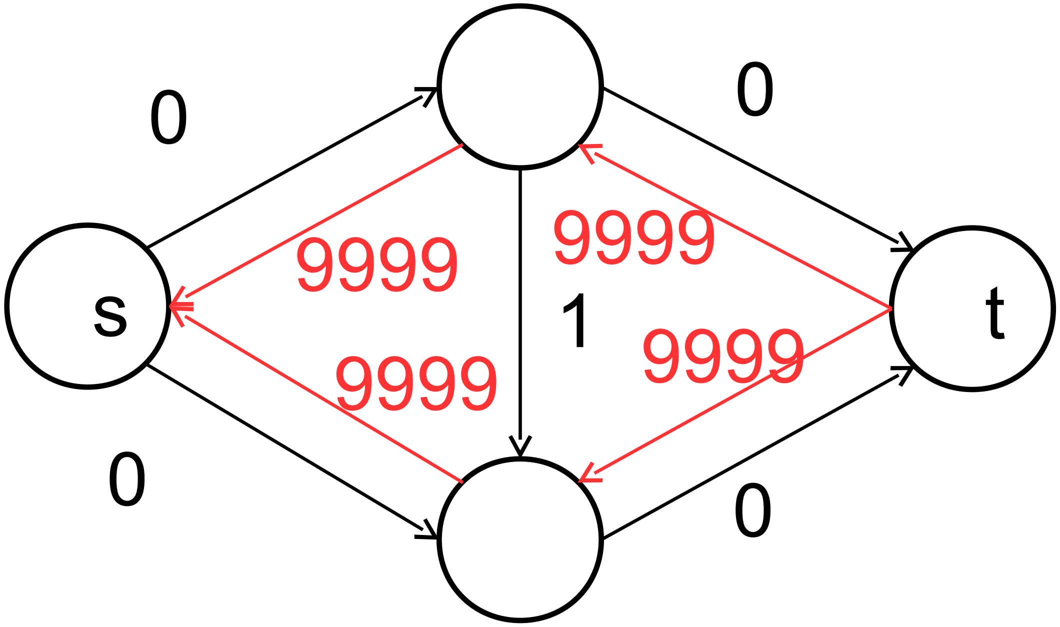
Edmonds—Karp algorithm

Edmonds—Karp on previous example:



Edmonds—Karp algorithm

Edmonds—Karp on previous example:



Analysis of Edmonds—Karp algorithm

Correctness: ???

Analysis of Edmonds—Karp algorithm

Correctness: Follows from previous analysis.

Running time:

Analysis of Edmonds—Karp algorithm

Correctness: Follows from previous analysis.

Running time:

Let $\delta_f(s,v)$ be the distance from s to v in G_f

Lemma: Each time we update the flow, $\delta_f(s,v)$ does not decrease

i.e. $\delta_{f'}(s,v) \geq \delta_f(s,v)$ for every v, for every f' after f

Meaning: shortest path distances increase after each iteration.

Proof: We show that $\delta_{f'}(s,v) \geq \delta_f(s,v)$ if f' is right after f.

Proof: We show that $\delta_{f'}(s,v) \geq \delta_f(s,v)$ if f' is right after f.

Suppose not. Let v be the vertex v among $B:=\{v: \delta_{f'}(s,v) < \delta_f(s,v)\}$ such that $\delta_{f'}(s,v)$ is minimal.

Take shortest path s ~~~> u ~> v in $G_{f'}$.

Proof: We show that $\delta_{f'}(s,v) \geq \delta_f(s,v)$ if f' is right after f.

Suppose not. Let v be the vertex v among $B := \{v: \delta_{f'}(s,v) < \delta_f(s,v)\}$ such that $\delta_{f'}(s,v)$ is minimal.

Take shortest path s ~~~> u ~> v in $G_{f'}$.

We claim that $(u,v) \notin G_f$.

Proof: We show that $\delta_{f'}(s,v) \geq \delta_f(s,v)$ if f' is right after f.

Suppose not. Let v be the vertex v among $B := \{v : \delta_{f'}(s,v) < \delta_f(s,v)\}$ such that $\delta_{f'}(s,v)$ is minimal.

