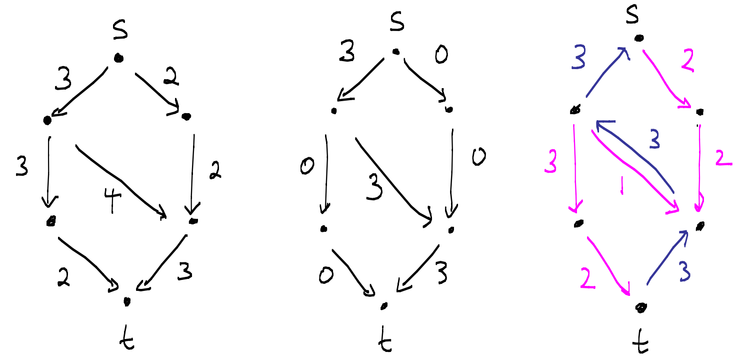
lecture 11

Network Flow 2

- max flow = min cut
- bipartite graphs and maximal matching

Recall from last lecture

Flow Network G, c — Flow f — Residual Graph G_f, c_f



Ford - Fulkerson Algorithm for computing maximum flow

[Note: we haven't yet proven that this algorithm computes max flow]

$f = 0$

$G_f = G$

while there is an s-t path P in G_f {

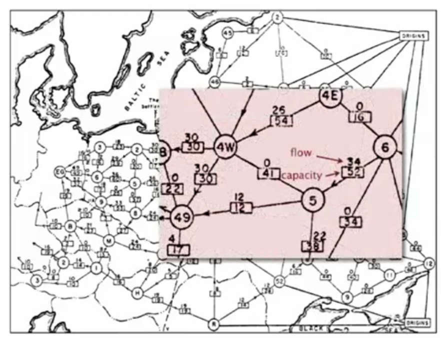
  f. augment (P)

  recompute G_f based on new f

}

Maxflow application (1950s)

Soviet Union goal. Maximize flow of supplies to Eastern Europe.



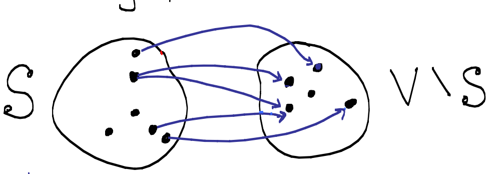
rail network connecting Soviet Union with Eastern European countries
(map declassified by Pentagon in 1999)

http://homepages.cwi.nl/~lex/files/histtrpclean.pdf

Sedgewick: Coursera 2
https://class.coursera.org/algs4partII-002/lecture/22

Recall from Dijkstra etc.

A graph cut is a partition of the graph vertices into two sets.



The "crossing edges" from S to $V \setminus S$ are $\{ (u,v) : u \in S,\ v \in V \setminus S \}$, also sometimes called the cut set.

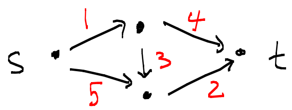
Definition: an s-t cut of a flow network is a cut A, B such that $s \in A$ and $t \in B$.

We sometimes write the cut set as cut (A, B). It is the set of edges from A to B.

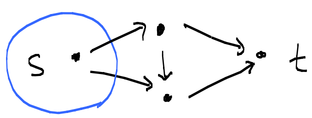
Definition: the capacity of an s-t cut is

$$\sum_{e \text{ in } cut(A,B)} c(e)$$

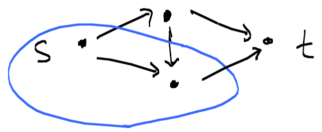
Example: G, c



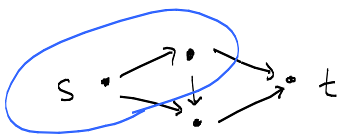
What are the capacities of these cuts?



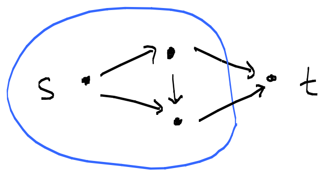
$1+5=6$



$1+2=3$



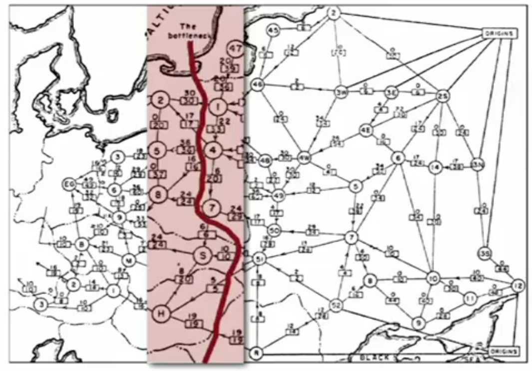
$5+3+4=12$



$4+2=6$

Mincut application (1950s)

"Free world" goal. Cut supplies (if cold war turns into real war).



rail network connecting Soviet Union with Eastern European countries
(map declassified by Pentagon in 1999)

We will show that for any flow network,

the maximum value of a flow

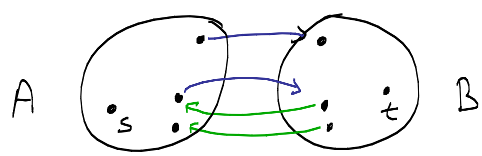
= the minimum capacity of any cut

Moreover, Ford-Fulkerson gives the "max flow" and the "min cut".

