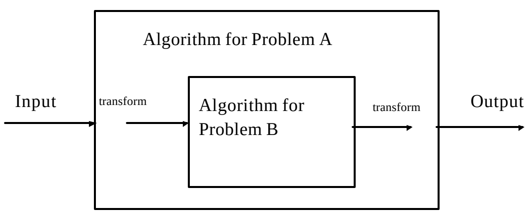
CS140: Algorithms

Z Sweedyk
Lecture 8
2/27/01

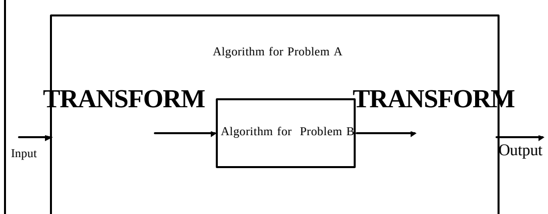
Algorithm Design Techniques

- Induction
- **Reduction**

Reduction: A ∝ B



Reduction: A ∝ B

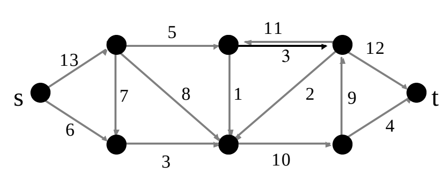


Network Flow

Network Flow:
- Solving network flow problems
- Reductions to the network flow problem

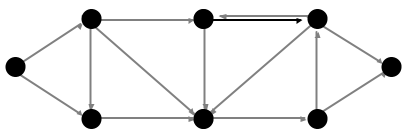
Max Flow in a Network

- Input: Flow Network



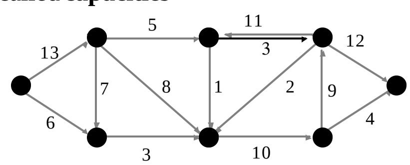
Flow Network

- Directed graph



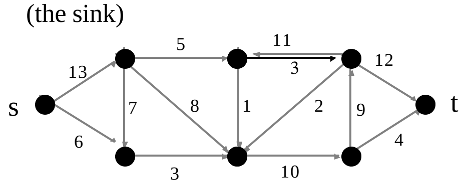
Flow Network

- Directed graph with positive edge weights called **capacities**



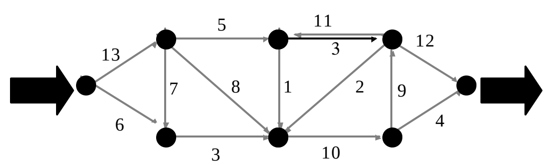
Flow Network

- Directed graph with positive edge capacities and two special vertices s (the source) and t (the sink)



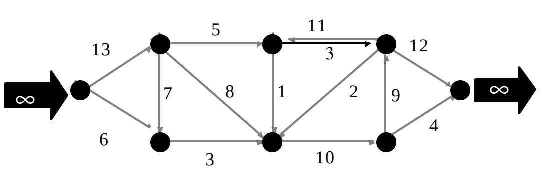
Max Flow in a Network

- Output: Maximum flow that can be pumped through the network from s to t.



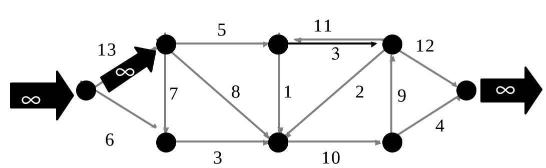
The Rules

0. The source has infinite input capacity. The sink has infinite output capacity.



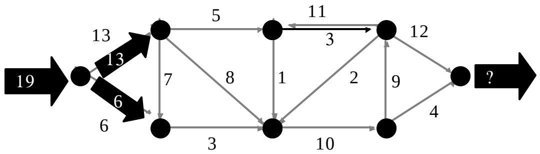
The Rules

1. The total flow into a node must equal the total flow out of a node.



The Rules

2. The flow along an edge cannot exceed its capacity. (No edge means 0 capacity.)

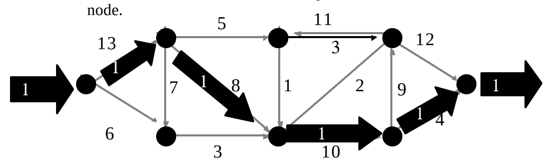


2/27/07 CS140 Lec 8 13

What is a feasible flow?