Take shortest path s ~~~> u ~> v in $G_{f'}$.

We claim that $(u,v) \notin G_f$. Otherwise,

$$\delta_f(s,v) \quad \leq \delta_f(s,u) + 1$$

Proof: We show that $\delta_{f'}(s,v) \geq \delta_f(s,v)$ if f' is right after f.

Suppose not. Let v be the vertex v among $B := \{v : \delta_{f'}(s,v) < \delta_f(s,v)\}$ such that $\delta_{f'}(s,v)$ is minimal.

Take shortest path s ~~~> u ~> v in $G_{f'}$.

We claim that $(u,v) \notin G_f$. Otherwise,

$$\begin{aligned} \delta_f(s,v) &\leq \delta_f(s,u) + 1 \text{ (Triangle inequality)} \\ &\leq \delta_{f'}(s,u) + 1 \end{aligned}$$

Proof: We show that $\delta_{f'}(s,v) \geq \delta_f(s,v)$ if f' is right after f.

Suppose not. Let v be the vertex v among $B := \{v : \delta_{f'}(s,v) < \delta_f(s,v)\}$ such that $\delta_{f'}(s,v)$ is minimal.

Take shortest path s ~~~> u ~> v in $G_{f'}$.

We claim that $(u,v) \notin G_f$. Otherwise,

$$
\begin{aligned}
\delta_f(s,v) &\leq \delta_f(s,u) + 1 \text{ (Triangle inequality)} \\
&\leq \delta_{f'}(s,u) + 1 \text{ } (u \notin B \text{ because } \delta_{f'}(s,u) < \delta_{f'}(s,v)) \\
&= \delta_{f'}(s,v)
\end{aligned}
$$

Proof: We show that $\delta_{f'}(s,v) \geq \delta_f(s,v)$ if f' is right after f.

Suppose not. Let v be the vertex v among $B:=\{v: \delta_{f'}(s,v) < \delta_f(s,v)\}$ such that $\delta_{f'}(s,v)$ is minimal.

Take shortest path s ~~~> u ~> v in $G_{f'}$.

We claim that $(u,v) \notin G_f$. Otherwise,

$$\begin{aligned} \delta_f(s,v) &\leq \delta_f(s,u) + 1 \text{ (Triangle inequality)} \\ &\leq \delta_{f'}(s,u) + 1 \text{ (}u \notin B \text{ because } \delta_{f'}(s,u) < \delta_{f'}(s,v)\text{)} \\ &= \delta_{f'}(s,v) \end{aligned}$$

Contradicting our assumption. So we have (u,v) in $G_{f'}$ but $(u,v) \notin G_f$

That means ???

Proof: We show that $\delta_{f'}(s,v) \geq \delta_f(s,v)$ if f' is right after f.

Suppose not. Let v be the vertex v among $B:=\{v: \delta_{f'}(s,v) < \delta_f(s,v)\}$ such that $\delta_{f'}(s,v)$ is minimal.

Take shortest path s ~~~> u ~> v in $G_{f'}$.

We claim that $(u,v) \notin G_f$. Otherwise,

$$\begin{aligned} \delta_f(s,v) &\leq \delta_f(s,u) + 1 \text{ (Triangle inequality)} \\ &\leq \delta_{f'}(s,u) + 1 \text{ } (u \notin B \text{ because } \delta_{f'}(s,u) < \delta_{f'}(s,v)) \\ &= \delta_{f'}(s,v) \end{aligned}$$

Contradicting our assumption. So we have (u,v) in $G_{f'}$ but $(u,v) \notin G_f$

That means the augmentation from f to f' must have (v, u) on the augmented path.

Proof: We show that $\delta_{f'}(s,v) \geq \delta_f(s,v)$ if f' is right after f.

Suppose not. Let v be the vertex v among $B:=\{v: \delta_{f'}(s,v) < \delta_f(s,v)\}$ such that $\delta_{f'}(s,v)$ is minimal.

Take shortest path s ~~~> u ~> v in $G_{f'}$. We have (u,v) in $G_{f'}$ but $(u,v) \notin G_f$

That means the augmentation from f to f' must have (v, u) on the augmented path.

