CSE 421: Algorithms

Winter 2014

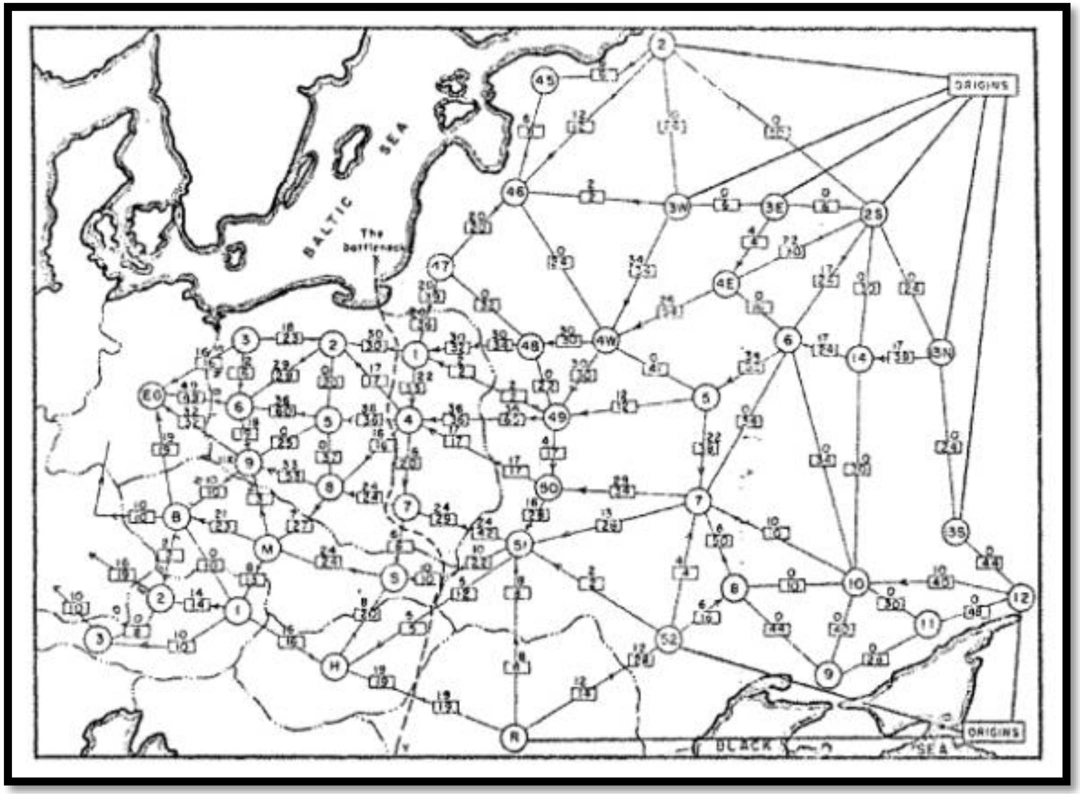
Lecture 19: The max-flow/min-cut theorem

