15.082J & 6.855J & ESD.78J
October 7, 2010

Introduction to Maximum Flows

The Max Flow Problem

G	=	(N,A)
x_{ij}	=	flow on arc (i,j)
u_{ij}	=	capacity of flow in arc (i,j)
s	=	source node
t	=	sink node

Maximize v

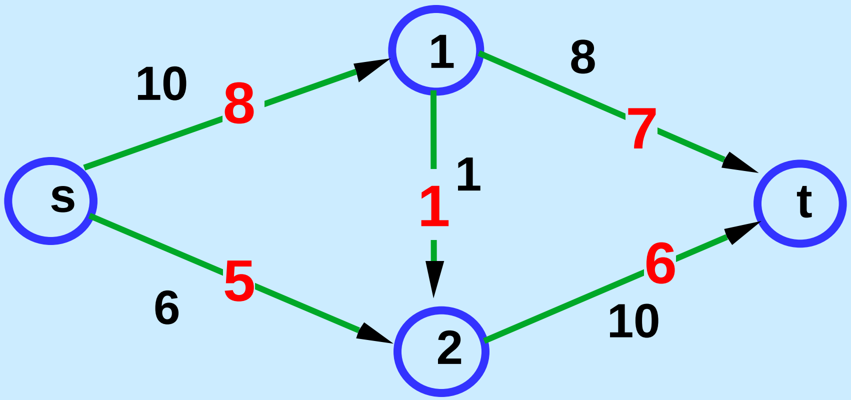
Subject to $\sum_j x_{ij} - \sum_k x_{ki} = 0$ for each $i \neq s,t$

$\sum_j x_{sj} = v$

$0 \le x_{ij} \le u_{ij}$ for all $(i,j) \in A$.

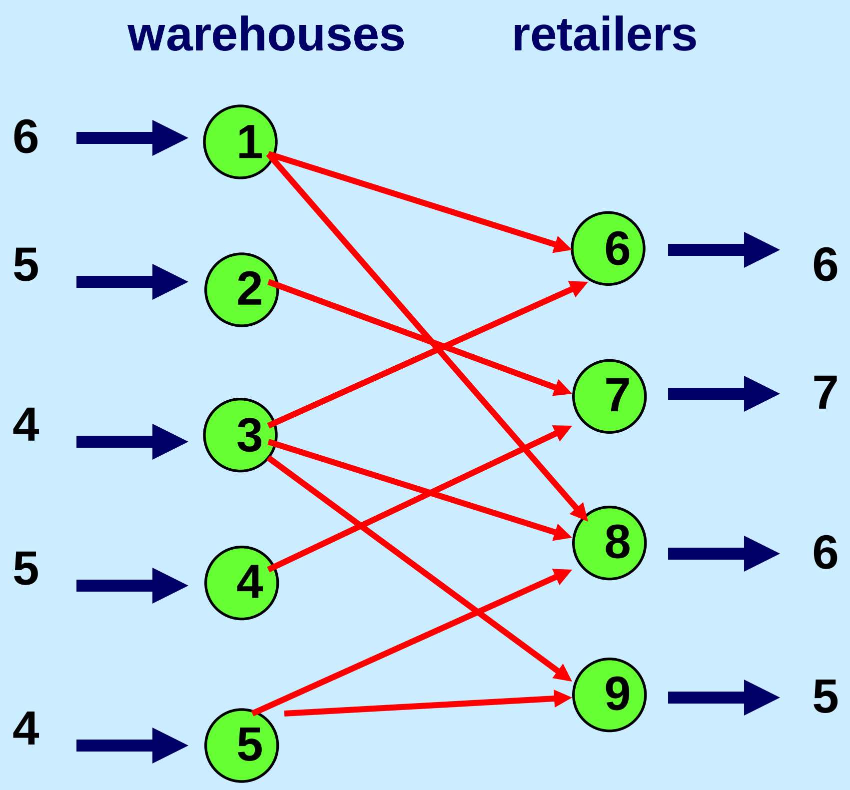
Maximum Flows

We refer to a flow x as ***maximum*** if it is feasible and maximizes v. Our objective in the max flow problem is to find a maximum flow.



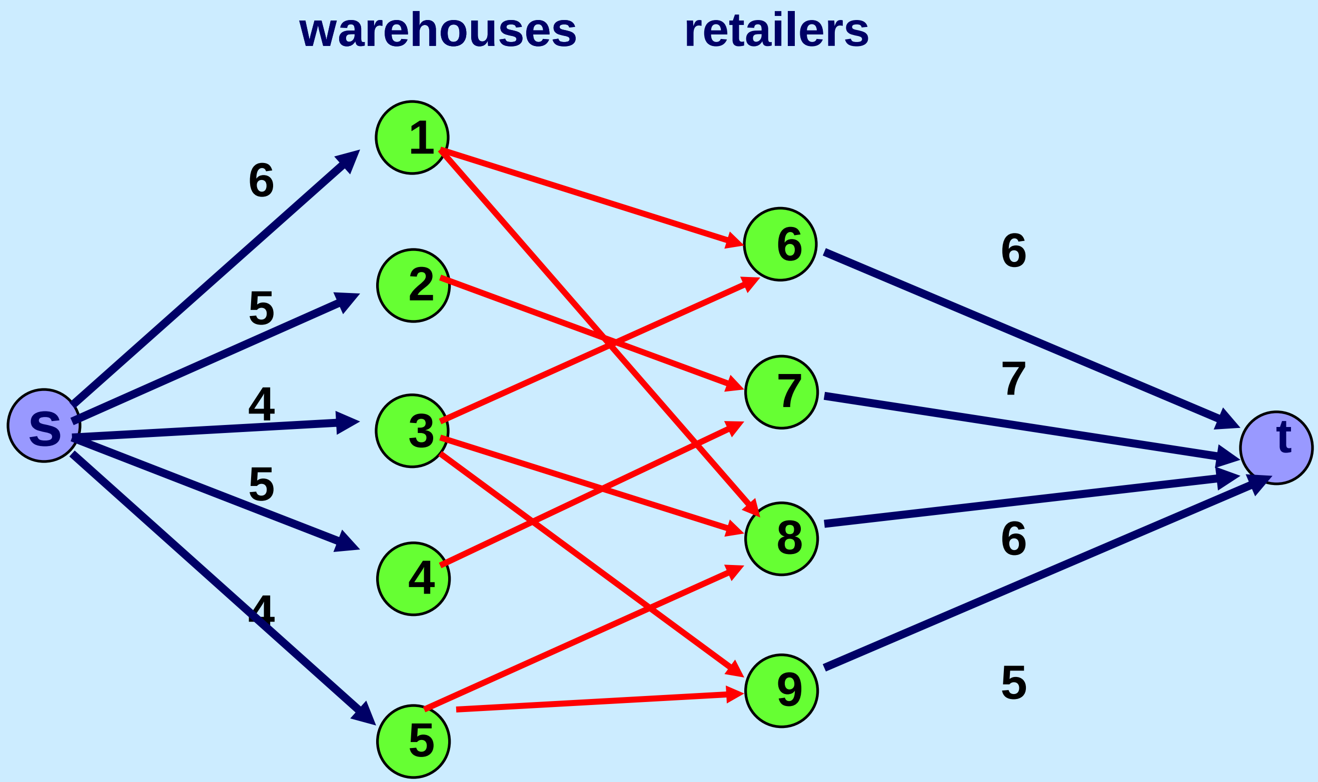
A max flow problem. Capacities and a non-optimum flow.

