

Algorithms - Spring '25

MSSPs
Intro to Flows

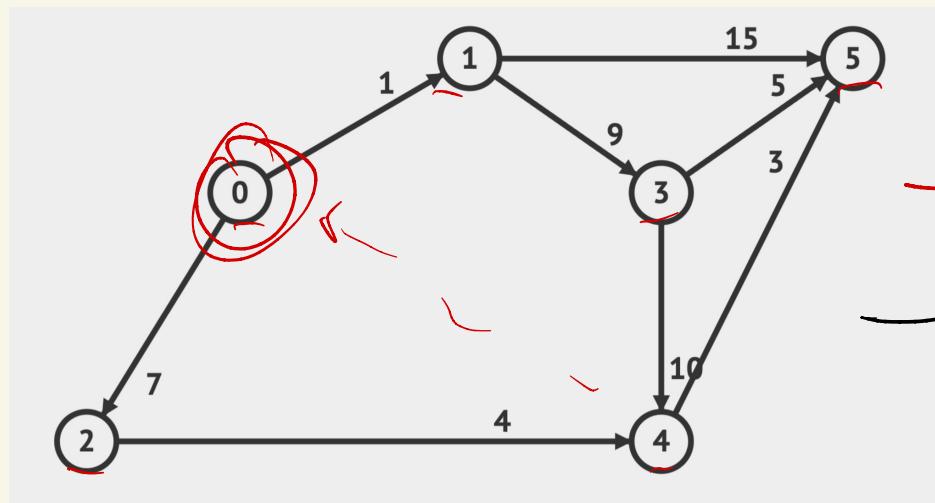


Recap

- Readings & HW next week

All - Multiple Source Shortest paths

For all pairs $u, v \in V$, store $\text{dist}(u, v)$.



Source:

Lookup: $O(1)$

dest:

0	0	1	2	3	4	5
0	0	1	7	10	11	14
1	∞	0	∞	9	19	15
2	∞		0			
3	∞			0		
4	∞				0	
5	8	∞	∞	∞	8	0

MSSP Algorithms

Approaches:

$\forall v$, compute SSSP(v)

Johnson's alg:
reweight

Fisher: Divide &
Conquer

no negatives

$O(V \cdot E \log V)$

\times

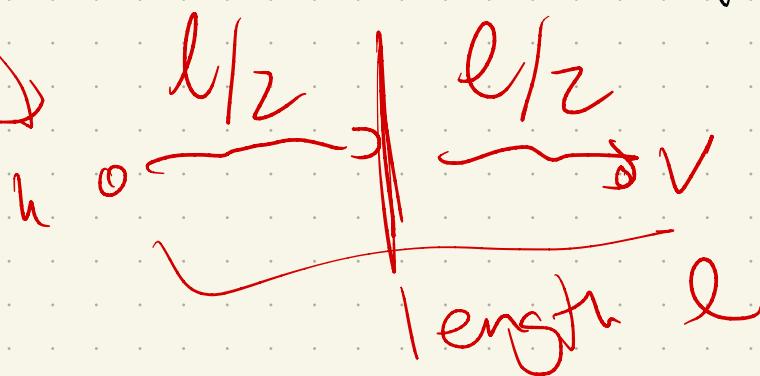
negatives

$O(V^2 E)$

V. (BF)

$O(V^3 \log V)$

$\alpha V^3 \log V$



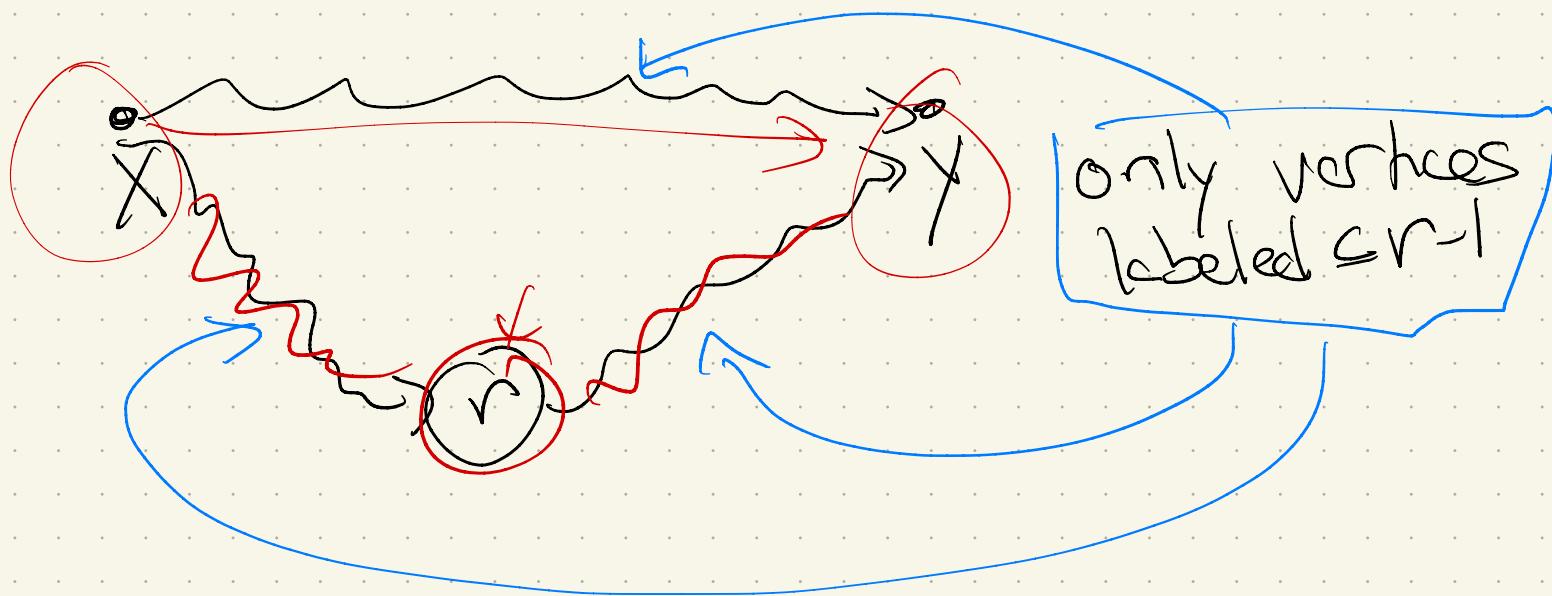
Can we get better?

