In-Class Exercise 10

CSCI 432

28 October 2019

Group Number:
Group members present today:

Max Flow / Min Cut

Input: We are given an acyclic graph $G = (V, E)$ with a weight function on the edges $c: E \to \mathbb{R}$. The weights are all non-negative and will be referred to as *capacities* of the edges.

Goal: To find a flow of maximum value in the graph. A *flow* is an assignment of weights to edges $\omega: E \to \mathbb{R}$ such that $0 \leq \omega(e) \leq c(e)$ for all $e \in E$, and for all vertices $v \neq s, t$, flow in is equal to flow out:

$$\sum_{(x,v)\in E} \omega(x, v) = \sum_{(v,x)\in E} \omega(v, x)$$

The *value of a flow* is equal to the flow out of s (and hence the flow into t): $val(\omega) = \sum_{(s,x)\in E} \omega(s, x)$.

The Ford-Fulkerson Algorithm is one algorithm that computes the max flow. Examime the pseudocode and answer the questions below.

Input: $G = (V, E, c)$
Output: value of the maximum flow

Initialize weighted residual graph $R = (V' = V, E' = E, r = c)$
for $e \in E$ **do**
  Add edge e^{-1} to E' with weight 0
end for
$flow \leftarrow 0$
while $\exists$ path p from s to t using positive edges of R **do**
  $\delta \leftarrow \min_{e\in p} r(e)$
  for $e \in p$ **do**
    $r(e) \leftarrow r(e) - \delta$
    $r(e^{-1}) \leftarrow r(e^{-1}) + \delta$
  end for
  $flow \leftarrow flow + \delta$
end while

---

recall from Monday

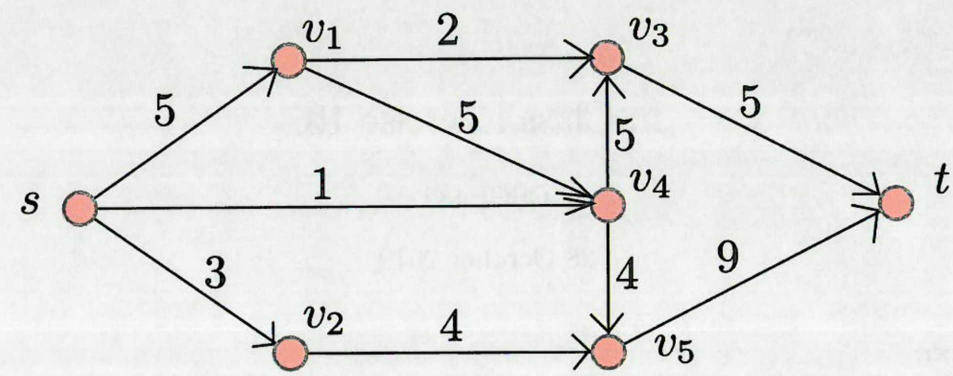
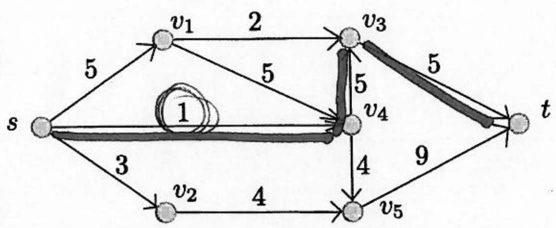


Figure 1: A graph. We wish to find the maximum flow from s to t.

1. Walk though the algorithm for Figure 1.
2. How can the algorithm be updated in order to return the actual flow ω in addition to the value of ω?
3. The number of times that the while loop executes is not unique. Give an example of a graph and two different sequences of paths selected that result in the while loop executing a different number of times.
4. How do we know that the while loop terminates?
5. What is the loop invariant of the while loop?