The feasibility problem: find a feasible flow



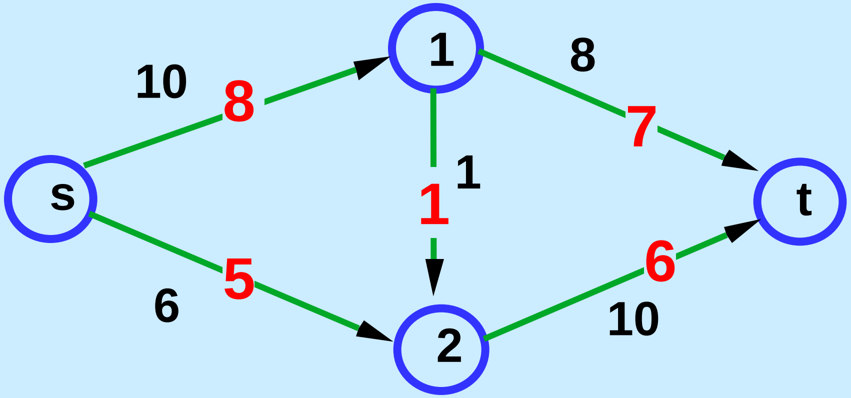
Is there a way of shipping from the warehouses to the retailers to satisfy demand?

Transformation to a max flow problem

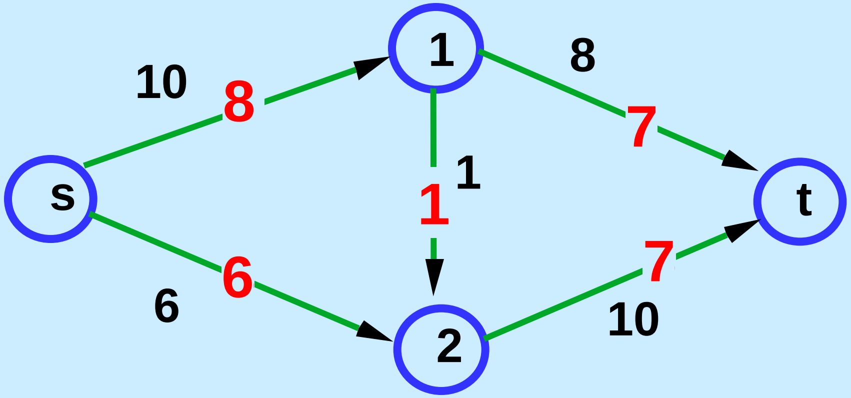


There is a 1-1 correspondence with flows from s to t with 24 units (why 24?) and feasible flows for the transportation problem.

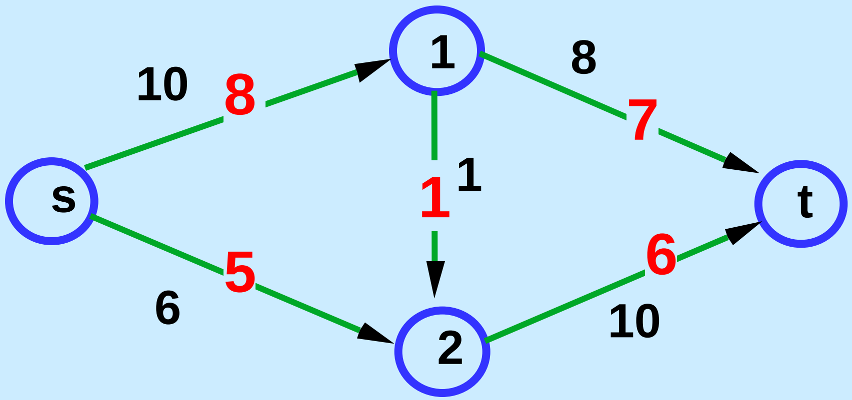
sending flows along s-t paths



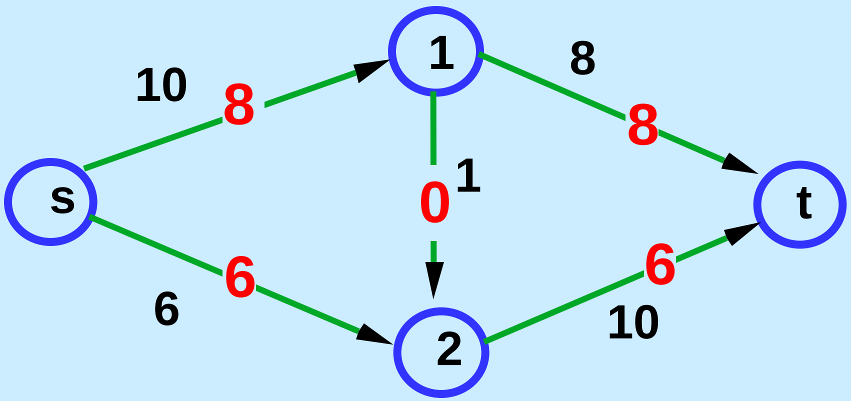
One can find a larger flow from s to t by sending 1 unit of flow along the path s-2-t



A different kind of path

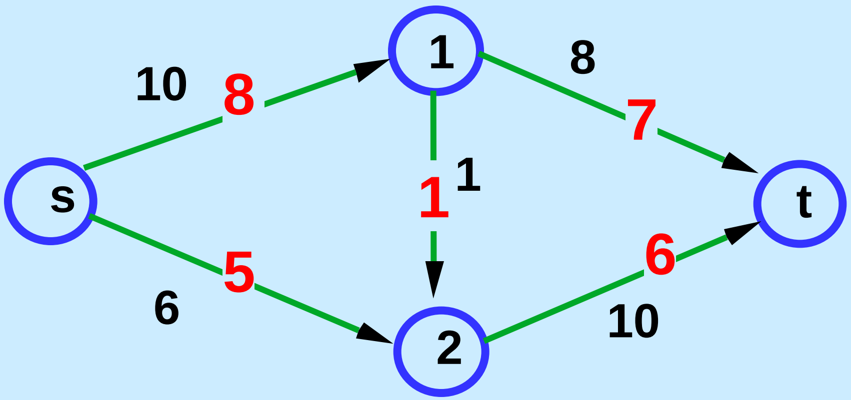


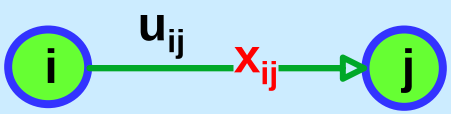
One could also find a larger flow from s to t by sending 1 unit of flow along the path s-2-1-t. (Backward arcs have their flow decreased.)