Floyd-Warshall: instead of path length
order vertices $1, \dots, V$

Let $d(x, y, r) =$

best length path via
vertices labeled $1 \dots r$

Then:



Recursion:

K o S O Y

$$dist(u, v, r) = \begin{cases} w(u \rightarrow v) & \text{if } r = 0 \\ \min \left\{ dist(u, v, r - 1), dist(u, r, r - 1) + dist(r, v, r - 1) \right\} & \text{otherwise} \end{cases}$$

don't use vertex r

So: for $r \leftarrow \underline{1}$ to V
for all $u, v \in G$

update $\text{dist}(u, v, r)$

$$r = \theta$$

1

A hand-drawn diagram on grid paper illustrating a boundary condition for a differential equation. A vertical line segment on the left is labeled 'u'. A horizontal line segment at the top is labeled 'v'. A central rectangular region is bounded by a black line. Red annotations include a circle labeled 'O' at the top right, a bracket labeled 'edge' pointing to the right side of the rectangle, and another circle labeled 'O' with an arrow pointing to the right labeled 'or'.

$\text{dist}(u, v)$)
 ~~$r = 1$~~ : can v_1 be useful?

A hand-drawn diagram on grid paper. It features a large black-outlined rectangle. Inside, there is a smaller red-outlined square. A red arrow originates from the top-left corner of the square and points to its exact center. Red arcs are drawn around the square and the rectangle's corners, suggesting a path or a sequence of moves.

be useful?
 $(v \rightarrow v_i) + w(v_i \rightarrow v)$
better than
of better
prev matrix entry
code

KLEENEAPSP(V, E, w):

```
for all vertices  $u$ 
  for all vertices  $v$ 
     $dist[u, v, 0] \leftarrow w(u \rightarrow v)$  (or  $\infty$  if no edge  $u \rightarrow v$ )
  for  $r \leftarrow 1$  to  $V$ 
    for all vertices  $u$ 
      for all vertices  $v$ 
        if  $dist[u, v, r - 1] < dist[u, r, r - 1] + dist[r, v, r - 1]$ 
           $dist[u, v, r] \leftarrow dist[u, v, r - 1]$ 
        else
           $dist[u, v, r] \leftarrow dist[u, r, r - 1] + dist[r, v, r - 1]$ 
```

Runtime:

$\mathcal{O}(V^3)$

Space:

Save
Space:
don't
keep older
r's, just
Overwrite

FLOYDWARSHALL(V, E, w):

```
for all vertices  $u$ 
  for all vertices  $v$ 
     $dist[u, v] \leftarrow w(u \rightarrow v)$ 
for all vertices  $r$ 
  for all vertices  $u$ 
    for all vertices  $v$ 
      if  $dist[u, v] > dist[u, r] + dist[r, v]$ 
         $dist[u, v] \leftarrow dist[u, r] + dist[r, v]$ 
```

$\mathcal{O}(V^2)$ space

Special cases

Best known published result for
general graphs: $O(V^3)$

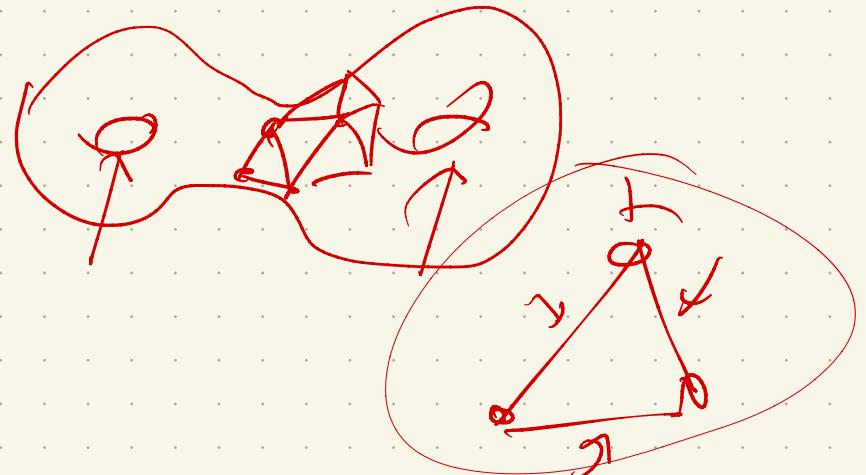
Conjecture: no $\underline{O(V^{3-\epsilon})}$ algorithm

Planar graphs: $O(V \log V)$

Meshes (graphs embedded in 3d):

$$O(g^2 V \log V)$$

genus



Many others...

Ch 10: Flows + Cuts

Motivation:

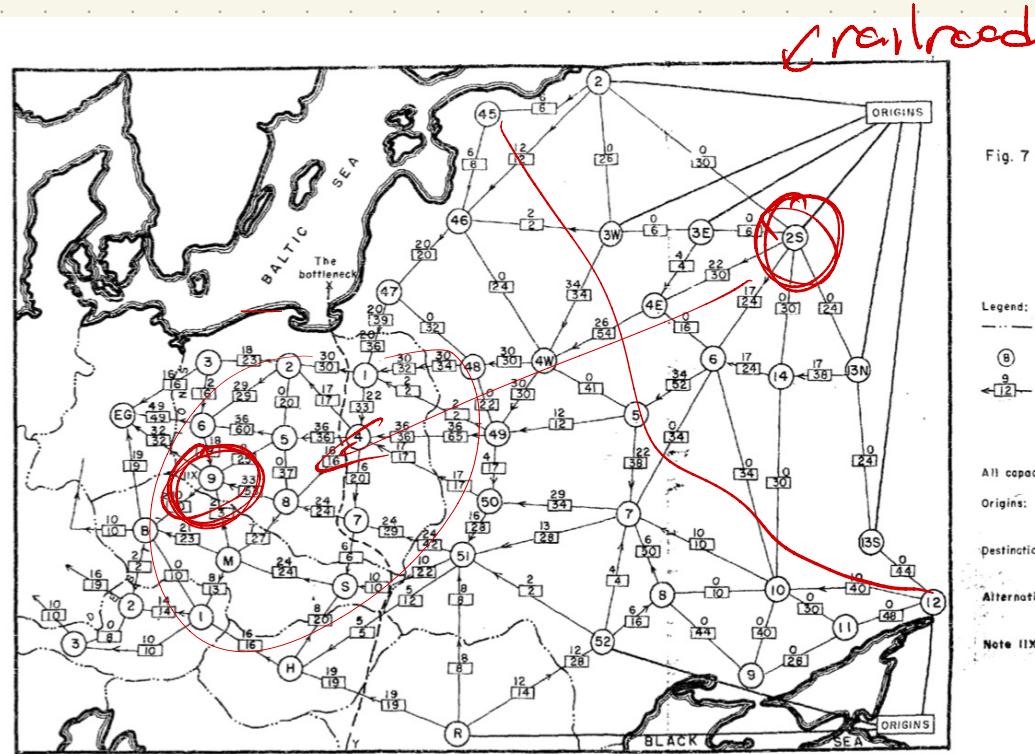


Figure 10.1. Harris and Ross's map of the Warsaw Pact rail network. (See Image Credits at the end of the book.)

Question: How much can I ship from u to v?

Question: How can I separate u from v with "least cost"?

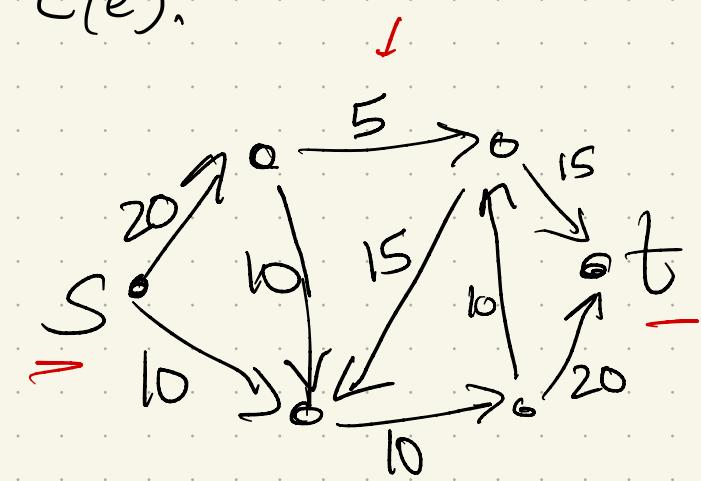
More formally:

Given a directed graph with two designated vertices, s and t .

Each edge is given a capacity $c(e)$.

Assume: - No edges enter s

- No edges leave t
- Every $c(e) \in \mathbb{Z}^+$

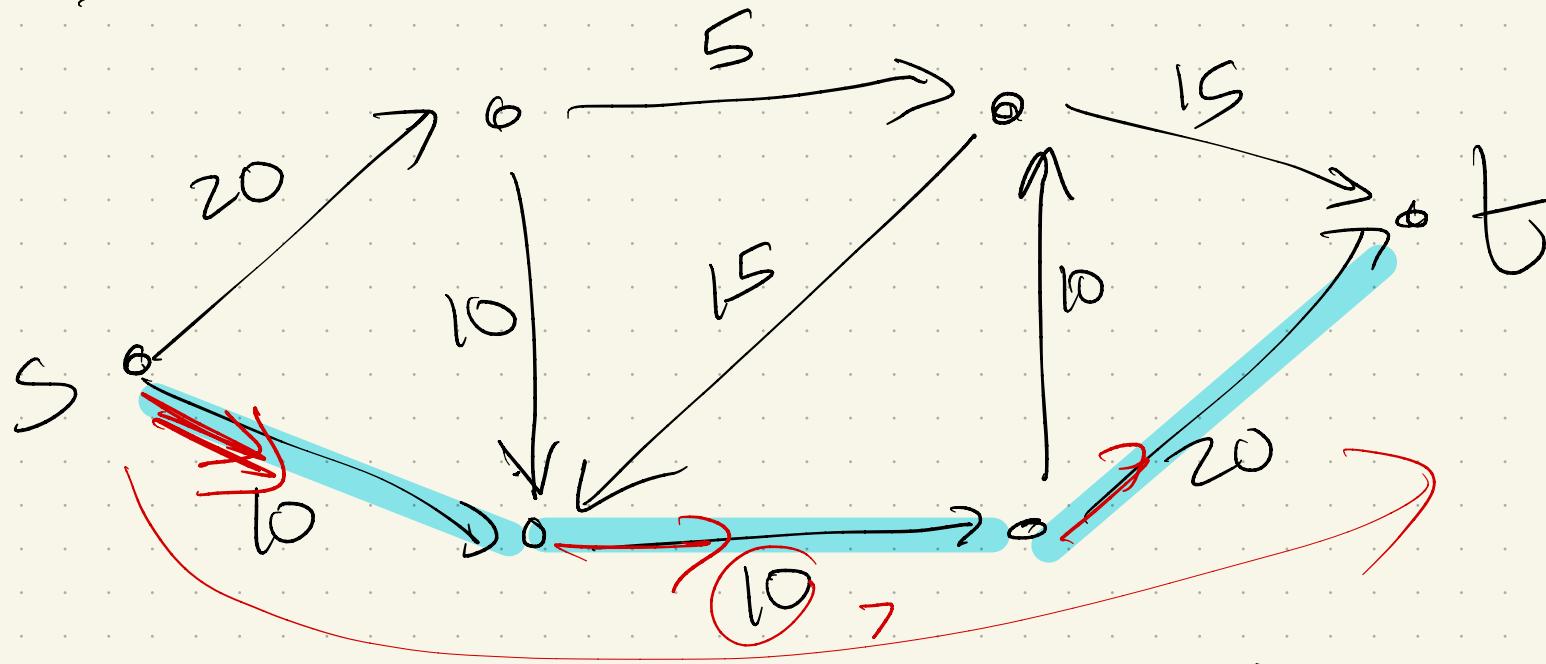


Max flow: Find most I can send from s to t without exceeding edge capacities.