But augmentations are along shortest paths, so

$$\delta_f(s,v) \quad = \delta_f(s,u) - 1$$

Proof: We show that $\delta_{f'}(s,v) \geq \delta_f(s,v)$ if f' is right after f.

Suppose not. Let v be the vertex v among $B := \{v: \delta_{f'}(s,v) < \delta_f(s,v)\}$ such that $\delta_{f'}(s,v)$ is minimal.

Take shortest path s ~~~> u ~> v in $G_{f'}$. We have (u,v) in $G_{f'}$ but $(u,v) \notin G_f$

That means the augmentation from f to f' must have (v, u) on the augmented path.

But augmentations are along shortest paths, so

$$\begin{aligned} \delta_f(s,v) &= \delta_f(s,u) - 1 \\ &\leq \delta_{f'}(s,u) - 1 \end{aligned}$$

Proof: We show that $\delta_{f'}(s,v) \geq \delta_f(s,v)$ if f' is right after f.

Suppose not. Let v be the vertex v among $B:=\{v: \delta_{f'}(s,v) < \delta_f(s,v)\}$ such that $\delta_{f'}(s,v)$ is minimal.

Take shortest path s ~~~> u ~> v in $G_{f'}$. We have (u,v) in $G_{f'}$ but $(u,v) \notin G_f$

That means the augmentation from f to f' must have (v, u) on the augmented path.

But augmentations are along shortest paths, so

$$\begin{aligned}
\delta_f(s,v) &= \delta_f(s,u) - 1 \\
&\leq \delta_{f'}(s,u) - 1 \text{ (because } u \notin B) \\
&= \delta_{f'}(s,v) - 1 - 1
\end{aligned}$$

Proof: We show that $\delta_{f'}(s,v) \geq \delta_f(s,v)$ if f' is right after f.

Suppose not. Let v be the vertex v among $B := \{v: \delta_{f'}(s,v) < \delta_f(s,v)\}$ such that $\delta_{f'}(s,v)$ is minimal.

Take shortest path s ~~~> u ~> v in $G_{f'}$. We have (u,v) in $G_{f'}$ but $(u,v) \notin G_f$

That means the augmentation from f to f' must have (v, u) on the augmented path.

But augmentations are along shortest paths, so

$$\begin{aligned}\delta_f(s,v) &= \delta_f(s,u) - 1 \\ &\leq \delta_{f'}(s,u) - 1 \text{ (because } u \notin B) \\ &= \delta_{f'}(s,v) - 1 - 1 \text{ (because } \delta_{f'}(s,u) < \delta_{f'}(s,v))\end{aligned}$$

which contradicts our assumption. □

Theorem: Total # of flow augmentations in Edmond—Karp is O(|V| |E|)

Proof:

Theorem: Total # of flow augmentations in Edmond—Karp is O(|V| |E|)

Proof: Call (u,v) critical in residual network G_f if $c_f(u,v)$ is minimial among all edges on an augmenting path.

(i.e. (u,v) is the bottleneck edge of the augmenting path).

Theorem: Total # of flow augmentations in Edmond—Karp is $O(|V| |E|)$

Proof: Call (u,v) critical in residual network G_f if $c_f(u,v)$ is minimial among all edges on an augmenting path.

Note there is always a critical edge. After the flow augmentation, the edge will be ???

Theorem: Total # of flow augmentations in Edmond—Karp is $O(|V|\,|E|)$

Proof: Call (u,v) critical in residual network G_f if $c_f(u,v)$ is minimial among all edges on an augmenting path.

Note there is always a critical edge. After the flow augmentation, the edge will be saturated and thus will disappear from the network.

Theorem: Total # of flow augmentations in Edmond—Karp is O(|V| |E|)

Proof: Call (u,v) critical in residual network G_f if $c_f(u,v)$ is minimial among all edges on an augmenting path.

Note there is always a critical edge. After the flow augmentation, the edge will be saturated and thus will disappear from the network.

It can only come back to the critical edge if ???

Theorem: Total # of flow augmentations in Edmond—Karp is O(|V| |E|)

Proof: Call (u,v) critical in residual network G_f if $c_f(u,v)$ is minimial among all edges on an augmenting path.