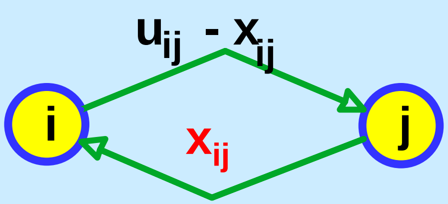


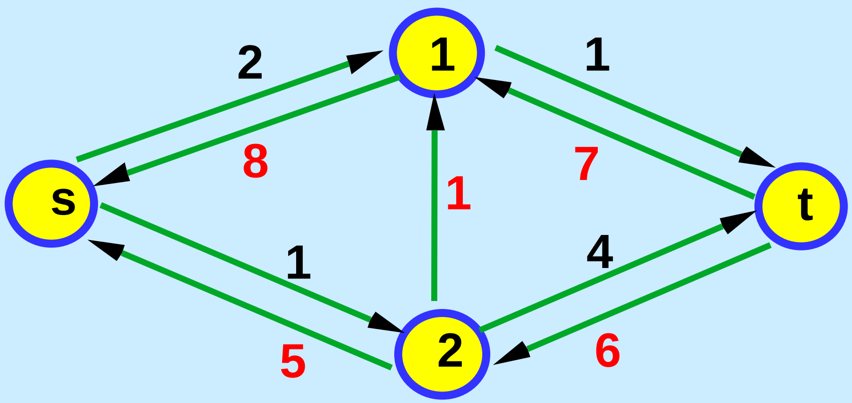
Decreasing flow in (1, 2) is mathematically equivalent to sending flow in (2, 1) w.r.t. node balance constraints.

The Residual Network









The *Residual Network* G(x)

We let r_{ij} denote the *residual capacity* of arc (i,j)

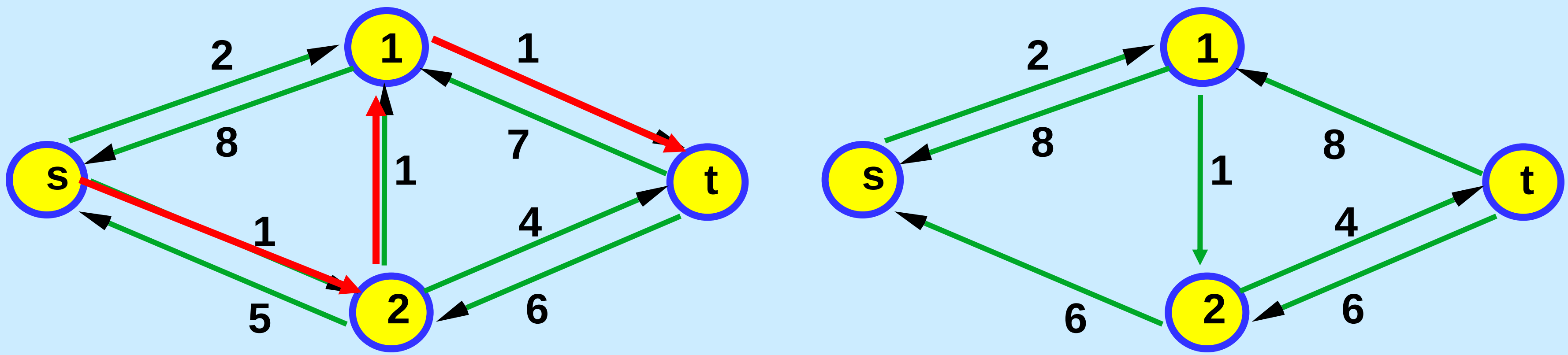
A Useful Idea: Augmenting Paths

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

The *residual capacity* of the augmenting path P is ™(P) = min{r_{ij} : (i,j) ∈ P}.

To *augment along P* is to send ™ (P) units of flow along each arc of the path. We modify x and the residual capacities appropriately.

r_{ij} := r_{ij} - ™(P) and r_{ji} := r_{ji} + ™(P) for (i,j) ∈ P.



The Ford Fulkerson Maximum Flow Algorithm

x := 0;

create the residual network G(x);

while there is some directed path from s to t in G(x) do

- **let P be a path from s to t in G(x);**
- **™ := ™(P);**
- **send ™ units of flow along P;**
- **update the r's;**

Ford-Fulkerson Algorithm Animation

To prove correctness of the algorithm

Invariant: at each iteration, there is a feasible flow from s to t.

Finiteness (assuming capacities are integral and finite):

- **The residual capacities are always integer valued**
- **The residual capacities out of node s decrease by at least one after each update.**

Correctness

- **If there is no augmenting path, then the flow must be maximum.**
- **max-flow min-cut theorem.**

Integrality

Assume that all data are integral.

Lemma*: *At each iteration all residual capacities are integral.

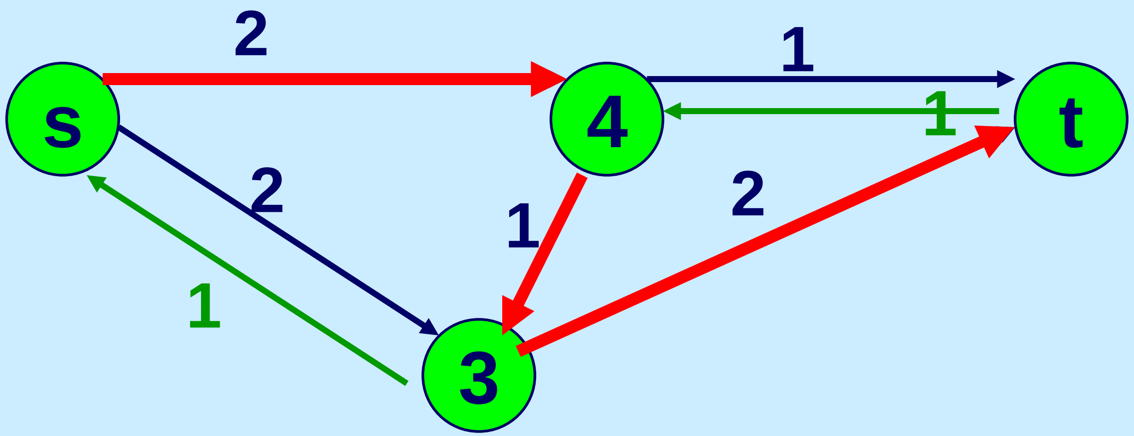
Proof. It is true at the beginning. Assume it is true after the first k-1 augmentations, and consider augmentation k along path P.