Min cut: find lightest set of edges separating s from t

Aside:

Not path length:



Consider a path s to t :

$$\text{length: } 10 + 10 + 20 = 40$$

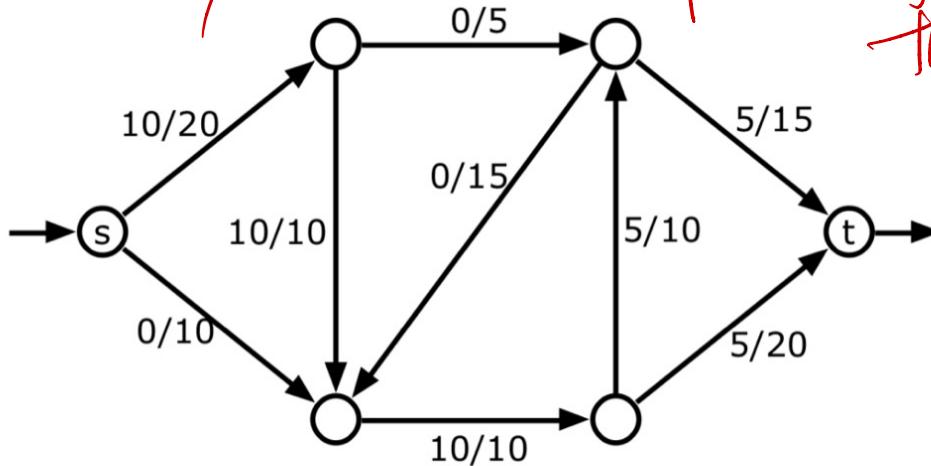
Flow: can send 10 along it

Formalizing flow:

A flow is a function $f: E \rightarrow \mathbb{R}^+$, where $f(e)$ is the amount of flow going over edge e .

Must satisfy 2 things:

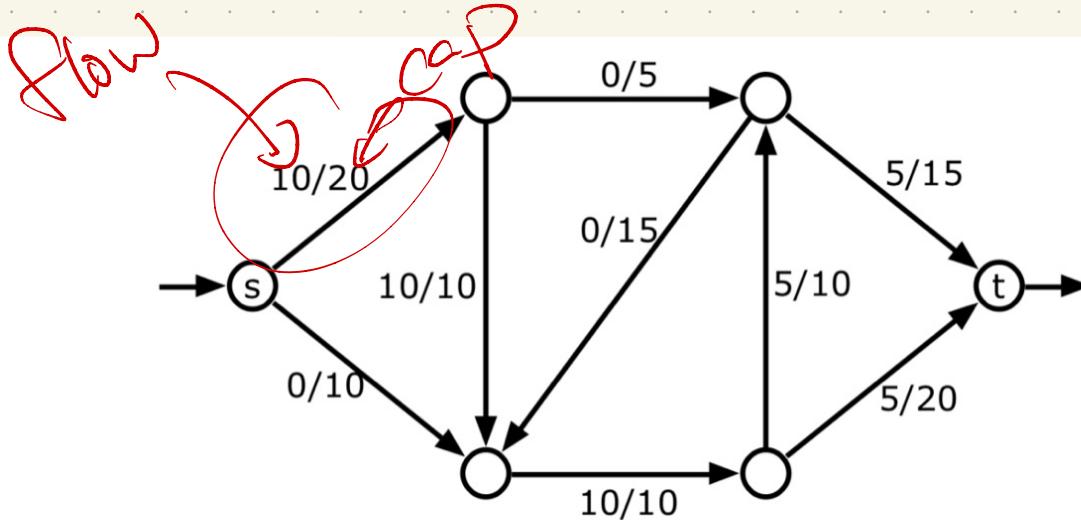
- Edge constraints: $0 \leq f(e) \leq c(e)$
(Don't overflow edge)
- Vertex constraints:
 $\sum_{v \neq s, t} f(v) = \sum_{e \text{ out of } s}$ Flow in to v
 $\geq \sum_{e \text{ into } t}$ Flow out
only s can ship out, & no other vertex
then t can store



An (s, t) -flow with value 10. Each edge is labeled with its flow/capacity.

$$\begin{aligned}\text{Value}(f) &= \sum_{e \text{ out of } s} f(e) \\ &= \sum_{e \text{ into } t} f(e)\end{aligned}$$

Note on notation & conventions:



An (s, t) -flow with value 10. Each edge is labeled with its flow/capacity.

A flow is a function on edges!
(so are capacities)

Here: $f(e)$ in figures
 $c(e)$
edge constraints $\Rightarrow 0 \leq f_e \leq c_e$