Note there is always a critical edge. After the flow augmentation, the edge will be saturated and thus will disappear from the network.

It can only come back to the critical edge if edge (v,u) is used on a flow augmentation of a new residual network $G_{f'}$.

Theorem: Total # of flow augmentations in Edmond—Karp is O(|V| |E|)

Proof: Call (u,v) critical in residual network G_f if $c_f(u,v)$ is minimial among all edges on an augmenting path.

Note there is always a critical edge. After the flow augmentation, the edge will be saturated and thus will disappear from the network.

It can only come back to the critical edge if edge (v,u) is used on a flow augmentation of a new residual network $G_{f'}$.

But then $\delta_{f'}(s,u) = \delta_{f'}(s,v) + 1$

Theorem: Total # of flow augmentations in Edmond—Karp is O(|V| |E|)

Proof: Call (u,v) critical in residual network G_f if $c_f(u,v)$ is minimial among all edges on an augmenting path.

Note there is always a critical edge. After the flow augmentation, the edge will be saturated and thus will disappear from the network.

It can only come back to the critical edge if edge (v,u) is used on a flow augmentation of a new residual network $G_{f'}$.

But then $\delta_{f'}(s,u) = \delta_{f'}(s,v) + 1$

(since always use shortest augmentation path)

$\geq \delta_f(s,v) + 1$

Theorem: Total # of flow augmentations in Edmond—Karp is O(|V| |E|)

Proof: Call (u,v) critical in residual network G_f if $c_f(u,v)$ is minimial among all edges on an augmenting path.

Note there is always a critical edge. After the flow augmentation, the edge will be saturated and thus will disappear from the network.

It can only come back to the critical edge if edge (v,u) is used on a flow augmentation of a new residual network $G_{f'}$.

But then $\delta_{f'}(s,u) = \delta_{f'}(s,v) + 1$

(since always use shortest augmentation path)

$\geq \delta_f(s,v) + 1$ (by previous lemma)

$= \delta_f(s,u) + 1 + 1$

Theorem: Total # of flow augmentations in Edmond—Karp is O(|V| |E|)

Proof: Call (u,v) critical in residual network G_f if $c_f(u,v)$ is minimial among all edges on an augmenting path.

Note there is always a critical edge. After the flow augmentation, the edge will be saturated and thus will disappear from the network.

It can only come back to the critical edge if edge (v,u) is used on a flow augmentation of a new residual network $G_{f'}$.

But then $\delta_{f'}(s,u) = \delta_{f'}(s,v) + 1$

(since always use shortest augmentation path)

$\geq \delta_f(s,v) + 1$ (by previous lemma)

$= \delta_f(s,u) + 1 + 1$

(again because you augment along shortest path).

- Note that between two times that (u,v) becomes critical, the distance of u increases by ???

- Note that between two times that (u,v) becomes critical, the distance of u increases by 2.

Thus (u,v) becomes critical O(|V|) times.

- Note that between two times that (u,v) becomes critical, the distance of u increases by 2.

Thus (u,v) becomes critical O(|V|) times.

Thus, the total # of augmentation is ???

- Note that between two times that (u,v) becomes critical, the distance of u increases by 2.

Thus (u,v) becomes critical O(|V|) times.

Thus, the total # of augmentation is O(|V| |E|). □

● Note that between two times that (u,v) becomes critical, the distance of u increases by 2.

Thus (u,v) becomes critical O(|V|) times.

Thus, the total # of augmentation is O(|V| |E|). □

Remark

This is NOT saying every augmentation increases the distance of some node

This is saying every 2 augmentations of same edge increase distance of starting point.

Application: Maximum matching

Definition: Given a bipartite graph $(L \cup R, E)$. $M \subset E$ is a matching if no edges in M share a vertex.

Application: Maximum matching

Definition: Given a bipartite graph $(L \cup R, E)$. $M \subset E$ is a matching if no edges in M share a vertex.