Reading:
Sections 7.1-7.2

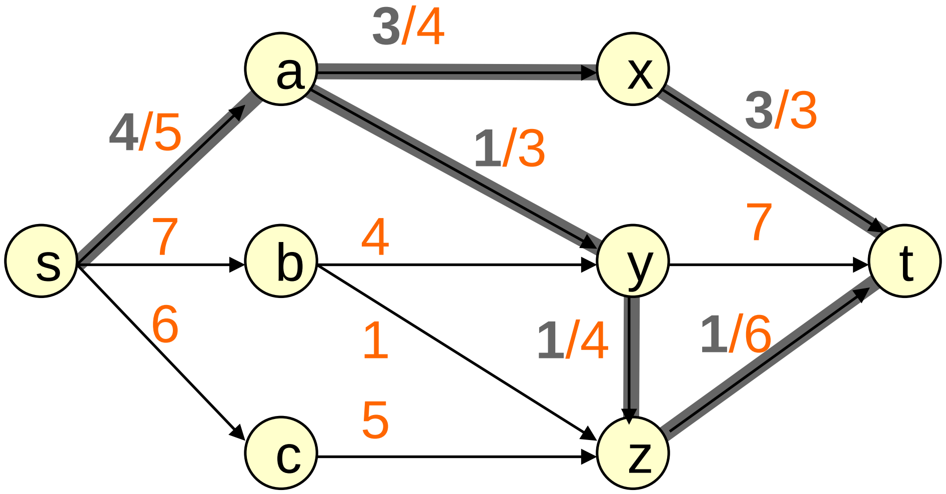
office hours

2:30 – 4 pm

CSE 640



finding maximum-flows (integer capacities)



Ford-Fulkerson method

Start with $f = 0$ for every edge

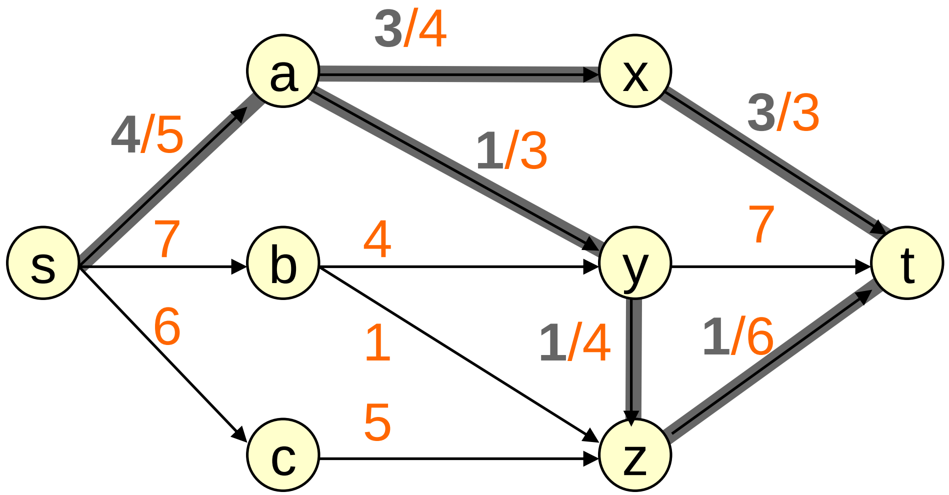
While G_f has an augmenting path, augment.

Questions:

- Does it halt?
- Does it find a maximum flow?
- How fast?

residual capacity

- The residual capacity (w.r.t. f) of (u,v) is $c_f(u,v) = c(u,v) - f(u,v)$ if $f(u,v) \leq c(u,v)$ and $c_f(u,v)=f(v,u)$ if $f(v,u) > 0$



- e.g. $c_f(s,b)=7$; $c_f(a,x) = 1$; $c_f(x,a) = 3$

residual graph & augmenting paths

- The residual graph (w.r.t. f) is the graph $G_f = (V,E_f)$, where $E_f = \{ (u,v) \mid c_f(u,v) > 0 \}$
  - Two kinds of edges

    Forward edges

    $f(u,v)<c(u,v)$ so $c_f(u,v)=c(u,v)-f(u,v)>0$

    Backward edges

    $f(u,v)>0$ so $c_f(v,u) \geq -f(v,u)=f(u,v)>0$

- An augmenting path (w.r.t. f) is a simple $s \rightarrow t$ path in G_f.

augmenting a flow along a path

augment(**f**,**P**)