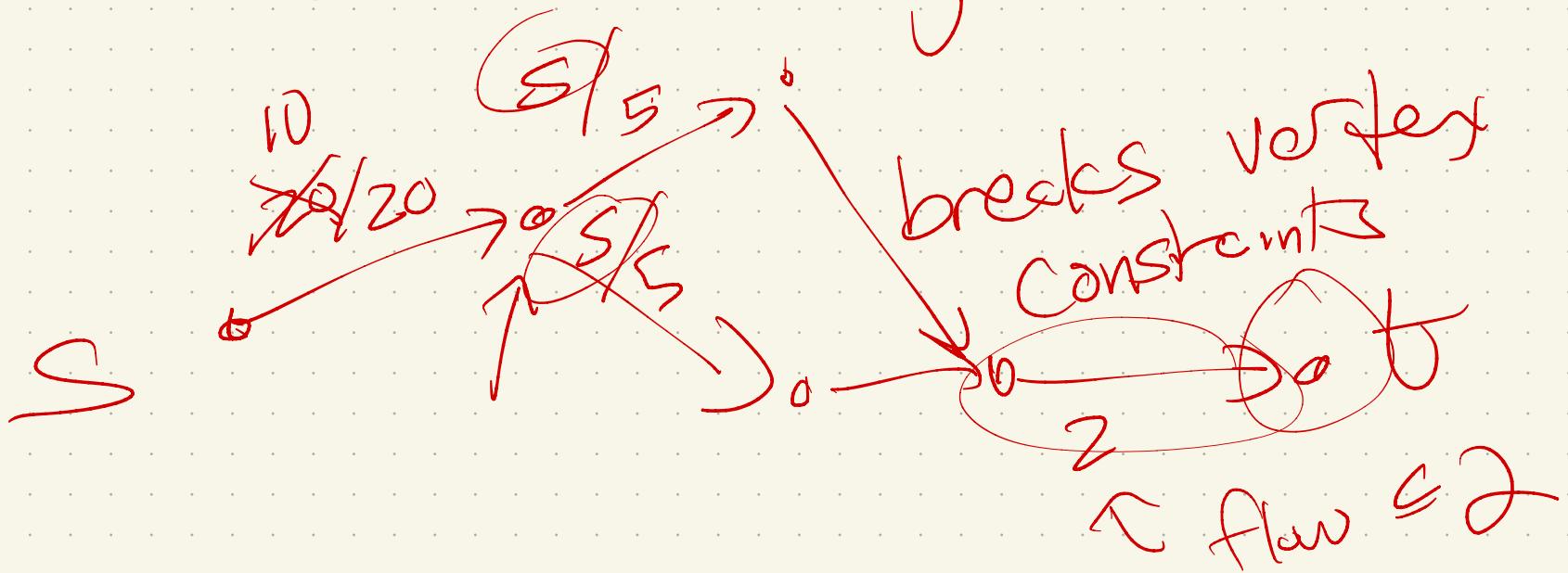
Possible Flows:

$$f(e) = 0 \quad \forall e \in E \quad \text{←}$$

"0-Flow"

Not: Set $f(e) = c(e)$

On all edges

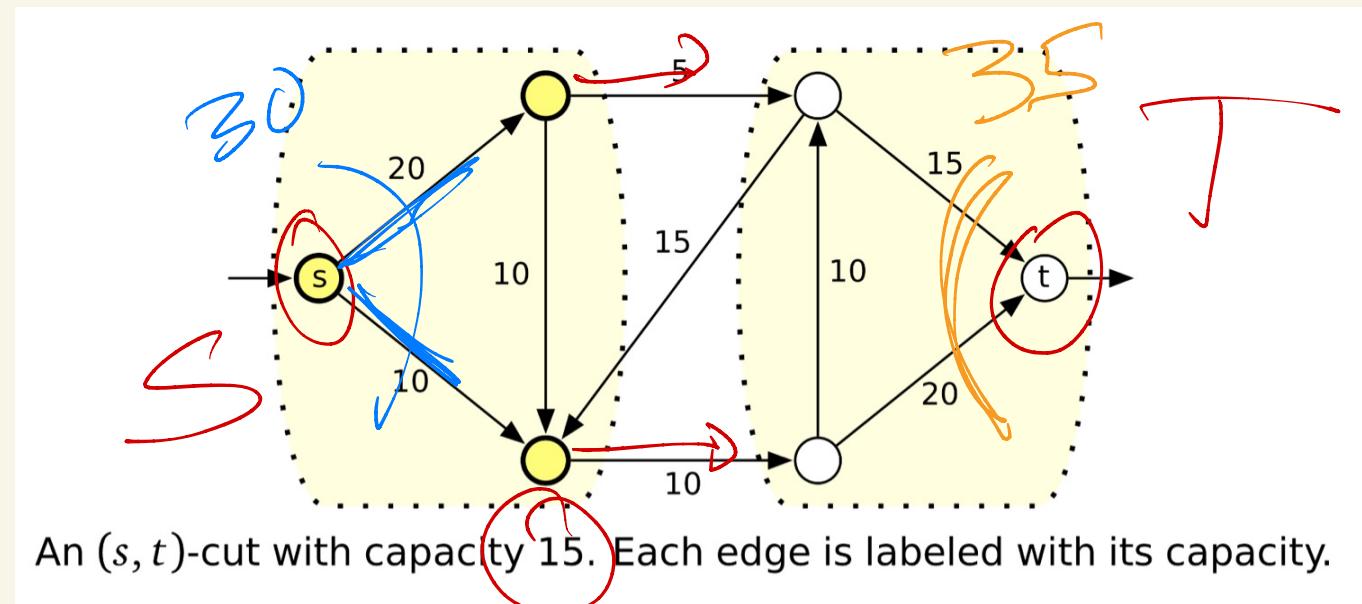


Formalizing Cuts

An s-t cut is a partition of the vertices into 2 sets, S and T , so that

- $s \in S$
- $t \in T$
- $S \cap T = \emptyset$,

$$S \cup T = V$$



S, T is a partition of V

The capacity of a cut is $\sum_{\substack{uv \in E \\ u \in S, v \in T}} c(\vec{uv})$

Min Cuts: not always so obvious!

There are many

S-T cuts.