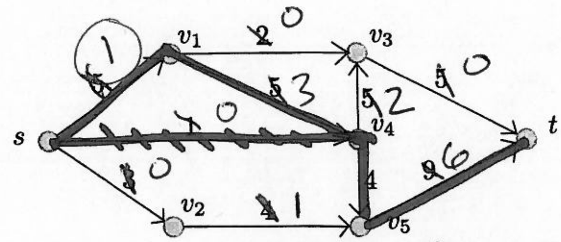
NOT USING RESIDUAL GRAPH

★ This is not the right way to calc. max flow ★

① 

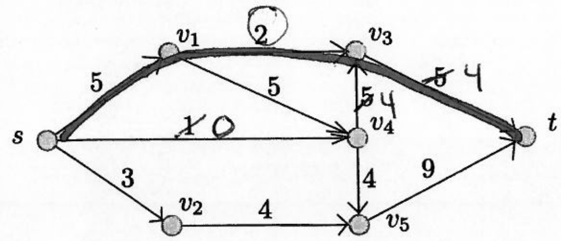
Flow

flow = 1



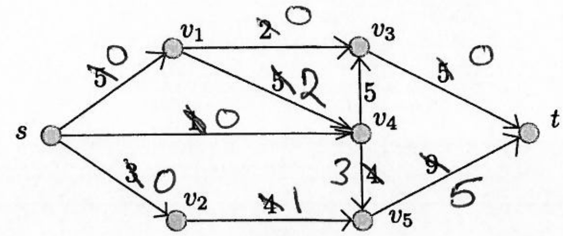
~~Residuals~~ flow = 9

5

② 

Flow

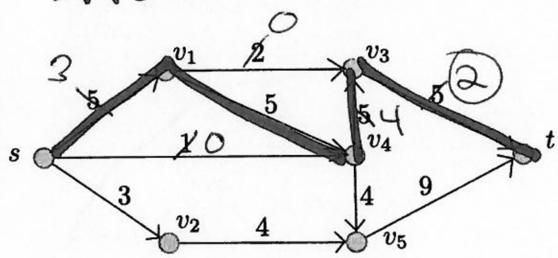
flow = 3



~~Residuals~~

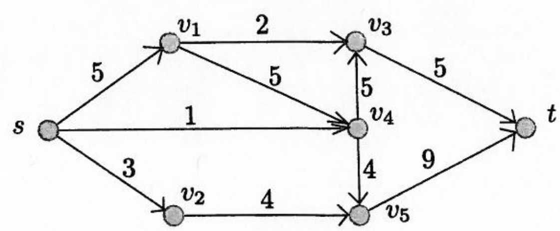
6

NO paths left!

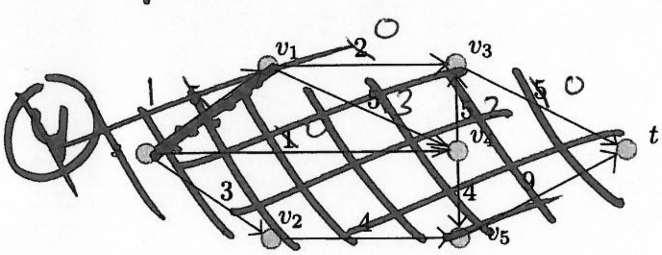
③ 

Flow

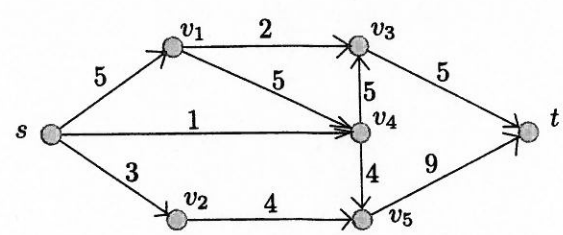
flow = 5



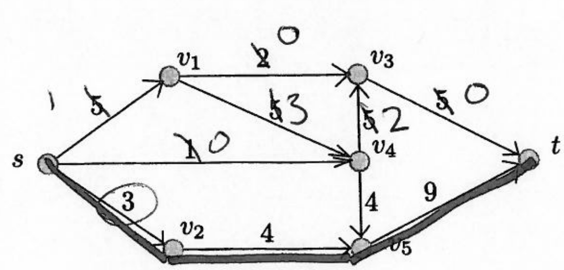
~~Residuals~~

④ 

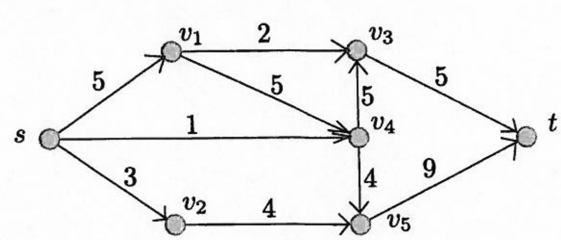
Flow



~~Residuals~~

④ 

Flow

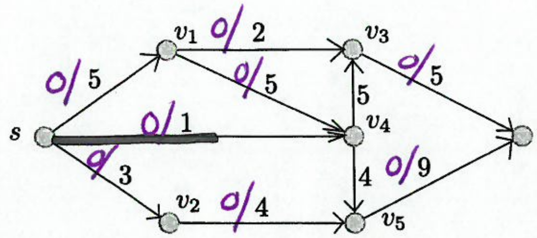


~~Residuals~~

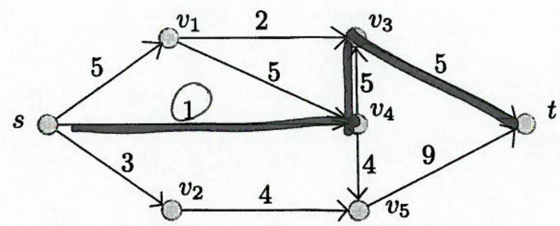
flow = 8

This is the max flow, but...