 $c_P \leftarrow \min_{(u,v)\in P} c_f(u,v)$ "bottleneck(P)"

 for each $e \in P$

  if **e** is a forward edge then

   increase **f**(**e**) by c_P

  else (**e** is a backward edge)

   decrease **f**(**e**) by c_P

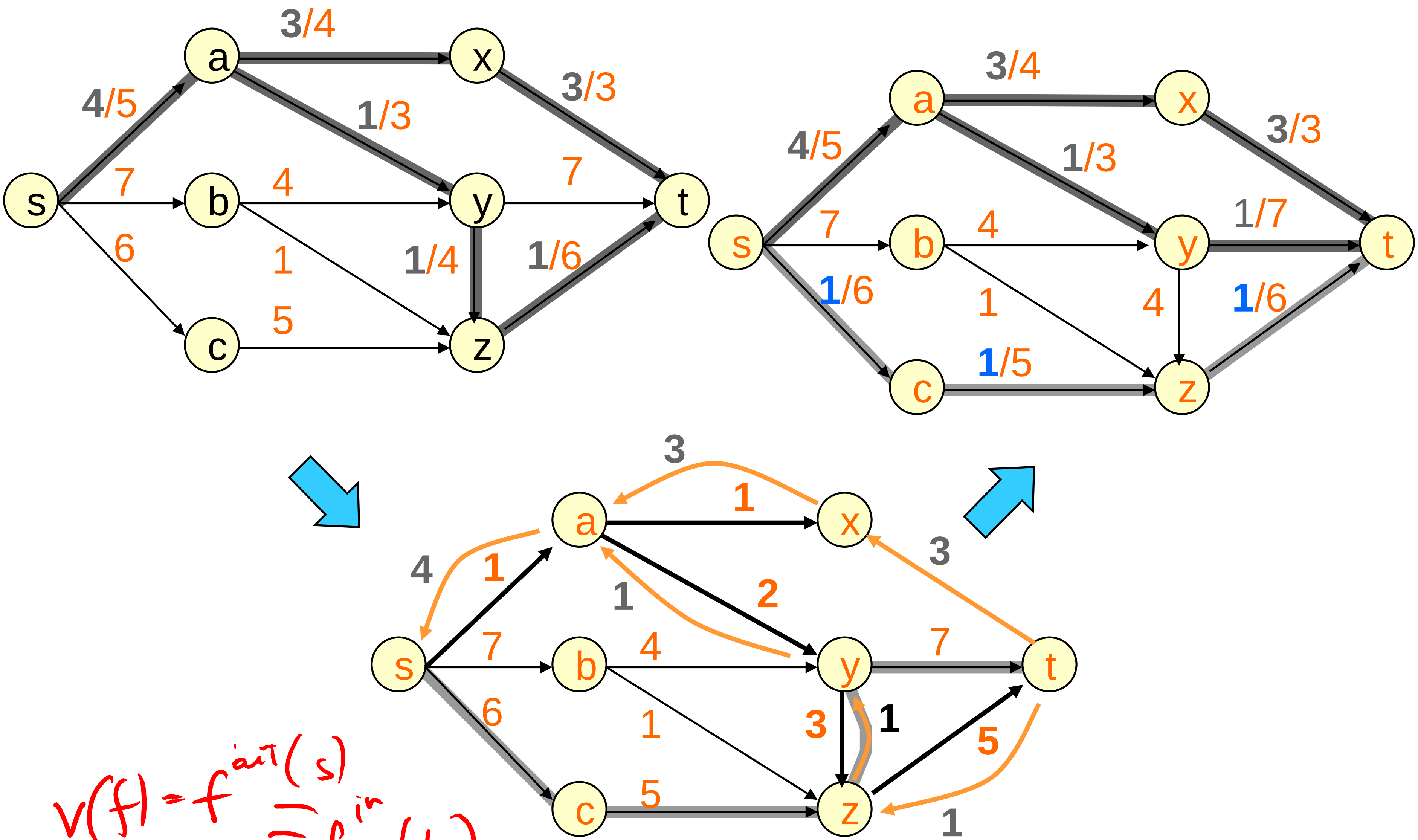
  endif

 endfor

 return(**f**)

augmenting a flow

begin: $f = 0$ $\quad G_f = G$



$v(f) = f^{out}(s) = f^{in}(t)$

last time

Lemma:

If G_f has an augmenting path P, then the function $f'=\text{augment}(f,P)$ is a legal flow.