Input: a bipartite graph $(L \cup R, E)$

Output: max $|M|$

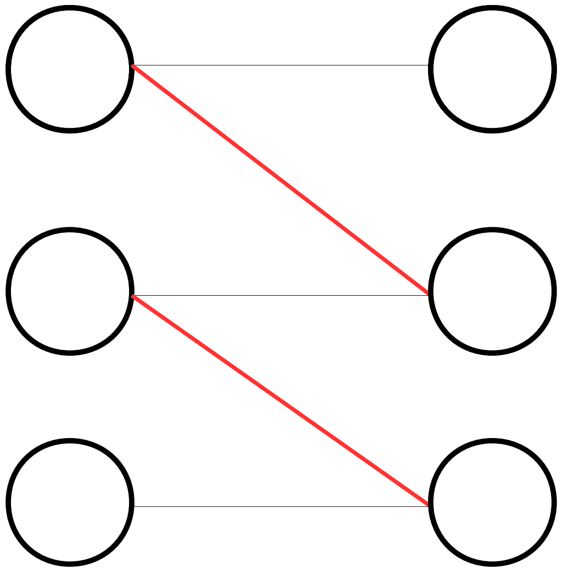
Application: Maximum matching

Definition: Given a bipartite graph (L∪R,E). M⊂E is a matching if no edges in M share a vertex.

Input: a bipartite graph (L∪R,E)

Output: max |M|

Example:



|M| = ???

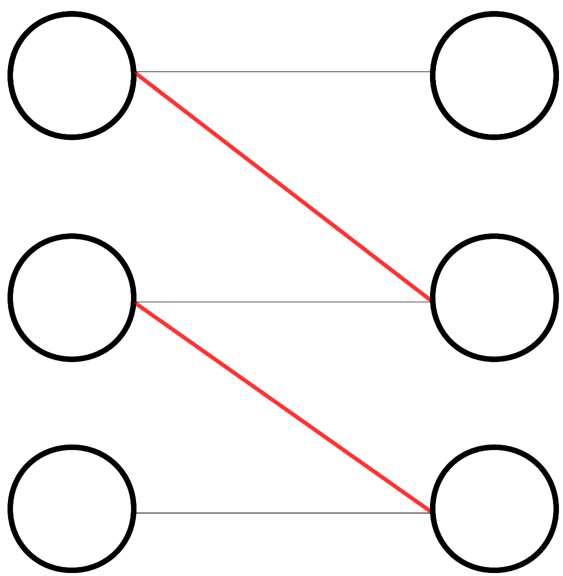
Application: Maximum matching

Definition: Given a bipartite graph (L∪R,E). M⊂E is a matching if no edges in M share a vertex.

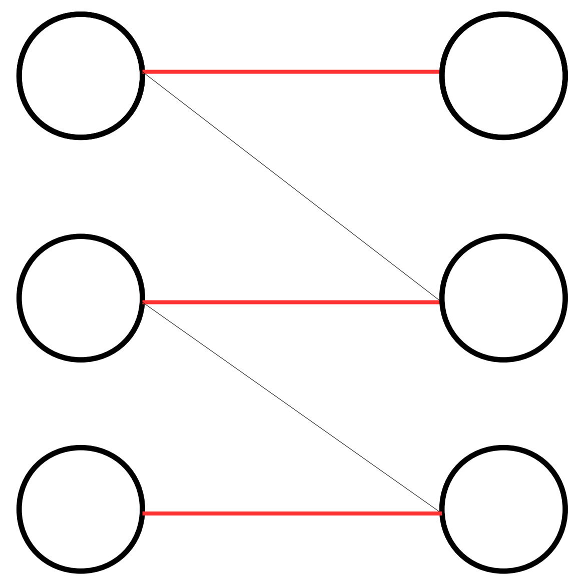
Input: a bipartite graph (L∪R,E)

Output: max |M|

Example:



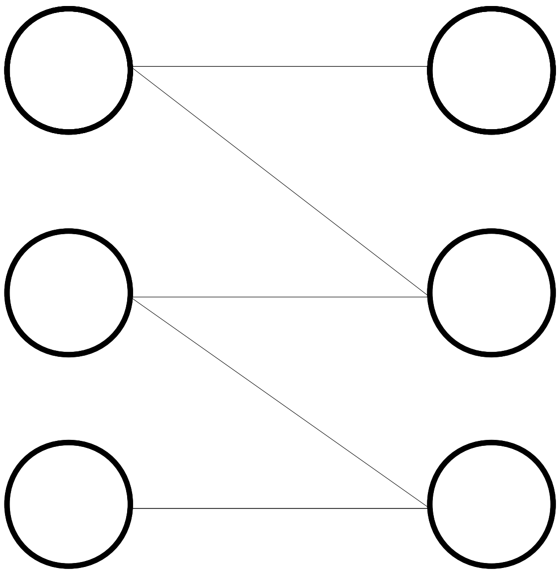
|M| = 2



|M| = 3

Application: Maximum matching

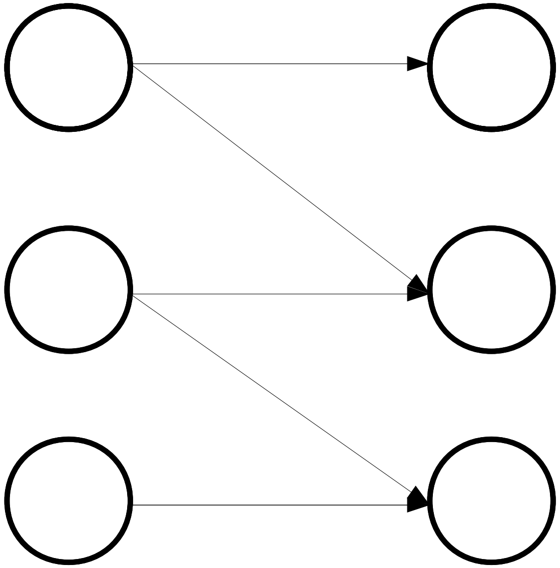
We turn it into a maximum flow problem



Application: Maximum matching

We turn it into a maximum flow problem