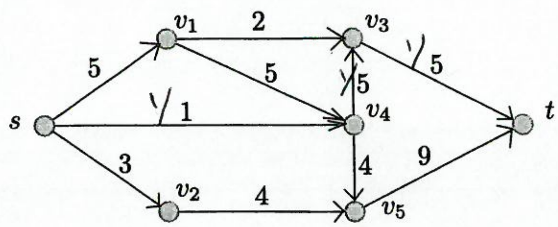
INIT



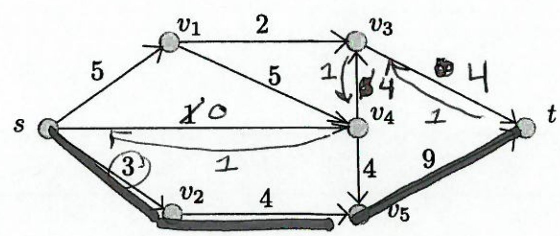
Flow = 0



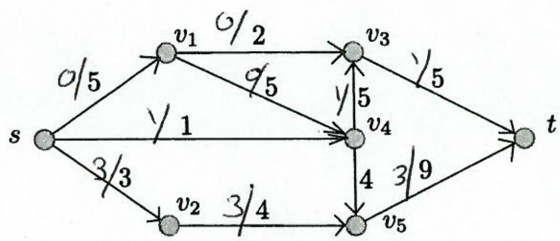
Residuals



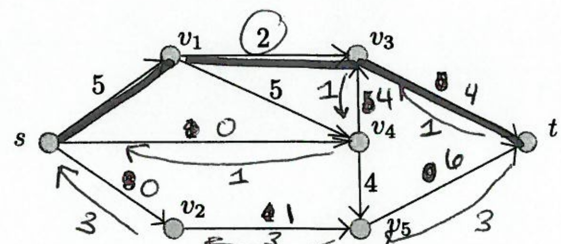
Flow = 1



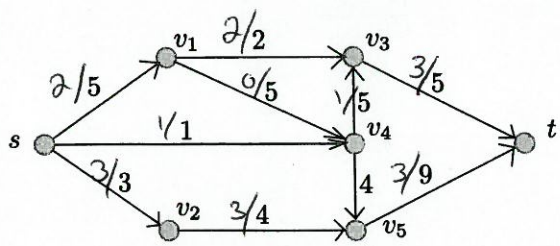
Residuals



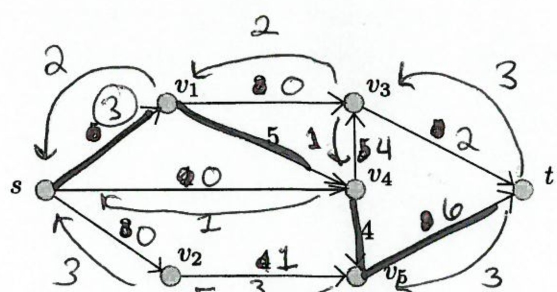