Claim: Given a flow network, let f be a flow and let A, B be an st cut. Then,

$$|f| = \sum_{e \text{ in } cut(A,B)} f(e) - \sum_{e \text{ in } cut(B,A)} f(e)$$

notation: $\equiv f^{out}(A) - f^{in}(A)$



Proof: Recall for any u in $V \setminus \{s,t\}$

$$f^{out}(u) = f^{in}(u).$$

Summing over u in $A \setminus s$ gives

$$\sum_{u \text{ in } A\setminus s} f^{out}(u) = \sum_{u \text{ in } A\setminus s} f^{in}(u)$$

Also, $|f| = f^{out}(s)$, $f^{in}(s) = 0$.

Thus,

$$|f| = f^{out}(s) = \sum_{u \text{ in } A} f^{out}(u) - \sum_{u \text{ in } A} f^{in}(u)$$

But each edge $e = (u,v)$ with both u, v in A contributes $f(e)$ to both sums. Removing these pairs from the sums gives

$$|f| = \sum_{e \text{ in } cut(A,B)} f(e) - \sum_{e \text{ in } cut(B,A)} f(e)$$

[recall notation] $\equiv f^{out}(A) - f^{in}(A)$

Claim: For any network flow f, and *any* s-t cut(A,B),

$$|f| \le \sum_{e \text{ in } \mathrm{cut}(A,B)} c(e)$$

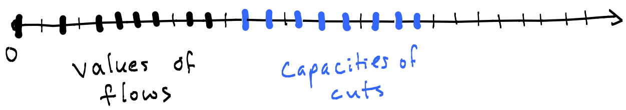
Proof:

$$\begin{aligned}|f| &= f^{out}(A) - f^{in}(A)\\ &\le f^{out}(A)\\ &\le \sum_{e \text{ in } \mathrm{cut}(A,B)} c(e)\end{aligned}$$

Some s-t cuts have greater capacity than others.

Some flows are greater than others.

But every flow must be less than or equal to the capacity of every s-t cut (from last slide)



Thus the value of the maximum flow is less than the capacity of the minimum cut.

Ford Fulkerson terminates when there is no augmenting st path in the residual graph G_f.

Let A be the set of vertices reachable from s in the residual graph. Let $B = V \setminus A$. Then A, B is an st cut in the residual graph.

Hence A, B is an s-t cut in the flow network G too, since G and G_f have the same vertices.

We showed earlier that

$$|f| = f^{out}(A) - f^{in}(A).$$

We now want to show:

$$|f| = \sum_{e \text{ in } \mathrm{cut}(A,B)} c(e)$$

We do so by showing two things:

① $f^{out}(A) = \sum_{e \text{ in } \mathrm{cut}(A,B)} c(e)$ ② $f^{in}(A) = 0$

---

① For any $e = (u,v)$ in $\mathrm{cut}(A,B)$

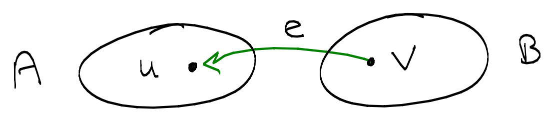
$$f(e) = c(e).$$

Proof: This follows immediately from the definition of A.

If $f(e) < c$, then $e = (u,v)$ would be a forward edge in the residual graph G_f with capacity $c_f = c(e) - f(e) > 0$ and so v in A. (Contradiction since v in B)

---

② $f^{in}(A) = 0$, in particular, for each $e = (v,u)$ in E such that v in B, u in A we have $f(e) = 0$.

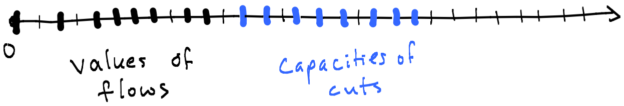


Proof: If $f(e) > 0$, then there would be a backwards edge (u,v) in G_f with residual capacity $c_f(e) = f(e)$ and so v would be reachable from s (contradiction since v in B).

Summary: why does max flow = min cut?

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

$$= \sum_{e \text{ in } cut(A,B)} f(e) - \sum_{e \text{ in } cut(B,A)} f(e)$$



$$\text{Ford Fulkerson flow} = \sum_{e \text{ in } cut(A,B)} c(e) - 0$$

$$= \text{capacity of } cut(A,B)$$

To summarize, Ford-Fulkerson terminates when there is no path from s to t in the residual graph G_f. This defines a cut of the graph G into sets of vertices A and B.