The residual capacity ™ of P is the smallest residual capacity on P, which is integral.

After updating, we modify residual capacities by 0, or ™, and thus residual capacities stay integral.

Theorem. The Ford-Fulkerson Algorithm is finite

Proof. The capacity of each augmenting path is at least 1.



r_{sj} decreases for some j.

So, the sum of the residual capacities of arcs out of s decreases at least 1 per iteration.

Number of augmentations is O(nU), where U is the largest capacity in the network.

Mental Break

What are aglets?

The plastic things on the ends of shoelaces.

How fast does the quartz crystal in a watch vibrate?

About 32,000 times per second.

If Barbie (the doll) were life size and 5’ 9” tall, how big would her waist be?

18 inches. Incidentally, Barbie’s full name is Barbara Millicent Roberts

Mental Break

True or false. In Alaska it is illegal to shoot a moose from a helicopter or any other flying vehicle.

True.

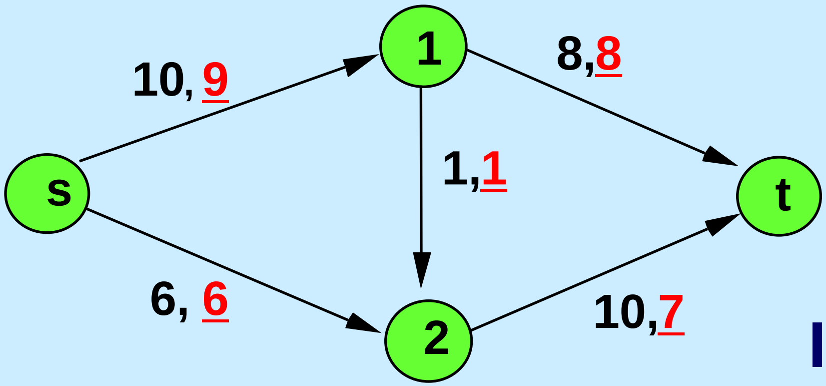
True or false. In Athens, Georgia, a driver's license can be taken away by law if the driver is deemed either "unbathed" or "poorly dressed."

False. However, it is true for Athens, Greece.

In Helsinki, Finland that don't give parking tickets to illegally parked cars. What do they do instead?

They deflate the tires of the car.

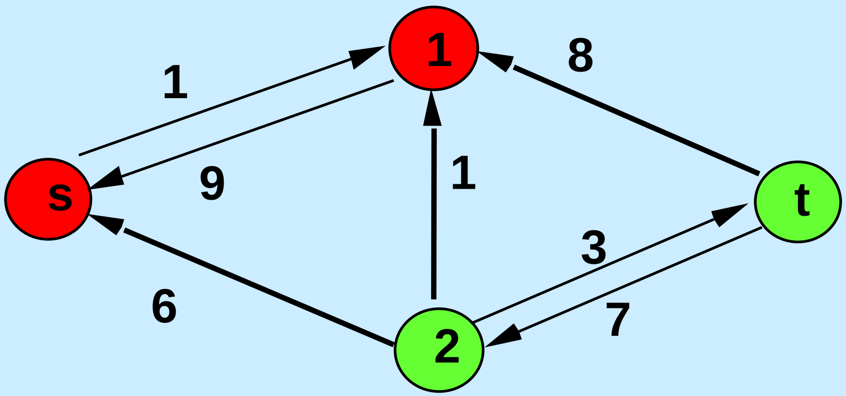
To be proved: If there is no augmenting path, then the flow is maximum

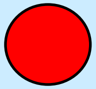


G(x) = residual network for flow x.

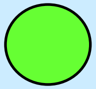
x* = final flow

If there is a directed path from i to j in G, we write i ➔ j.