Flow = 4



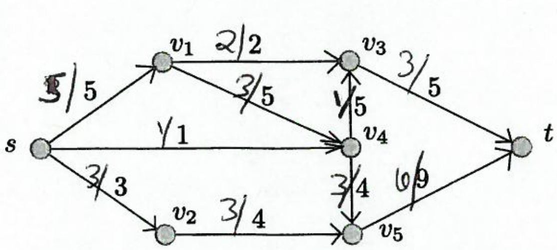
Residuals



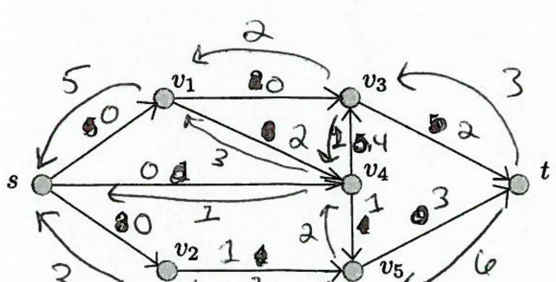
Flow = 6



Residuals

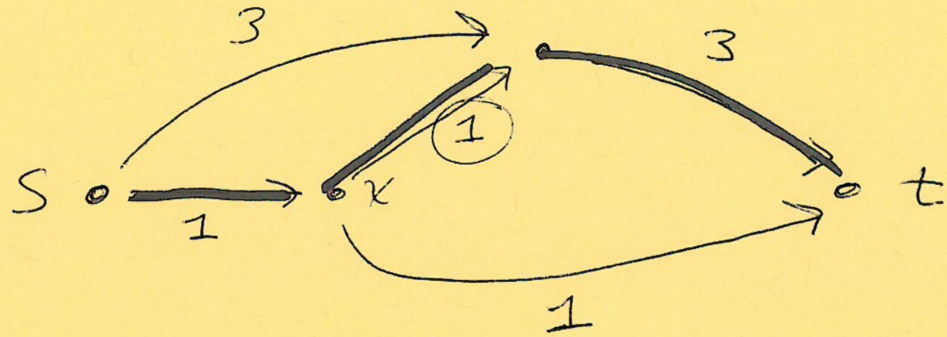


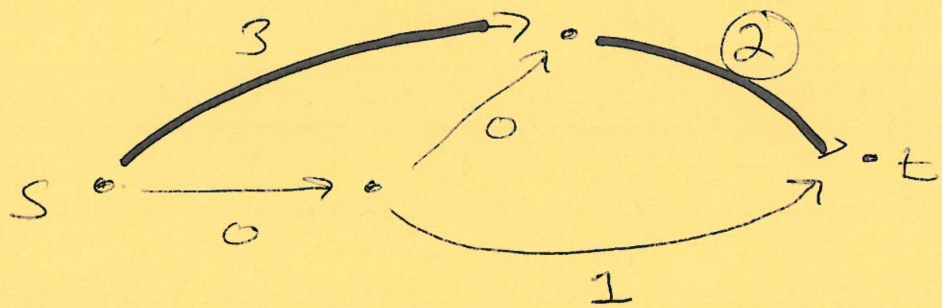
Flow



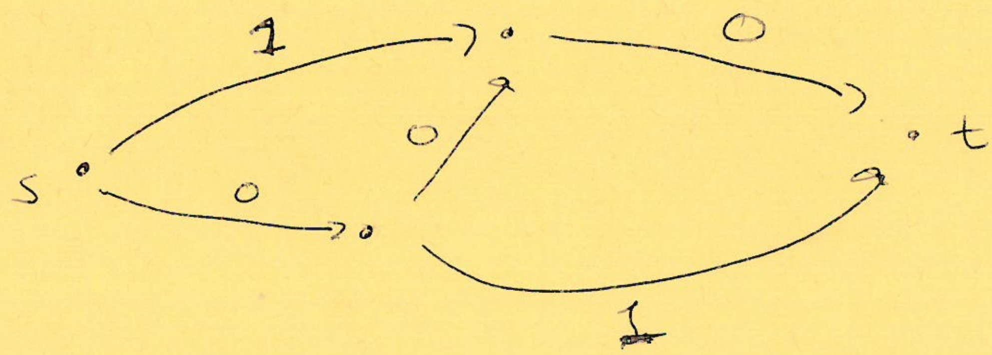
Residuals

No path left b/c no exit from s.