Any flow that satisfies the rules.

1. The flow along an edge cannot exceed its capacity
2. The total flow into a node must equal the total flow out of a node.



2/27/07 CS140 Lec 8 14

Conservation of Flow

- Flow in = Flow out (Rule 2)
- Flow into s = Flow out of t

2/27/07 CS140 Lec 8 15

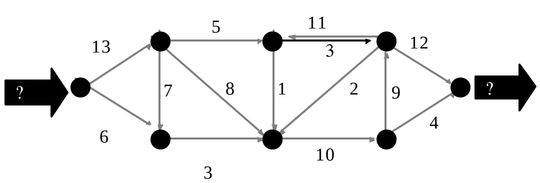
Network flow

- Flow into s = Flow out of t = Network flow

2/27/07 CS140 Lec 8 16

What is the max flow?

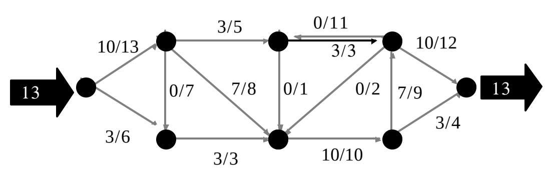
1. The flow along an edge cannot exceed its capacity
2. The total flow into a node must equal the total flow out of a node.



2/27/07 CS140 Lec 8 17

MAX FLOW?

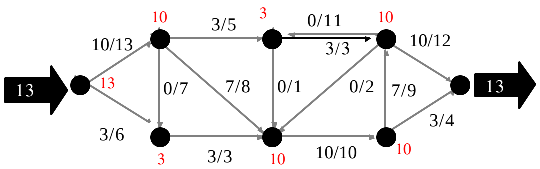
1. **The flow along an edge cannot exceed its capacity**
2. The total flow into a node must equal the total flow out of a node.



2/27/07 CS140 Lec 8 18

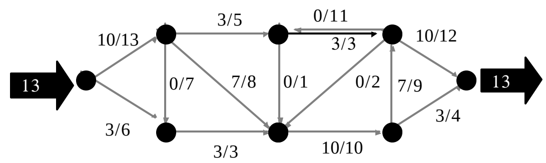
MAX FLOW?

1. The flow along an edge cannot exceed its capacity
2. **The total flow into a node must equal the total flow out of a node.**



Feasible Flow! But is it a Max Flow?

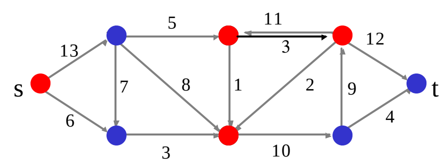
1. The flow along an edge cannot exceed its capacity
2. The total flow into a node must equal the total flow out of a node.



Cut of a Flow Network

- Cut (of a flow network) is a partition of the vertices into two sets R and B such that s ε R and t ε B

Example of a cut

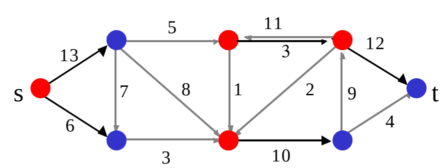


Capacity of a cut

- The capacity of a cut (R,B) of a network is the sum of the capacities of the edges that go from a R vertex to a B vertex.

Example of cut capacity

(R,B) has capacity 41

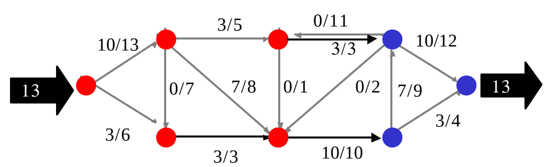


Max Flow-Min Cut Theorem

- The maximum flow of a network is equal to the capacity of the minimum capacity cut in the network.