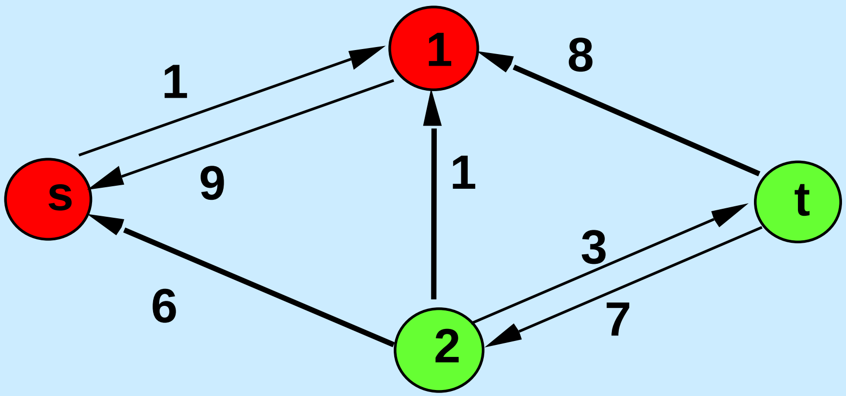


S* = { j : s ➔ j in G(x*)}



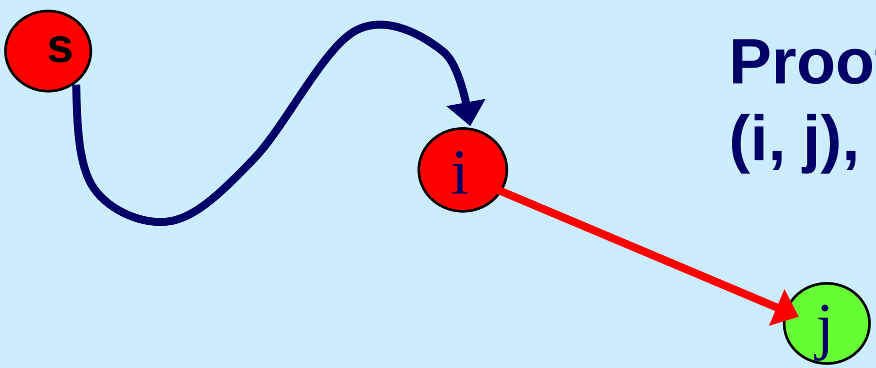
T* = N\S*

Lemma: there is no arc in G(x*) from S* to T*



S* = { j : s ➔ j in G(x*)}

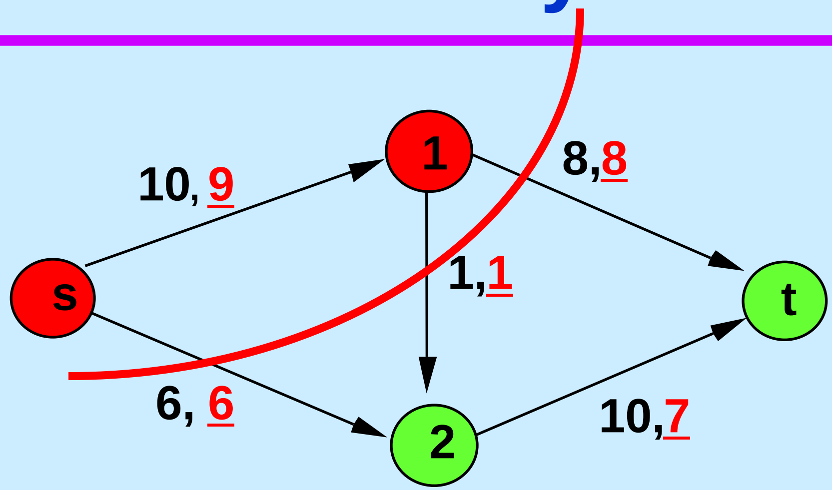
T* = N\S*



Proof. If there were such an arc (i, j), then j would be in S*.

We will use this Lemma in 6 slides.

Cut Duality Theory



An ***(s,t)-cut*** in a network G = (N,A) is a partition of N into two disjoint subsets S and T such that s ∈ S and t ∈ T, e.g., S = { s, 1 } and T = { 2, t }.

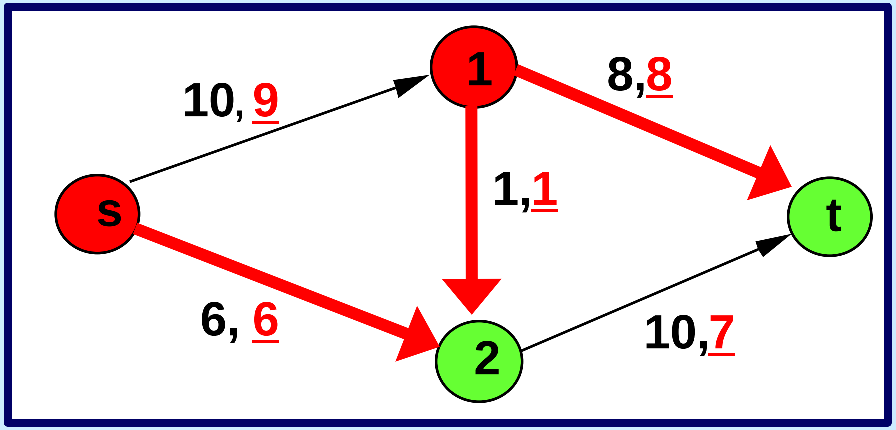
The ***capacity*** of a cut (S,T) is

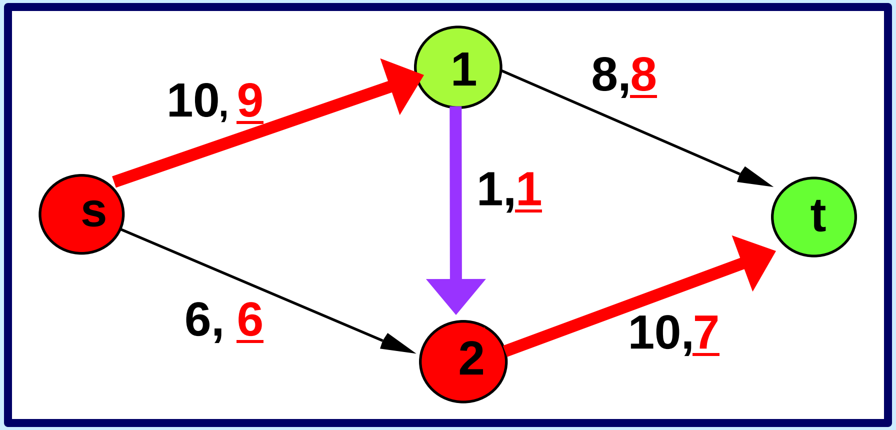
$$CAP(S,T) \quad = \quad \sum_{i \in S} \sum_{j \in T} u_{ij}$$

The flow across a cut

We define the *flow across the cut* (S,T) to be

$$F_x(S,T) = \sum_{i\in S}\sum_{j\in T} x_{ij} - \sum_{i\in S}\sum_{j\in T} x_{ji}$$





If S = {s, 1}, then

$F_x(S,T) = 6 + 1 + 8 = 15$

If S = {s, 2}, then

$F_x(S,T) = 9 - 1 + 7 = 15$

Max Flow Min Cut

***Theorem**. (Max-flow Min-Cut). The maximum flow value is the minimum value of a cut.*

Proof. The proof will rely on the following three lemmas:

Lemma 1. **For any flow x, and for any s-t cut (S, T), the flow out of s equals $F_x(S, T)$.**

Lemma 2. **For any flow x, and for any s-t cut (S, T), $F_x(S, T) \leq CAP(S, T)$.**

Lemma 3. **Suppose that x* is a feasible s-t flow with no augmenting path. Let S* = {j : s ➔ j in G(x*)} and let T* = N\S. Then $F_{x^*}(S^*, T^*) = CAP(S^*, T^*)$.**

Proof of Theorem (using the 3 lemmas)

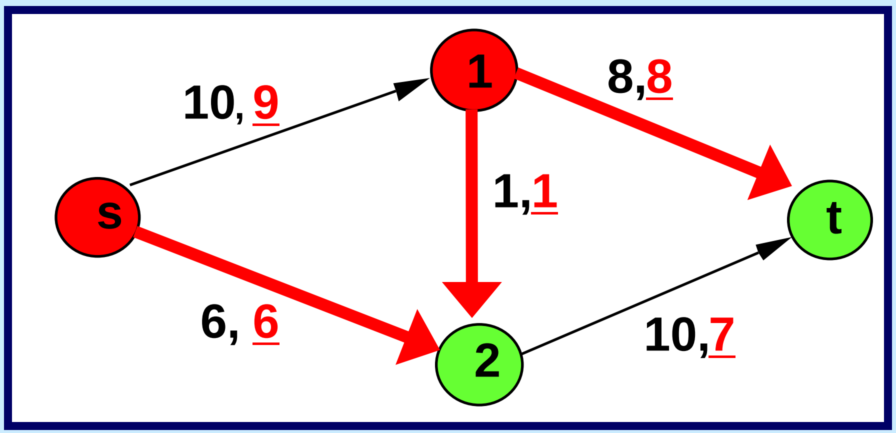
Let x' be a maximum flow
Let v' be the maximum flow value
Let x* be the final flow.
Let v* be the flow out of node s (for x*)
Let S* be nodes reachable in G(x*) from s.
Let T* = N\S*.