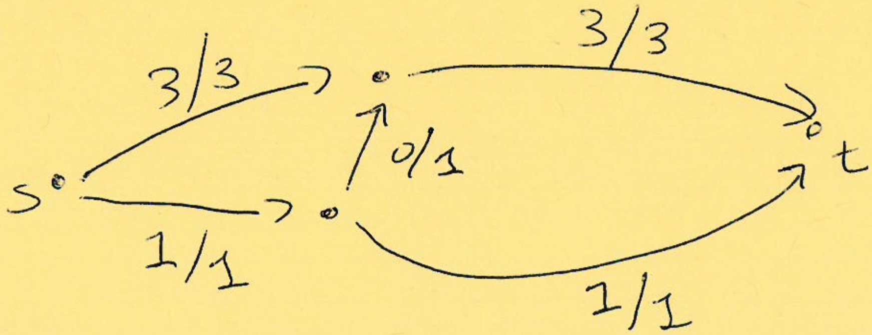
flow = 1

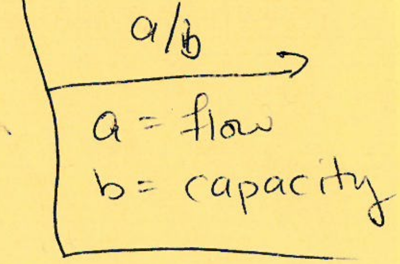


flow = 3

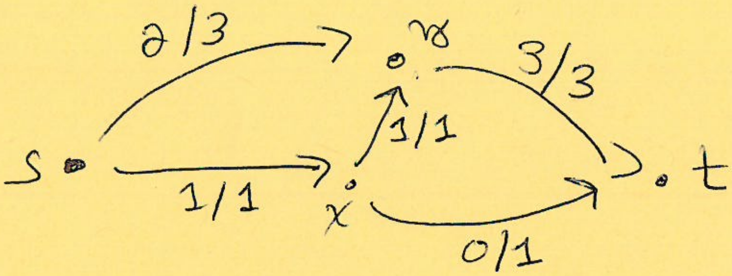
BUT...

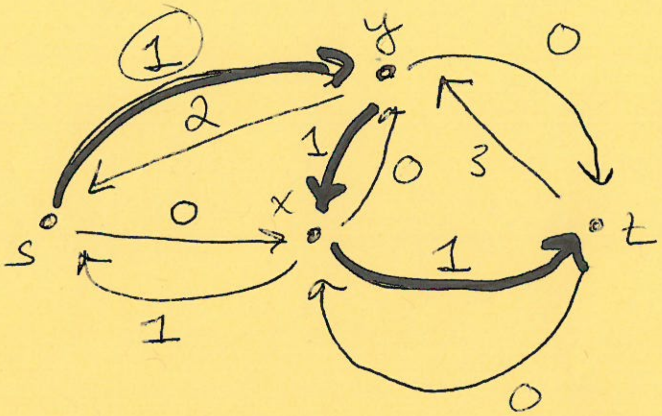
Consider this flow.





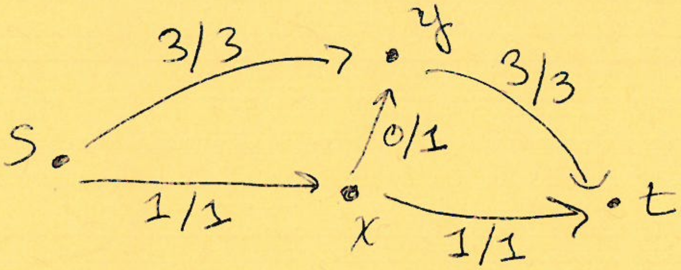
Where we got stuck before:

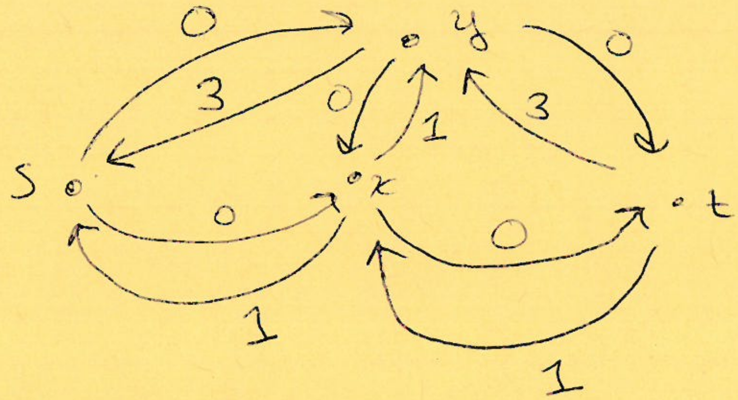




flow = 3

Residual Graphs





NO more paths from s to t.