Feasible Flow!
But is it a Max Flow?

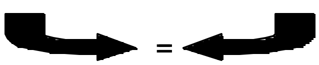
Here is a matching cut! But is it a min-cut?



Max Flow-Min Cut Theorem

- The maximum flow of a network is equal to the minimum capacity of any cut in the network.

- Flow $\leq$ Max flow = Min cut $\leq$ Cut

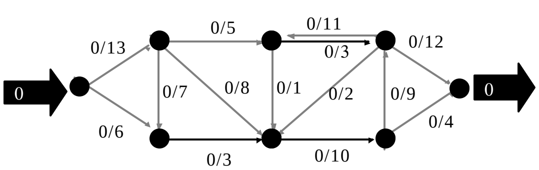


Network Flow
What is ahead?

- Algorithms for finding maximum flow in a network
- Reducing problems to the max flow problem

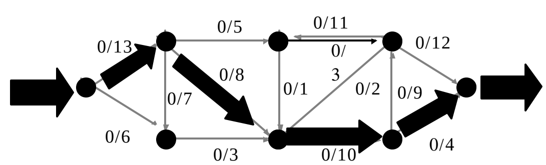
Greedy approach
(doesn't quite work)

- Find an augmenting path



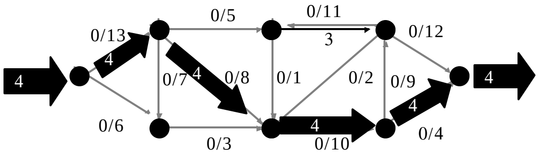
Augmenting path

Find an augment path: An s => t path in which the capacity of each edge exceeds its flow.



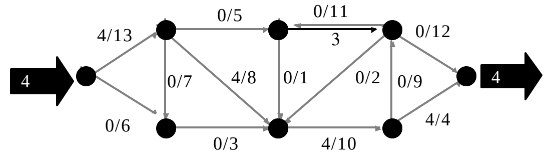
Augmenting path

1. Find an augment path
2. Push as much flow through the path as possible.



Repeat

1. Find an augment path
2. Push as much flow through the path as possible.



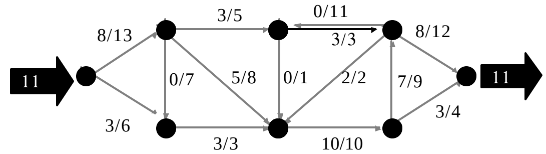
Greedy: Augmenting paths

- Does it work?

  (i.e. does it lead to a max flow?)

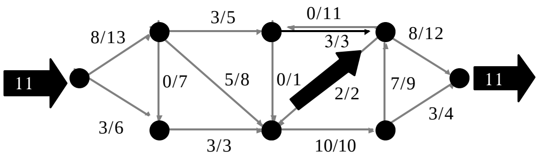
Augmenting Path in network doesn't always work!

We can get stuck! What should we do?



Augmenting Path in network doesn't always work!

What should we do? Reroute!



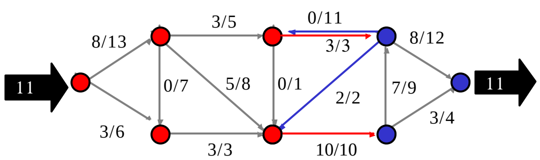
One more idea

Given a network and a flow:

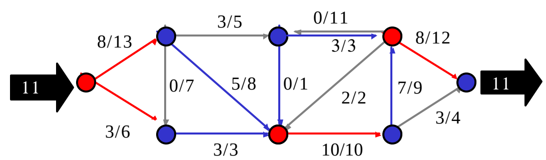
The flow across a cut (R,B) is

Σ f(<u,v>) - Σ f(<v,u>)

Flow across (R,B)



Flow across (R,B)