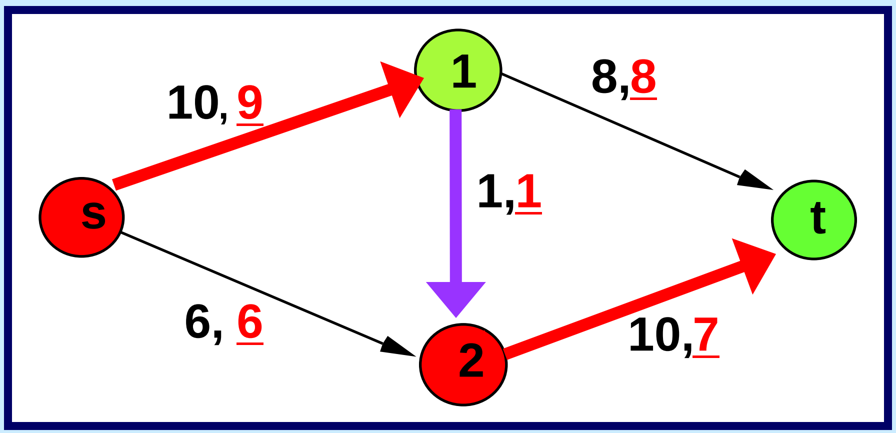
1.	$v^* \le v'$	by definition of v'
2.	$v' = F_{x'}(S^*, T^*)$	by Lemma 1.
3.	$F_{x'}(S^*, T^*) \le CAP(S^*, T^*)$	by Lemma 2.
4.	$v^* = F_{x^*}(S^*, T^*) = CAP(S^*, T^*)$	by Lemmas 1,3.

Thus all inequalities are equalities and v* = v'.

Proof of Lemma 1

Proof. Add the conservation of flow constraints for each node $i \in S - \{s\}$ to the constraint that the flow leaving s is v. The resulting equality is $F_x(S,T) = v$.





$$x_{s1} + x_{s2} = v$$
$$x_{12} + x_{1t} - x_{s1} = 0$$
$$\overline{\qquad\qquad\qquad\qquad}$$
$$x_{s2} + x_{12} + x_{1t} = v$$

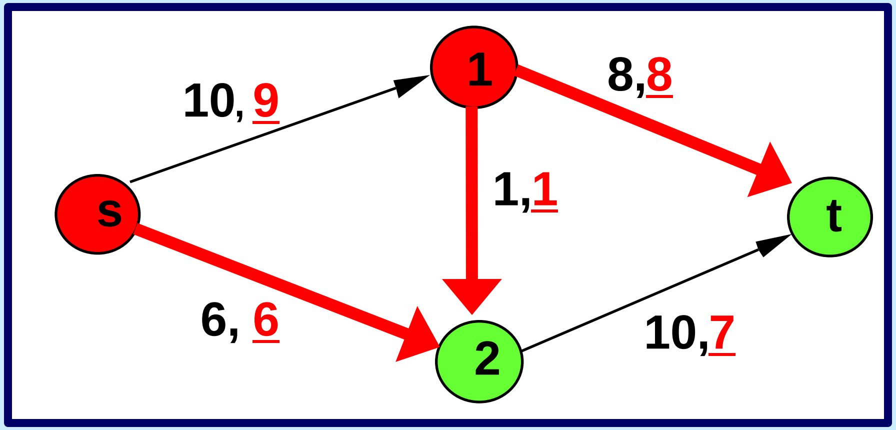
$$x_{s1} + x_{s2} = v$$
$$x_{2t} - x_{s2} - x_{12} = 0$$
$$\overline{\qquad\qquad\qquad\qquad}$$
$$x_{s1} - x_{12} + x_{2t} = v$$

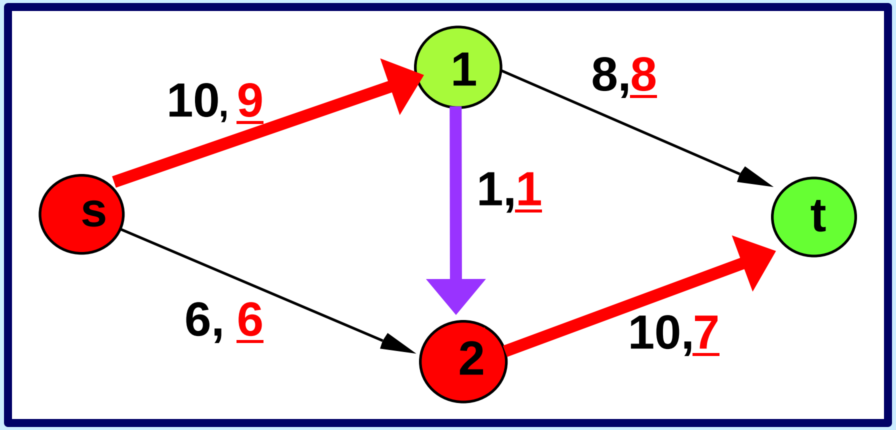
Proof of Lemma 2

Proof. If $i \in S$, and $j \in T$, then $x_{ij} \leq u_{ij}$. If $i \in T$, and $j \in S$, then $x_{ij} \geq 0$.

$$F_x(S,T) = \sum_{i \in S} \sum_{j \in T} x_{ij} - \sum_{i \in S} \sum_{j \in T} x_{ji}$$

$$CAP(S,T) = \sum_{i \in S} \sum_{j \in T} u_{ij} - \sum_{i \in S} \sum_{j \in T} 0$$





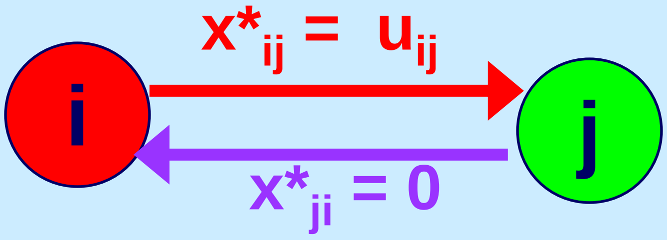
CAP(S, T) = 15

CAP(S, T) = 16

Proof of Lemma 3.