flow network $G = (V, E, c)$

E: directed

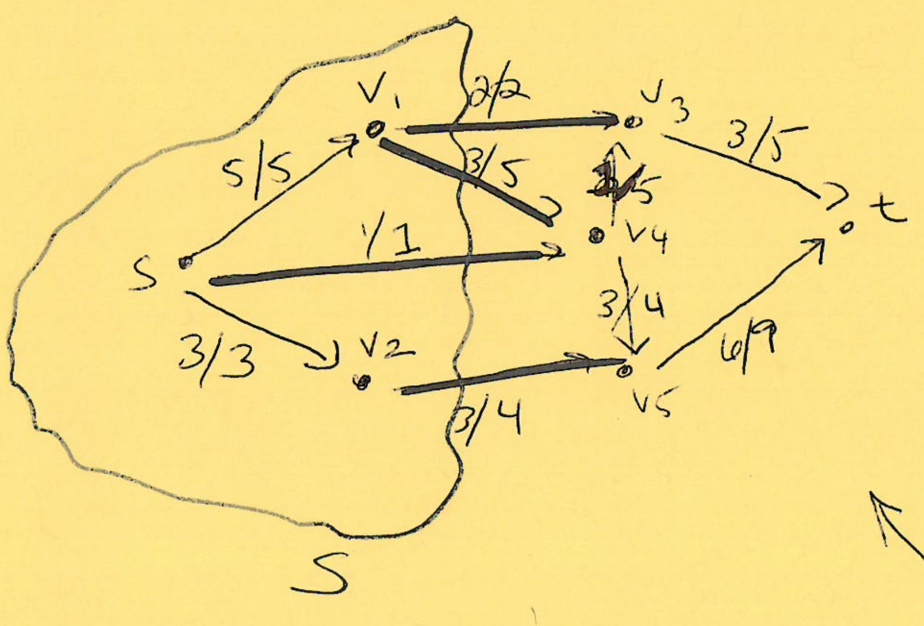
c: capacity fcn $c: E \to \mathbb{R}_{\geq 0}$

A <u>cut</u> is a partition of the vertices

$$V = S \sqcup T$$

(often written: (S,T) is a cut.)

such that: $s \in S$ and $t \in T$.



$S = \{s, v_1, v_2\}$

$T = \{t, v_3, v_4, v_5\}$

this cut has capacity

$2+5+1+4 = 12$

& flow

$2+3+1+3-0 = 9$

<u>def</u> capacity of cut:

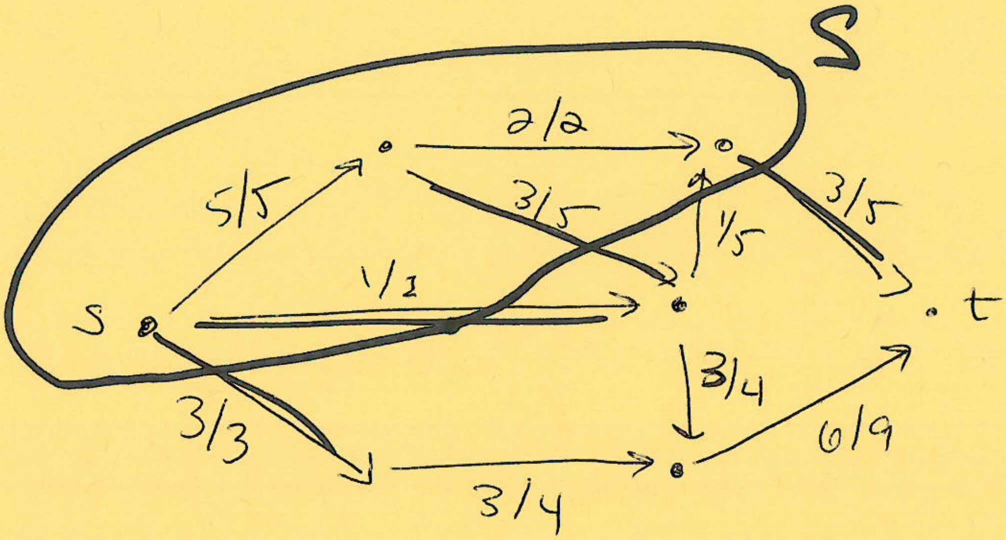
$$c(S,T) = \sum_{u \in S} \sum_{v \in T} c(u,v)$$

<u>def</u> flow of cut: Let $f: E \to \mathbb{R}$ be a flow.

$$f(S,T) = \underbrace{\sum_{u \in S} \sum_{v \in T} f(u,v)}_{\text{from } S \text{ to } T} - \underbrace{\sum_{u \in S} \sum_{v \in T} f(v,u)}_{\text{from } T \text{ to } S}$$

interesting... we got 9 when that was the ~~value~~ value of the flow.

Let's look at another cut



$$c(S,T) = 5 + 5 + 1 + 3 = 14$$

$$f(S,T) = 3 + 3 + 1 + 3 - 1 = 9$$

$\boxed{\text{Lemma}}$ f is a flow in flow graph $G = (V, E, c)$ and (S, T) is a cut.

Then, $f(S,T) = |f|$.

<u>Proof</u>: uses conservation of flow

see the textbook.