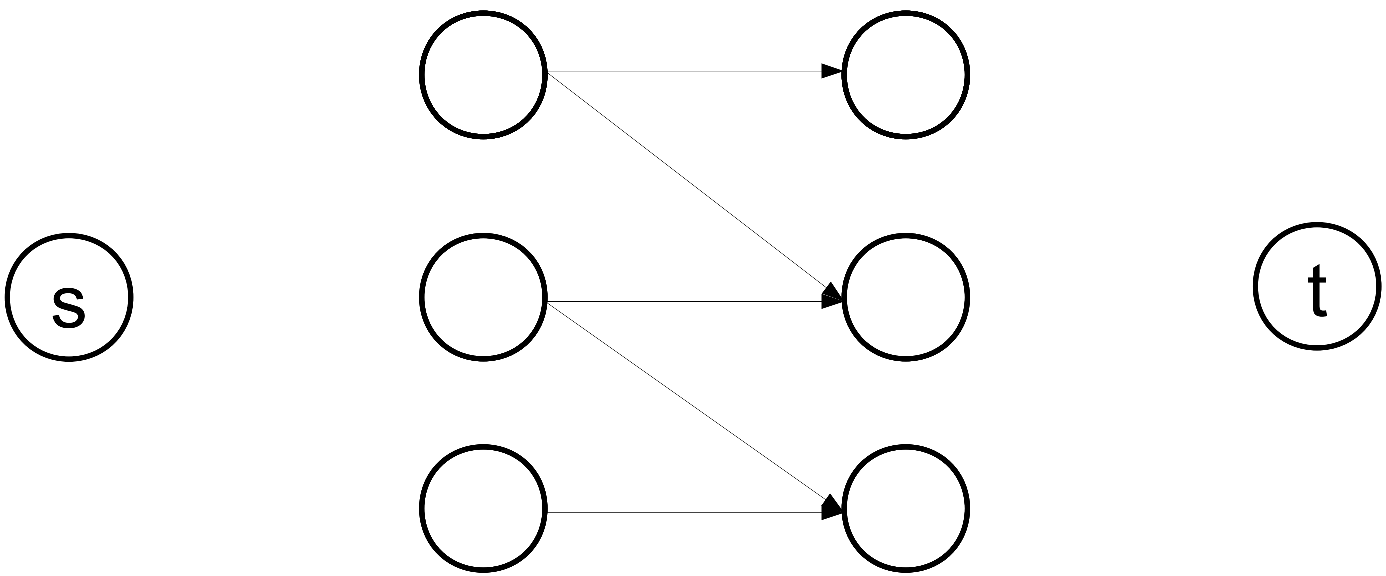
- Direct every edge from L to R



Application: Maximum matching

We turn it into a maximum flow problem

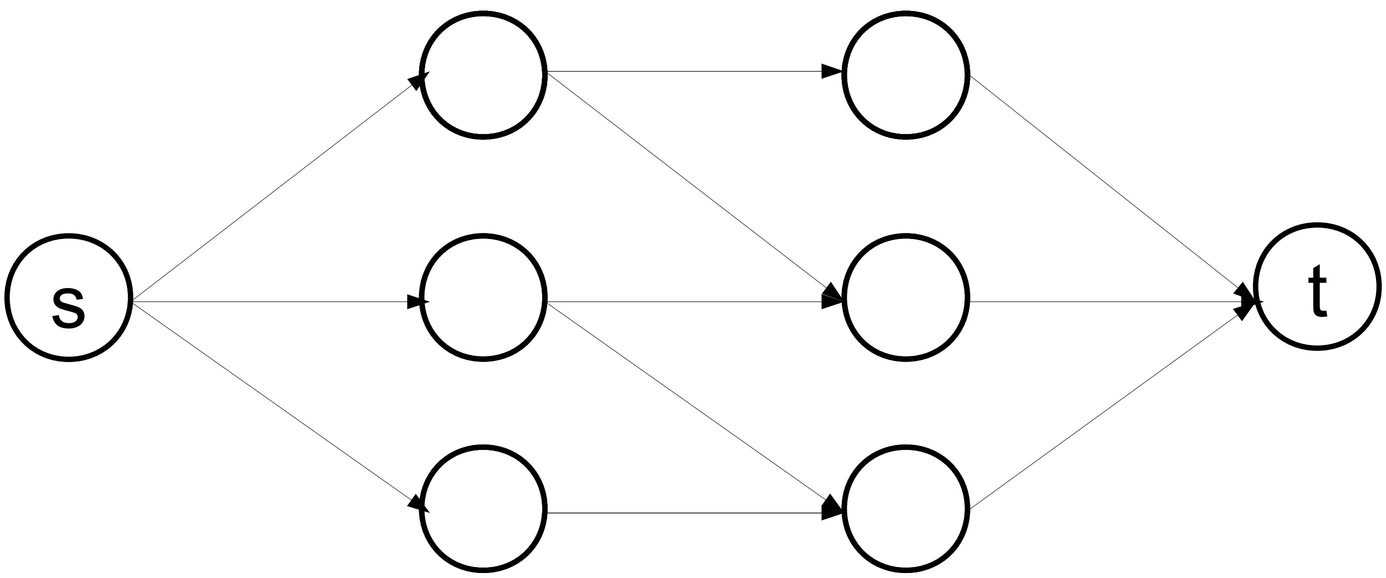
- Direct every edge from L to R
- Add a source and a sink