We have already seen that there is no arc from S^* to T^* in $G(x^*)$.

$i \in S^*$ and $j \in T^*$ $\Rightarrow$ $x^*_{ij} = u_{ij}$ and $x^*_{ji} = 0$



Otherwise, there is an arc (i, j) in $G(x^*)$

Therefore $F_{x^*}(S^*, T^*) = CAP(S^*, T^*)$

Review

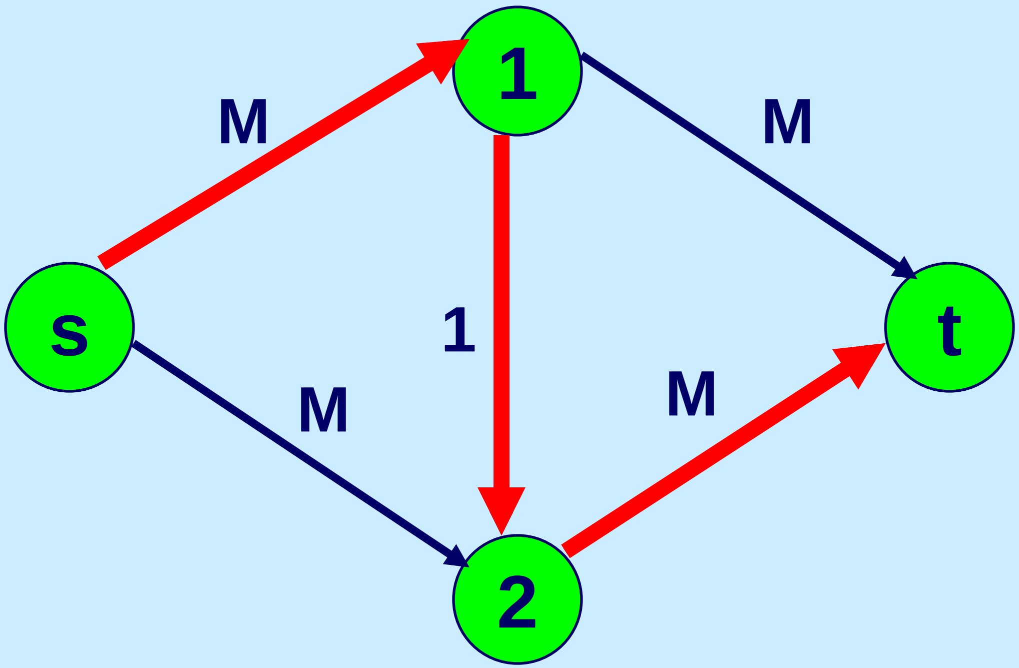
Corollary. If the capacities are finite integers, then the Ford-Fulkerson Augmenting Path Algorithm terminates in finite time with a maximum flow from s to t.

Corollary. If the capacities are finite rational numbers, then the Ford-Fulkerson Augmenting Path Algorithm terminates in finite time with a maximum flow from s to t. (why?)

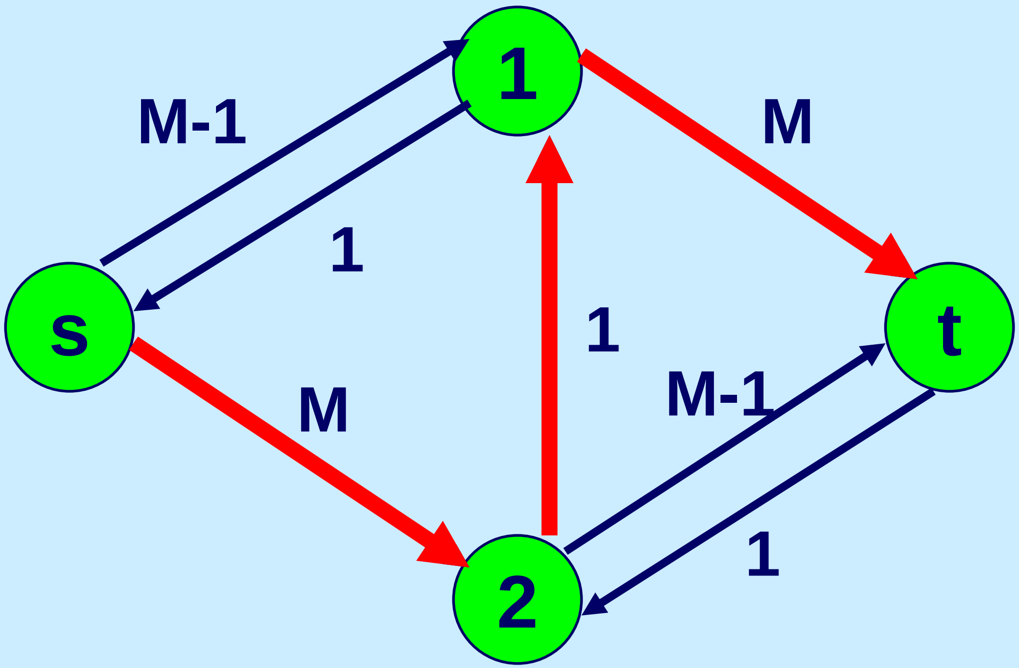
Corollary. To obtain a minimum cut from a maximum flow x*, let S* denote all nodes reachable from s in G(x), and T* = N\S*

Remark. This does not establish finiteness if $u_{ij} = \infty$ or if capacities may be irrational.