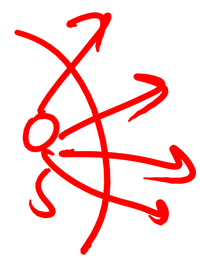
$G, f \longrightarrow G_f \longrightarrow$ find a path P in G_f

$f := \text{augment}(f,P)$

always halts

- At every stage the capacities and flow values are always integers (if they start that way)
- The flow value $\nu(f') = \nu(f) + c_P > \nu(f)$ for

  $f' = \text{augment}(f,P)$
  - Since edges of residual capacity **0** do not appear in the residual graph
- Let $C = \sum_{(s,u)\in E} c(s,u)$
  - $\nu(f) \leq C$
  - **F-F** does at most **C** rounds of augmentation since flows are integers and increase by at least **1** per step

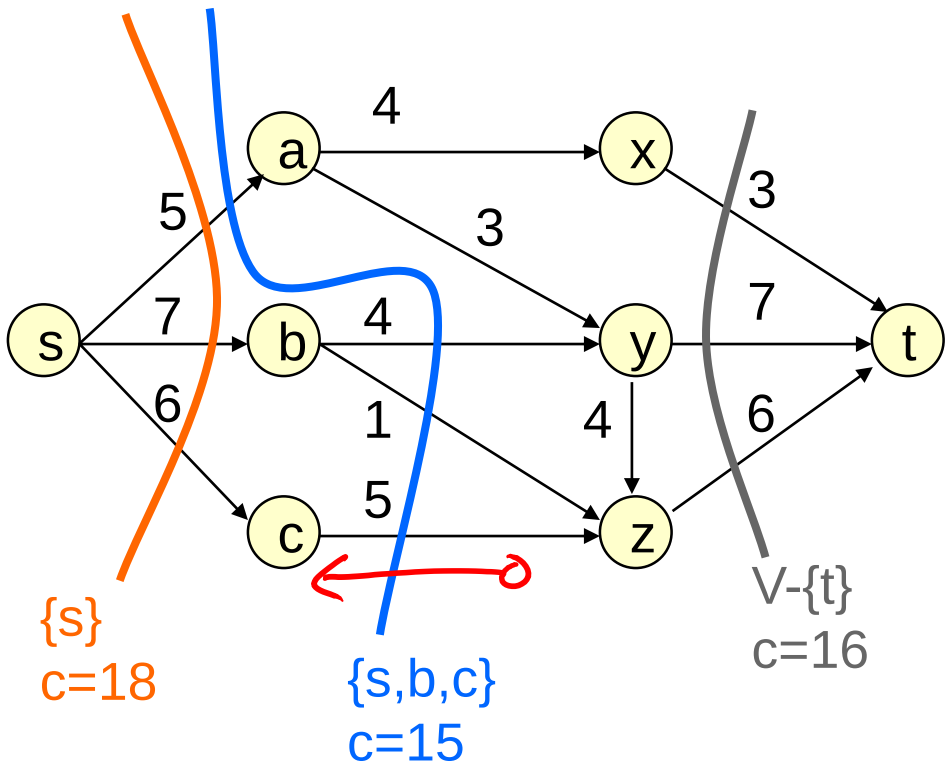


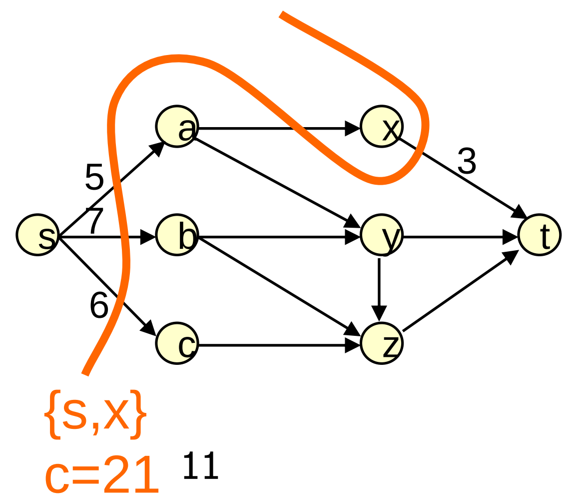
running time

- For $f = 0$, $G_f = G$