Finding any cut? **easy**

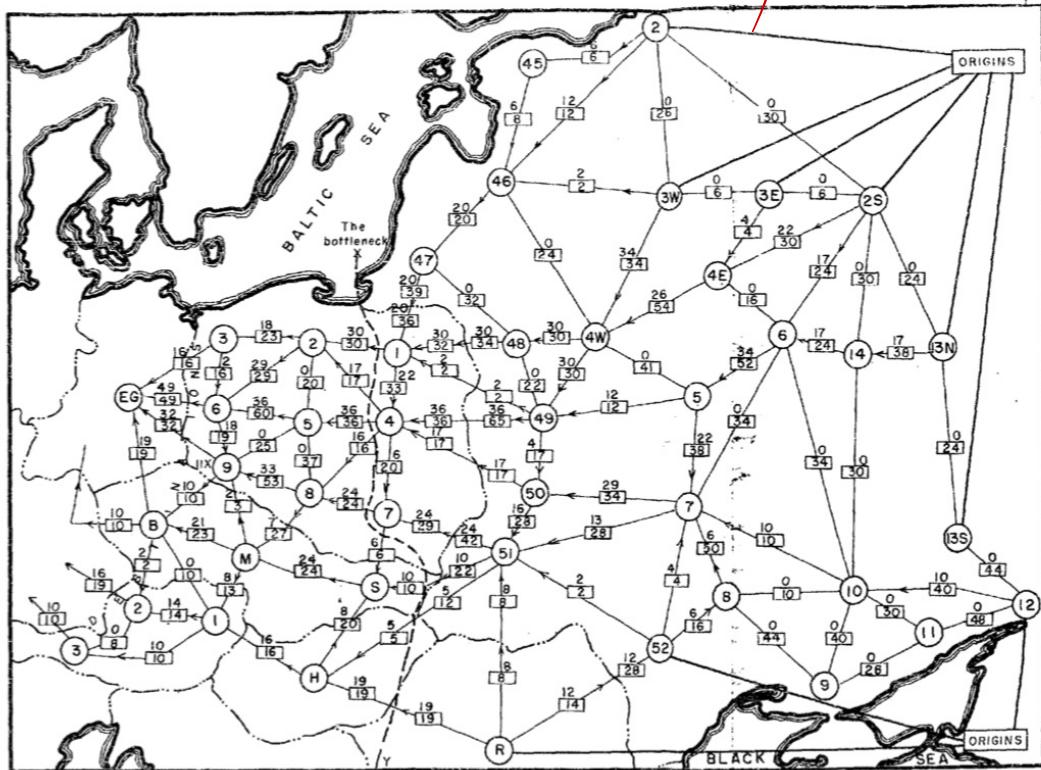


Figure 10.1. Harris and Ross's ~~map~~ of the Warsaw Pact rail network. (See Image Credits at the end of the book.)

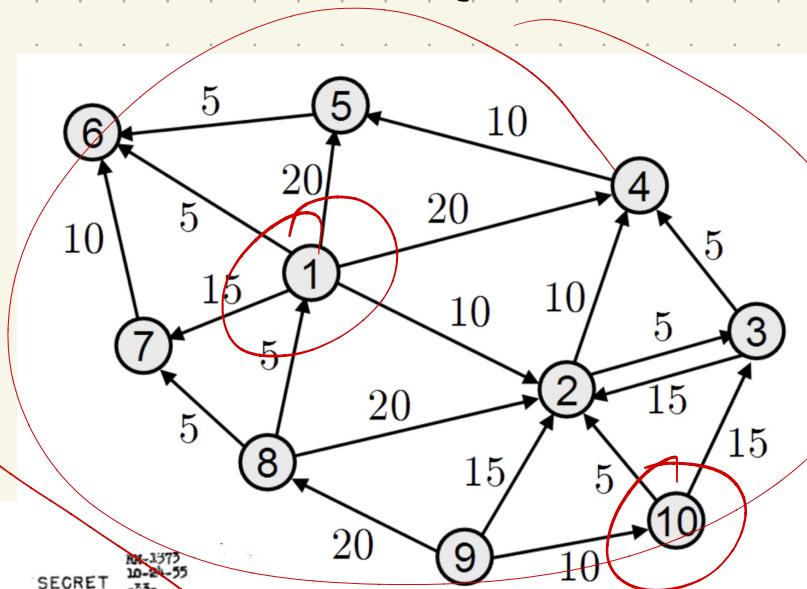


Fig. 7 — Traffic pattern: entire network available

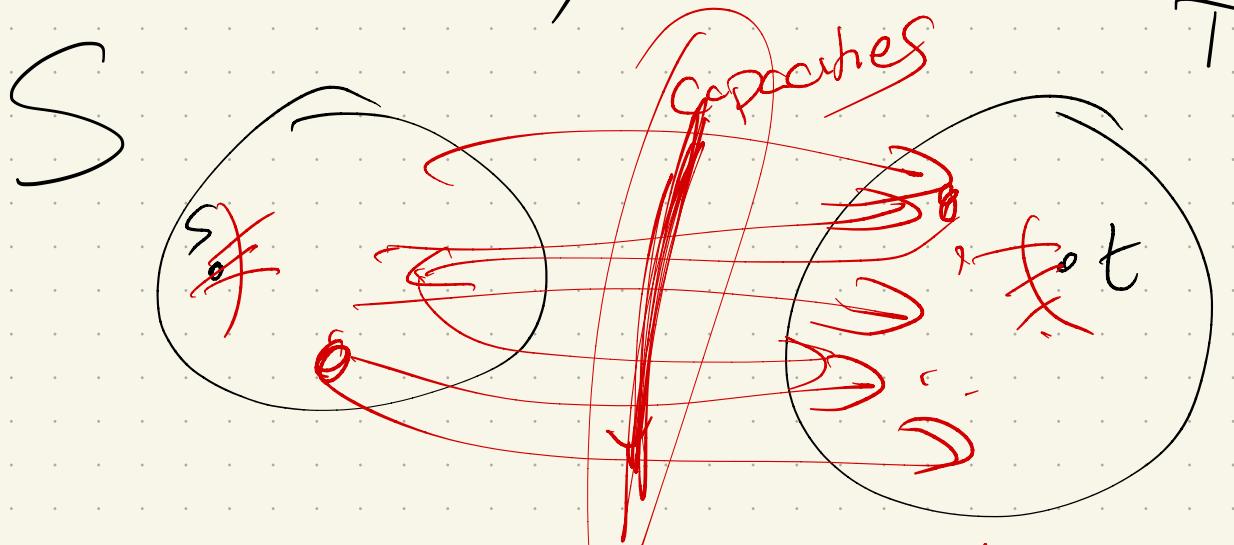
Legend:
— International boundary
⑧ Railway operating division
← [] Capacity: 12 each way per day.
Required flow of 9 per day toward
destinations (in direction of arrow)
with equivalent number of returning
trains in opposite direction
All capacities in { trains } each way per day
Origins: Divisions 2, 3W, 3E, 2S, 13N, 13S,
12, 52(USSR), and Roumania
Destinations: Divisions 3, 6, 9 (Poland);
B (Czechoslovakia); and 2, 3 (Austria)
Alternative destinations: Germany or East
Germany
Note IX of Division 9, Poland