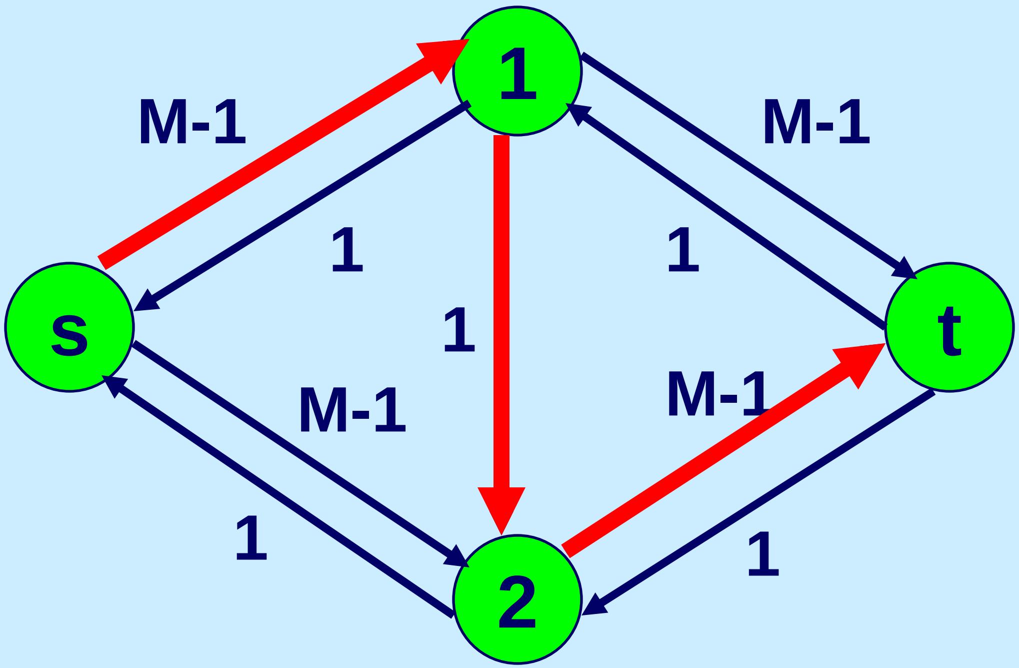
A simple and very bad example



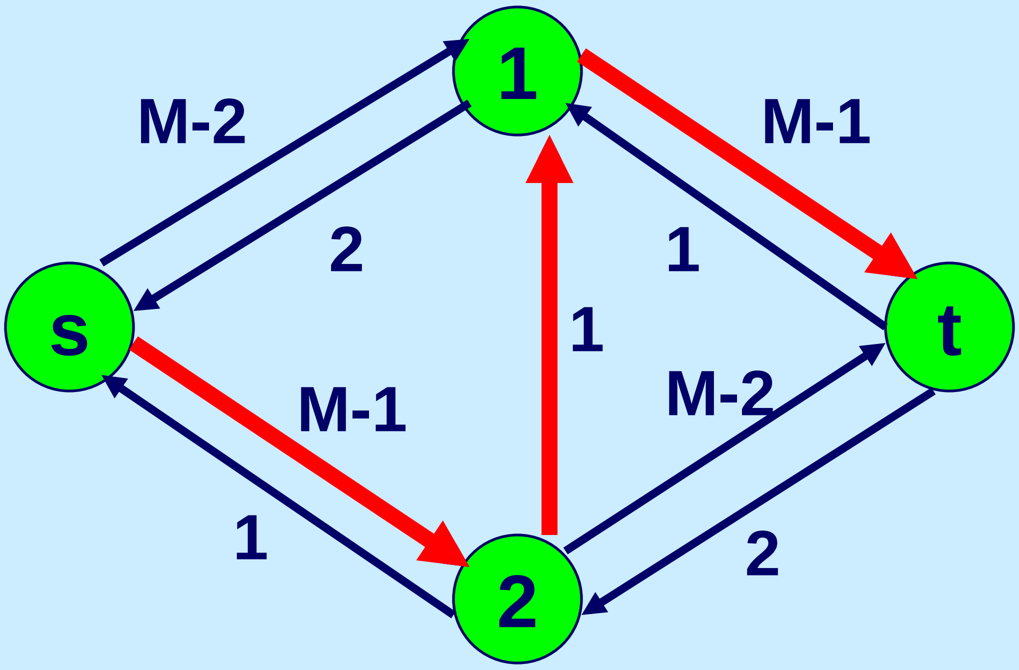
After 1 augmentation



After two augmentations



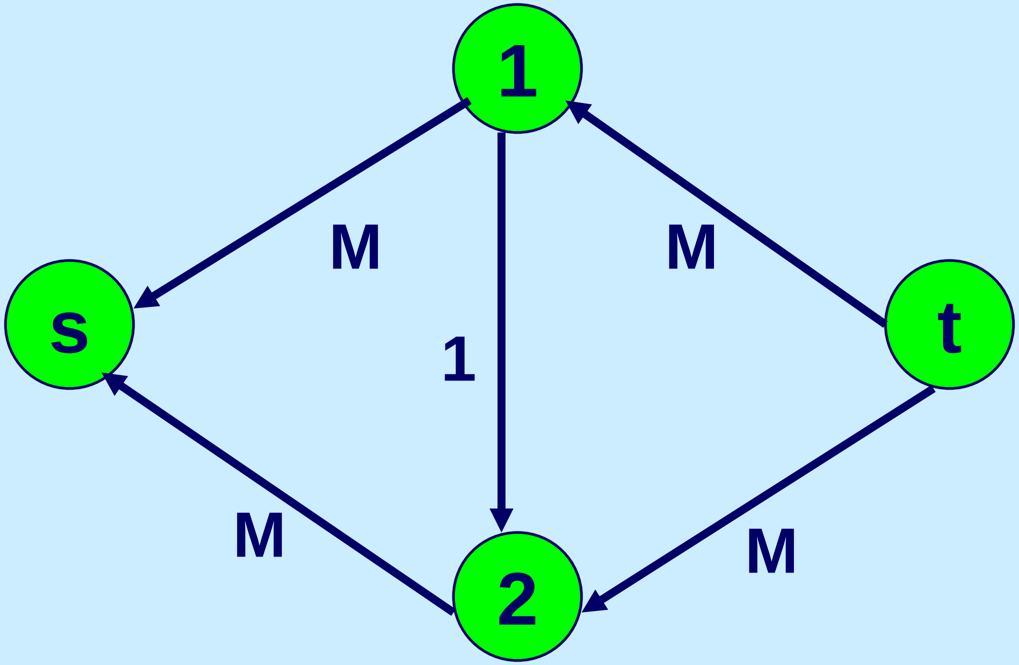
After 3 augmentations



And so on

• • •

After 2M augmentations



An even worse example

In Exercise 6.48, there is an example that takes an infinite number of augmentations on irrational data, and does not converge to the correct flow.

But we shall soon see how to solve max flows in a polynomial number of operations, even if data can be irrational.

Summary and Extensions

1. **Augmenting path theorem**

2. **Ford-Fulkerson Algorithm**

3. **Duality Theory.**

4. **Next Lecture:**
   - **Polynomial time variants of FF algorithm**
   - **Applications of Max-Flow Min-Cut**

15.082J / 6.855J / ESD.78J Network Optimization
Fall 2010