- Finding an augmenting path in G_f is graph search $O(n+m)=O(m)$ time
- Augmenting and updating G_f is $O(n)$ time
- Total $O(mC)$ time C = cap. out of s.
- Does it find a maximum flow?
  - Need to show that for every flow f that isn't maximum G_f contains an s-t-path

cuts

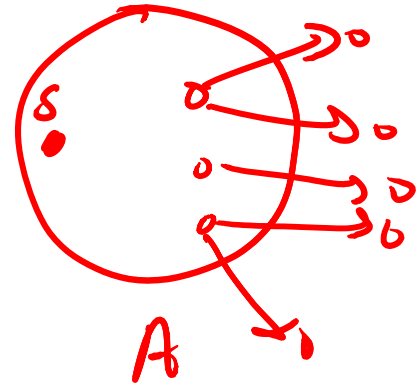
- A partition **(A,B)** of **V** is an ***s-t-cut*** if
  s∈A, t∈B
- **Capacity** of cut **(A,B)** is $\mathbf{c(A,B)} = \sum_{\substack{u \in A \\ v \in B}} \mathbf{c(u,v)}$



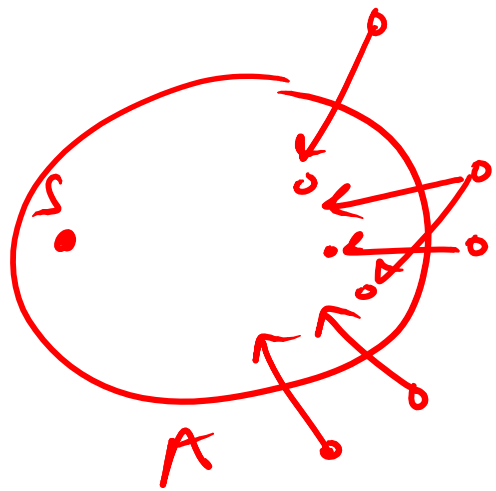


convenient definitions

- $f^{out}(A) = \sum_{v \in A,\, w \notin A} f(v,w)$



- $f^{in}(A) = \sum_{v \in A,\, u \notin A} f(u,v)$



claims we will prove

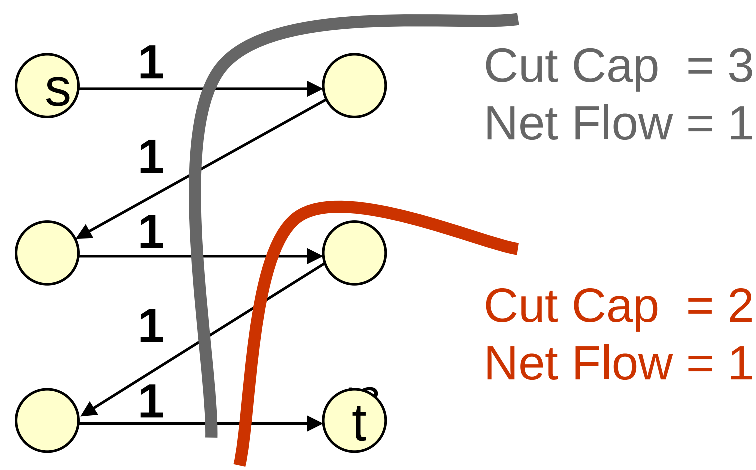
$v(f) \le c(A,B)$ for any cut (A,B)