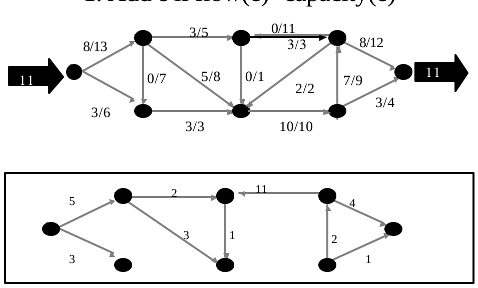
Flow across a cut

- Conservation of flow
- The flow across any cut equals the network flow

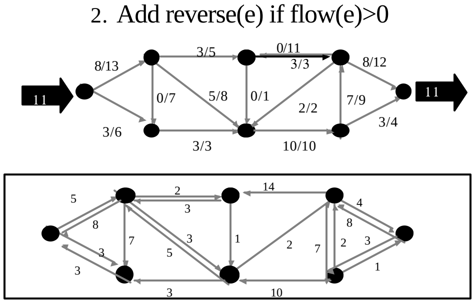
Flow across a cut

- Conservation of flow + max flow/min cut theorem
- If flow is max then for minimum capacity cut (R,B): Σ f(<v,u>) = 0

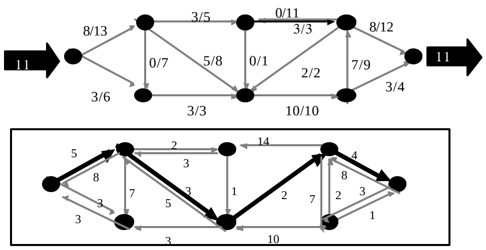
Build residual graph
1. Add e if flow(e)<capacity(e)



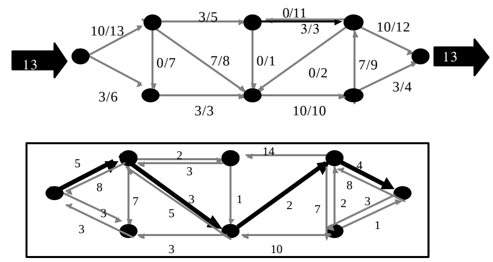
Build residual graph
2. Add reverse(e) if flow(e)>0



Find s→t path in the residual graph

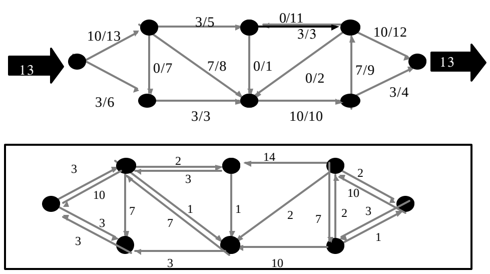


Augment the s→t path in the network



And then …

Repeat until residual graph has no s→t path



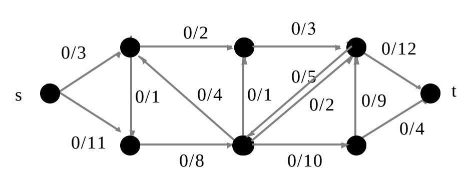
Ford-Fulkerson

- Build residual graph
- If s → t path does not exist then return current flow
- Find s → t path
- Augment flow in network
- Repeat

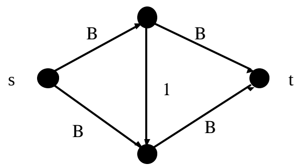
Augmenting Path Method (Ford-Fulkerson)

- Is it correct?
- Is it efficient?

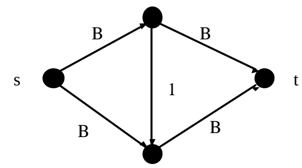
Exercise



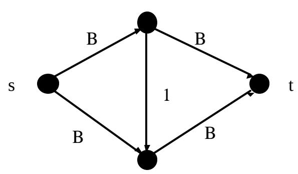
Problems: How many augmentations could this take?



B Augmentations – What is size of input?



B Augmentations
Input has size n + m lg(B)



Edmonds-Karp

- Choose shortest s to t path in residual graph to augment
- Total augmentations is O(mn)

Edmonds-Karp

- Build residual graph
- If $s \rightarrow t$ path does not exist then return current flow
- Find shortest $s \rightarrow t$ path
- Augment flow in network
- Repeat