V
J
S
t
V-2
2 cuts

Intuitively, these are connected:

Consider any cut:

$$\text{any flow} \leq \text{Cut}$$



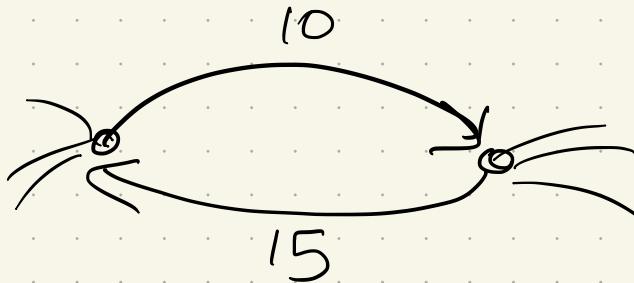
Cap of cut: edges from S to T

Any flow must use these edges

Amount of flow can't exceed
Cap. of edges in cut.

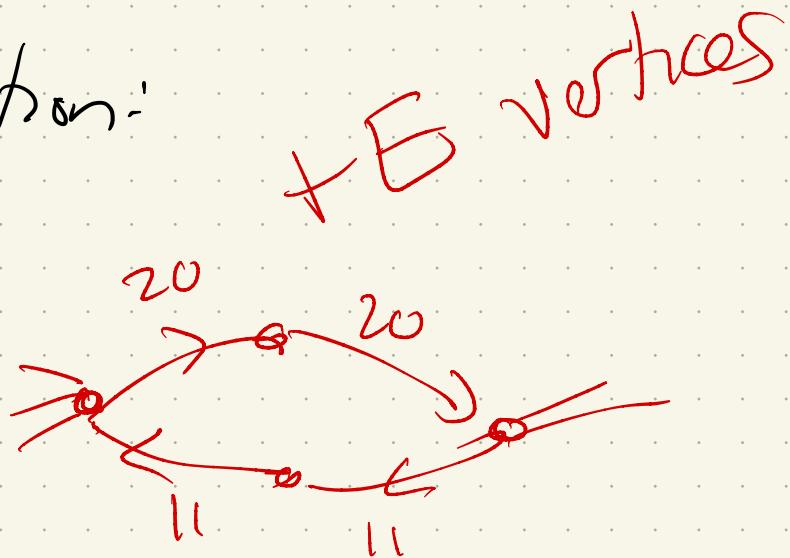
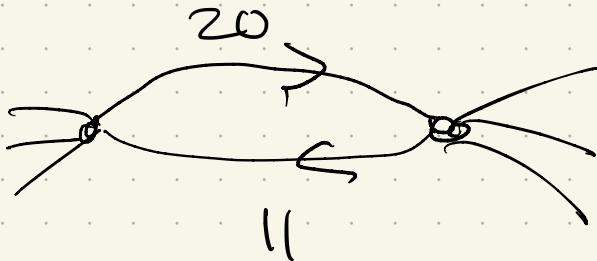
Note: We'll assume every pair of vertices has at most one edge.

So no:



Why? - Makes calculations easier!
(Stay tuned for why...)

How? Simple transformation:



Thm: (Ford - Fulkerson '54, Elias-Fernstein-Shannon '56)

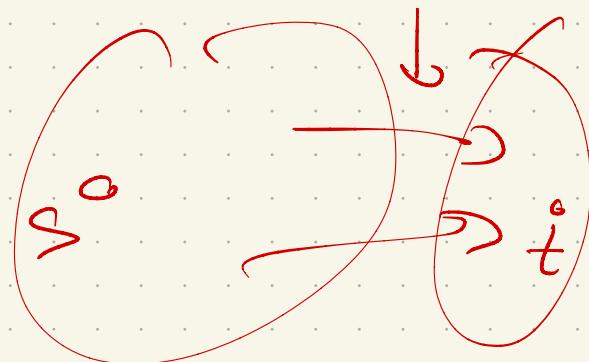
The max flow value

$$\xrightarrow{\quad} = \min \text{cut value}$$

Wow!

One way is easy:

Any flow \leq any cut.



Why?

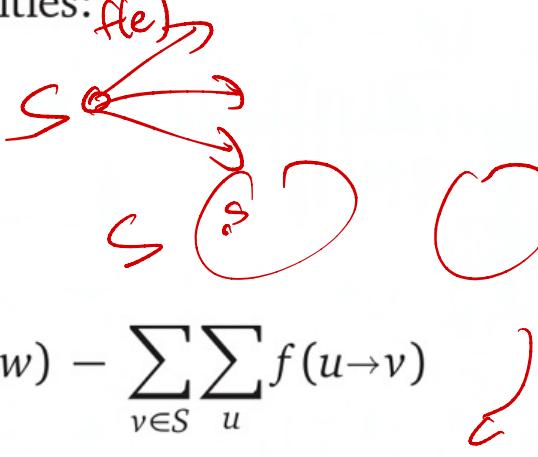
Can exceed edges out
of S \rightarrow into T

More formally:

any flow \leq any cut

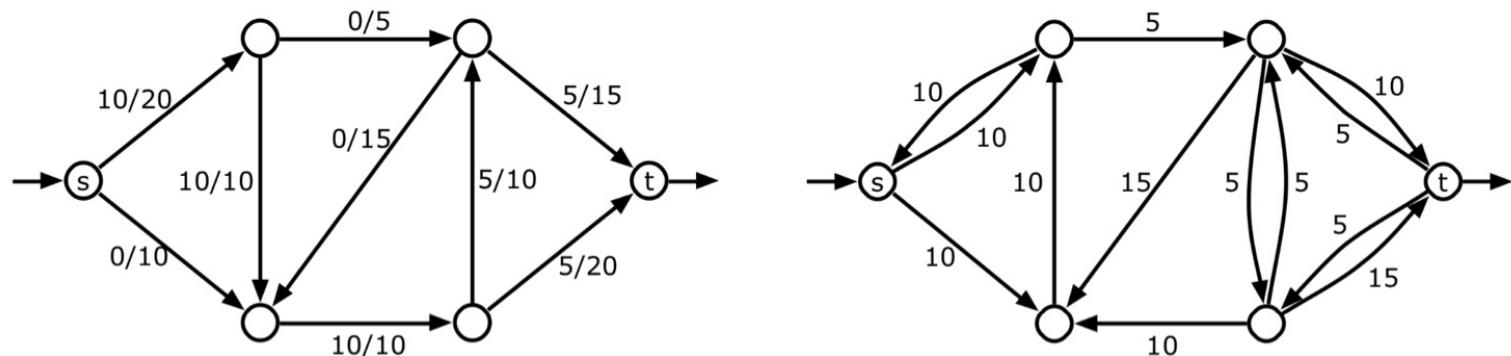
Proof: Choose your favorite flow f and your favorite cut (S, T) , and then follow the bouncing inequalities:

$$\begin{aligned}
 |f| &= \partial f(s) && [\text{by definition}] \\
 &= \sum_{v \in S} \partial f(v) && [\text{conservation constraint}] \\
 &= \sum_{v \in S} \sum_w f(v \rightarrow w) - \sum_{v \in S} \sum_u f(u \rightarrow v) && [\text{math, definition of } \partial] \\
 &= \sum_{v \in S} \sum_{w \notin S} f(v \rightarrow w) - \sum_{v \in S} \sum_{u \notin S} f(u \rightarrow v) && [\text{removing edges from } S \text{ to } S] \\
 &= \sum_{v \in S} \sum_{w \in T} f(v \rightarrow w) - \sum_{v \in S} \sum_{u \in T} f(u \rightarrow v) && [\text{definition of cut}] \\
 &\leq \sum_{v \in S} \sum_{w \in T} f(v \rightarrow w) && [\text{because } f(u \rightarrow v) \geq 0] \\
 &\leq \sum_{v \in S} \sum_{w \in T} c(v \rightarrow w) && [\text{because } f(v \rightarrow w) \leq c(v \rightarrow w)] \\
 &= \|S, T\| && [\text{by definition}]
 \end{aligned}$$



Key tool in proof:

Residual network G_f :

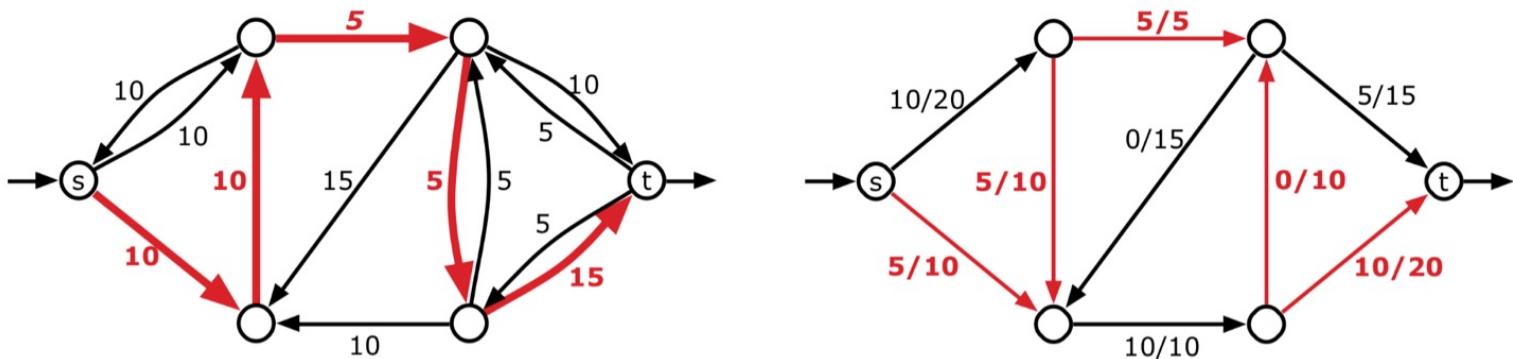


A flow f in a weighted graph G and the corresponding residual graph G_f .

Intuitively: Shows how much more
(or less) flow can be pushed
through an edge.

$$\frac{s/c}{\Rightarrow}$$

Augmenting a path:



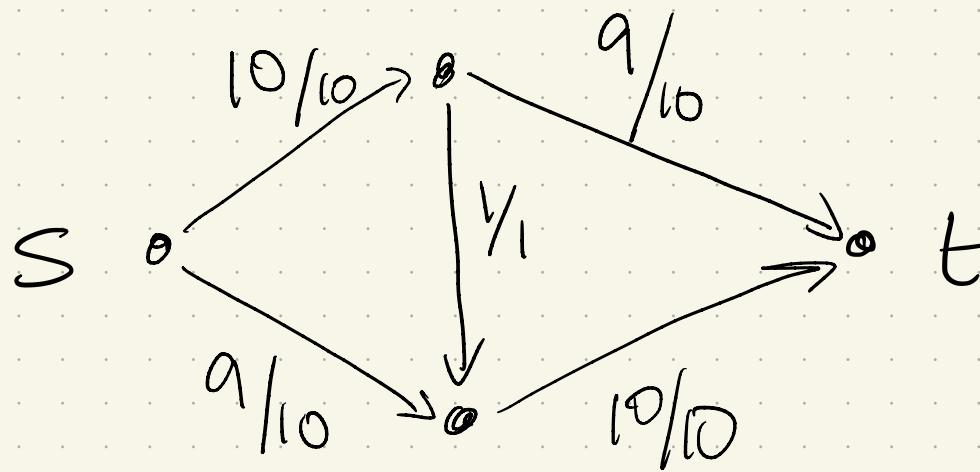
An augmenting path in G_f with value $F = 5$ and the augmented flow f' .

This is just an s - t path in G_f .

Then, find min capacity edge on that path

Claim: I can build a new flow whose value is bigger than f 's

Why can't we just be greedy?



Can get "stuck" if we choose wrong initially:

Are there any more flow paths?

Next week:

an algorithm to find max
flows

→ which will prove the FF theorem
along the way.