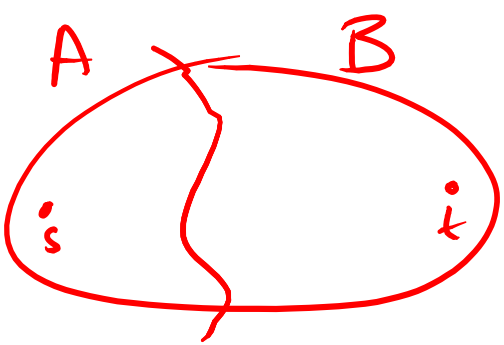
- For any flow **f** and any (s,t)-cut **(A,B)**,
  - the net flow across the cut equals the total flow: $\nu(f) = f^{out}(A) - f^{in}(A)$, and
  - the net flow across the cut cannot exceed the capacity of the cut: $f^{out}(A) - f^{in}(A) \le c(A,B)$

$f^{out}(A) - f^{in}(A) \le f^{out}(A) \le c(A,B)$

- **Corollary:**

  Max flow $\underset{=}{\le}$ Min cut





G, f

proof of claim

- Consider a set A with $s \in A$, $t \notin A$
- $f^{out}(A) - f^{in}(A) = \sum_{v \in A,\, w \notin A} f(v,w) - \sum_{v \in A,\, u \notin A} f(u,v)$
- We can add flow values for edges with both endpoints in A to **both** sums and they would cancel out so
- $f^{out}(A) - f^{in}(A) = \sum_{v \in A} \sum_{w \in V} f(v,w) - \sum_{v \in A} \sum_{u \in V} f(u,v)$

$$= \sum_{v \in A} \left(\sum_{w \in V} f(v,w) - \sum_{u \in V} f(u,v) \right)$$

- $\underline{v(f) = f^{out}(s) \text{ and } f^{in}(s)=0}$

$$= \sum_{v \in A} f^{out}(v) - f^{in}(v)$$

$$= f^{out}(s) = v(f)$$

proof of claim

$$\nu(f) = f^{out}(A) - f^{in}(A)$$

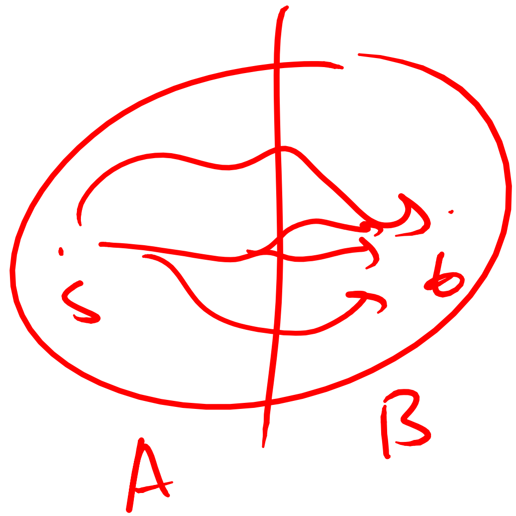
$$\le f^{out}(A)$$

$$= \sum_{v \in A,\ w \notin A} f(v,w)$$

$$\le \sum_{v \in A,\ w \notin A} c(v,w)$$

$$\le \sum_{v \in A,\ w \in B} c(v,w)$$

$$= c(A,B)$$



amount of flow from s->t $\le c(A,B)$.

max flow/min cut theorem

Theorem: For any flow $\mathbf{f}$, if G_f has no augmenting path then there is some s-t-cut (A,B) such that $\nu(f)=c(A,B)$ (proof on next slide)