Application: Maximum matching

We turn it into a maximum flow problem

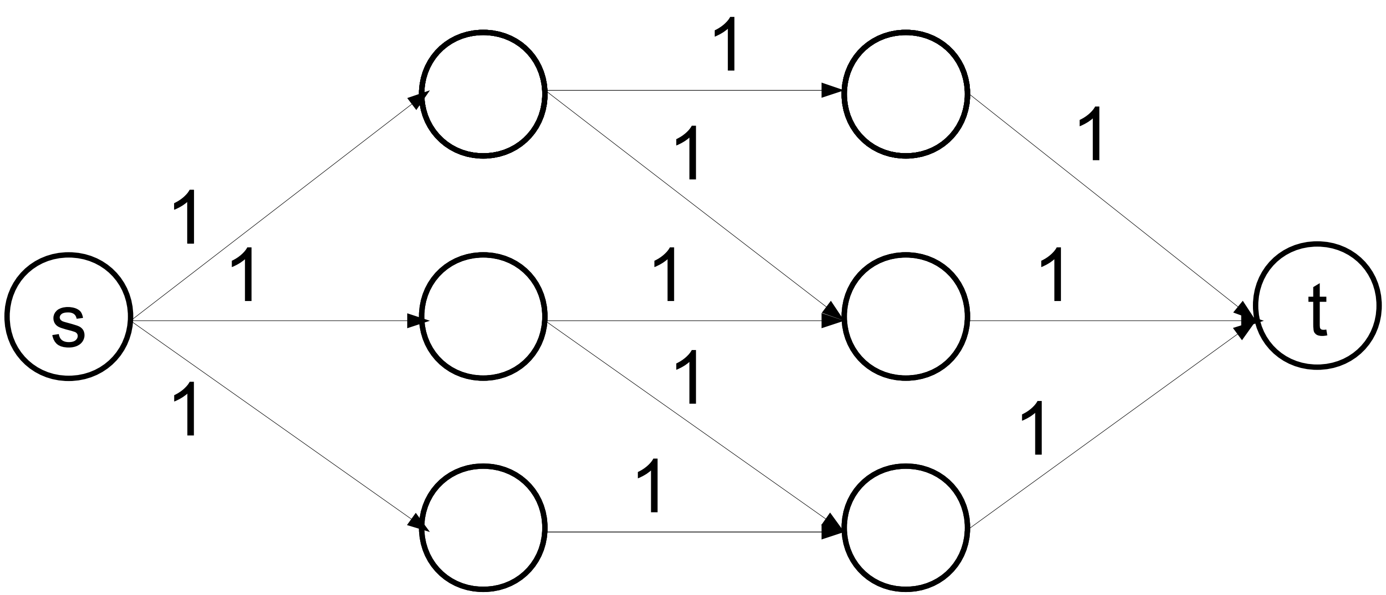
- Direct every edge from L to R
- Add a source and a sink
- Add edges between s and vertices in L, and between t and the vertices in R



Application: Maximum matching

We turn it into a maximum flow problem

- Direct every edge from L to R
- Add a source and a sink
- Add edges between s and vertices in L, and between t and the vertices in R
- Set capacities of all edges to 1



Application: Maximum matching

We turn it into a maximum flow problem

Claim: $\exists$Matching of size M $\Leftrightarrow$ $\exists$flow of value M

Proof:

$\Rightarrow$:

Application: Maximum matching

We turn it into a maximum flow problem

Claim: ∃Matching of size M ⇔ ∃flow of value M

Proof:

⇒: Clear

⇐: Let f be a flow. There are |f| units.

The |f| units form |f| edge-disjoint paths from s to t because ???

Application: Maximum matching

We turn it into a maximum flow problem

Claim: $\exists$Matching of size M $\Leftrightarrow$ $\exists$flow of value M

Proof:

$\Rightarrow$: Clear

$\Leftarrow$: Let f be a flow. There are $|f|$ units.

The $|f|$ units form $|f|$ edge-disjoint paths from s to t because all edges have capacity 1.

So there are exactly $|f|$ edges (u,v) in $(L \cup R, E)$ with $f(u,v) = 1$.

Application: Maximum matching

We turn it into a maximum flow problem

Claim: ∃Matching of size M $\Leftrightarrow$ ∃flow of value M

Proof:

$\Rightarrow$: Clear

$\Leftarrow$: Let f be a flow. There are |f| units.

The |f| units form |f| edge-disjoint paths from s to t because all edges have capacity 1.

So there are exactly |f| edges (u,v) in $(L \cup R, E)$ with $f(u,v) = 1$.

No two vertices in L and R shares these edges because ???

Application: Maximum matching

We turn it into a maximum flow problem

Claim: ∃Matching of size M ⇔ ∃flow of value M

Proof:

⇒: Clear

⇐: Let f be a flow. There are |f| units.

The |f| units form |f| edge-disjoint paths from s to t because all edges have capacity 1.

So there are exactly |f| edges (u,v) in $(L \cup R, E)$ with f(u,v) = 1.

No two vertices in L and R shares these edges because each edge touching s or t has capacity 1. So the |f| edges form a matching.□