$v(f') \le c(A,B)$ $\quad c(A,B) = v(f) \le c(A',B')$

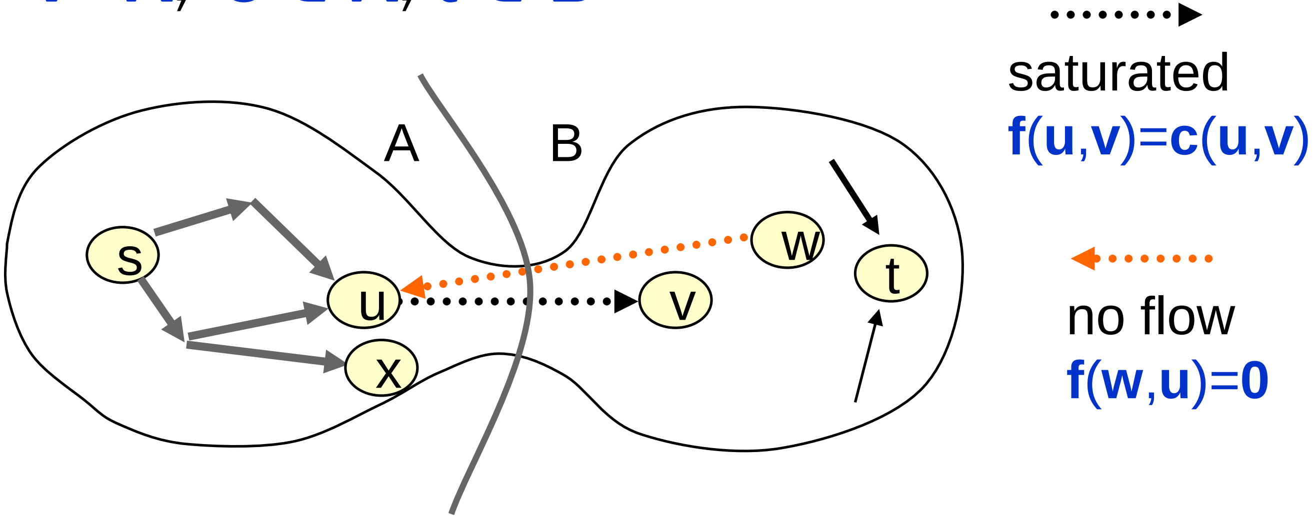
- We know by previous claims that any flow f' satisfies $\nu(f') \le c(A,B)$ and we know that F-F runs for finite time until it finds a flow f satisfying conditions of the theorem Therefore for any flow f', $\nu(f') \le \nu(f)$
- **Corollary:**
  - (1) F-F computes a maximum flow in G
  - (2) For any graph G, the value n(f) of a maximum flow = minimum capacity c(A,B) of any s-t-cut in G

proof of the theorem

Assume: G_f has no aug. path

Let $A = \{ u \mid \exists$ a path in G_f from s to $u \}$

$B = V - A$; $s \in A$, $t \in B$



This is true for **every** edge crossing the cut:

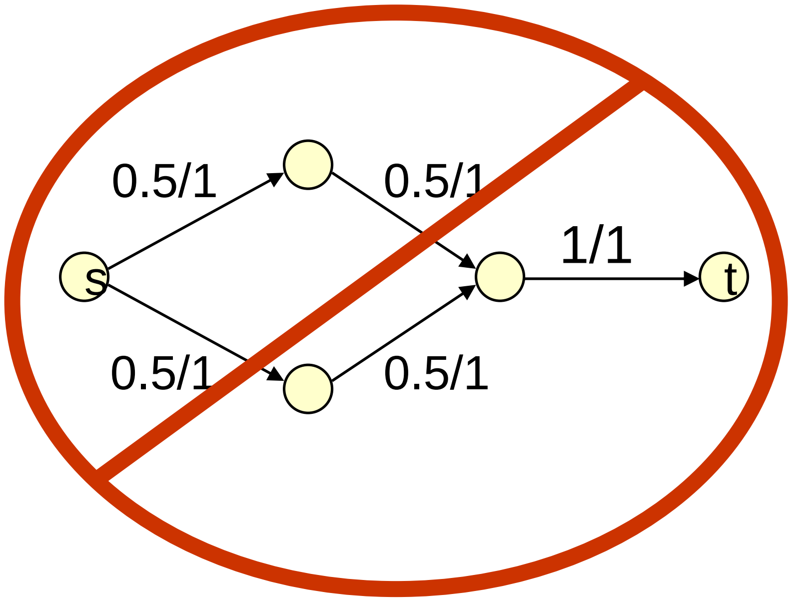
$\nu(f) = f^{out}(A) - f^{in}(A) = c(A,B)$ and $f^{in}(A) =$ 0 hence

$$f^{out}(A) = \sum_{\substack{u \in A \\ v \in B}} f(u,v) = \sum_{\substack{u \in A \\ v \in B}} c(u,v) = c(A,B)$$

flow integrality theorem

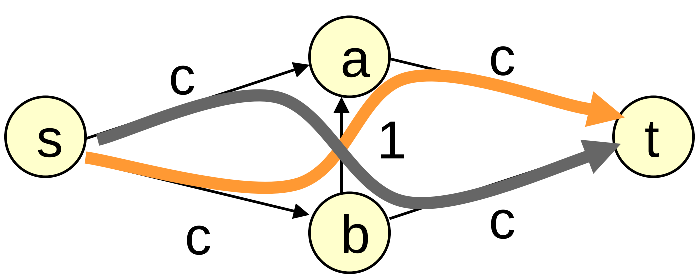
If all capacities are integers

- The max flow has an integer value
- Ford-Fulkerson method finds a max flow in which f(u,v) is an integer for all edges (u,v)



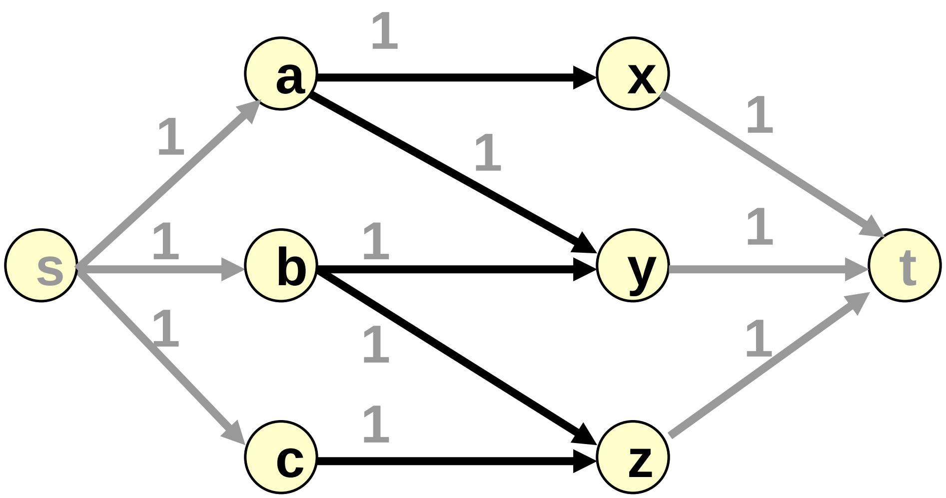
corollaries & facts

- If Ford-Fulkerson terminates, then it has found a max flow.
- It will terminate if **c(e)** integer or rational (but may not if they're irrational).
- However, may take exponential time, even with integer capacities:



$c = 10^9$, say

bipartite matching



Integer flows implies each flow is just a subset of the edges

Therefore flow corresponds to a matching

$O(mC)=O(nm)$ running time

next time: capacity-scaling algorithm

- General idea:
  - Choose augmenting paths **P** with 'large' capacity c_P
  - Can augment flows along a path **P** by any amount $\Delta \leq c_P$

    Ford-Fulkerson still works
  - Get a flow that is maximum for the high-order bits first and then add more bits later