All edges from A to B in E have flow equal to the edge capacity (since otherwise there would be a forward edge in E_f from A to B allowing for more flow from A to B), and all edges in E from B to A have zero flow (since otherwise there would be a backwards edge in E_f from A to B which would give a path in G_f from s to some vertex in B).

Thus, the flow found by Ford-Fulkerson is equal to the capacity of the cut, that is, the capacity of the edges from A to B. But the capacity of every cut from A to B is greater than or equal to the value of any flow. Since Ford-Fulkerson finds a flow that is equal to the capacity of some cut, Ford-Fulkerson finds a maximum flow, which must a minimum cut. Thus, "max flow equals min cut".

(We haven't proved uniqueness here. There may be other flows that have the same maximum values, and other cuts that have the same minimum capacities.)

Possible source of confusion:

$$\left.\begin{matrix} G = (V, E) \\ c, f \end{matrix}\right\} \quad \text{versus} \quad \left\{\begin{matrix} G_f = (V, E_f) \\ c_f \end{matrix}\right.$$

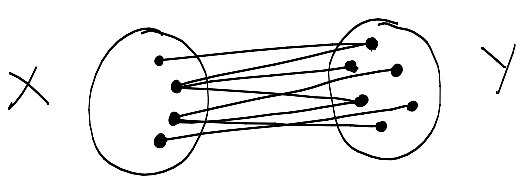
Edges e in summations are in E, not E_f.
The reasoning about the flow $f(e)$ on these edges depends on the edges in E_f.

Q: Given a flow network, how can we compute a minimum cut?

A: Use Ford Fulkerson to compute a maximum flow. (gives G_f).
Run BFS or DFS from s.
The reachable vertices define the set A for the cut.

Bipartite Matching (using network flows)

Suppose we have an undirected bipartite graph $G = (V, E)$.

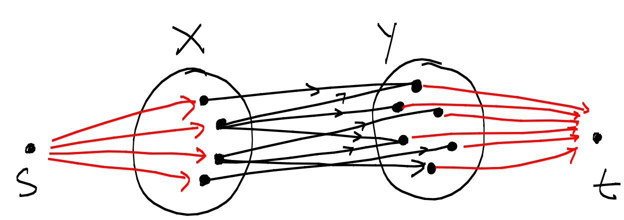


How can we find the maximal matching?
(recall lecture 9)

Define a flow network

$G' = G \cup \{s, t\}$

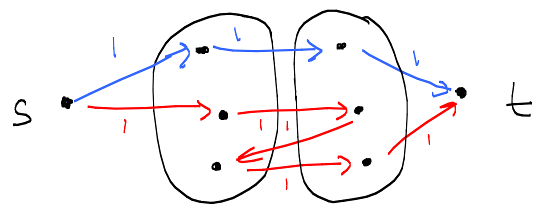
$E' = \{(u,v) \text{ in } E,\ u \text{ in } X,\ v \text{ in } Y\}$
$\cup \{(s,u) : u \text{ in } X\} \cup \{(v,t) : v \text{ in } Y\}$



Let the capacity of each edge be 1.

Ford-Fulkerson will find an augmenting path with $\beta = 1$ on each iteration. These augmenting paths (which are in G_f') are of the form:

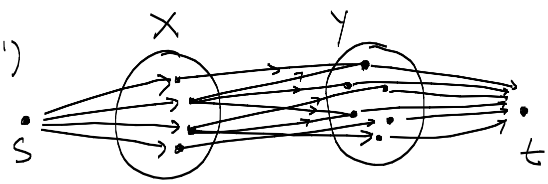
G_f'



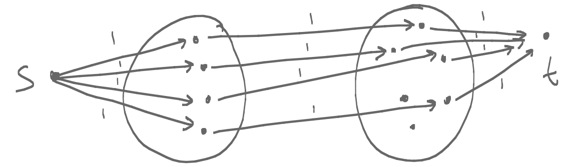
or have more than one zigzag.

$G' = (V', E')$

$c \equiv 1$



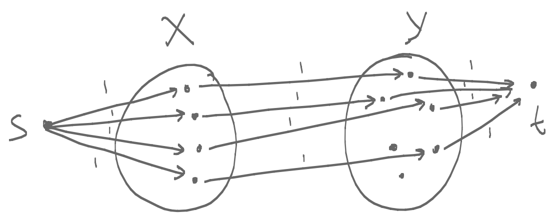
flow f



Note:

- There are no edges from Y to X in E'. The 'back edges' on the previous slide were in E_f.
- Edges with $f(e) = 0$ are not shown.

flow f



Exercise:

The max flow found by Ford-Fulkerson defines a maximal matching in the original graph G, namely the maximal set of edges $e = (u,v)$,

$u \in X$, $v \in Y$ such that $f(e) = 1$.

How long to find maximal matching?

Recall for a general flow network, the Ford Fulkerson takes $O(C\,|E|)$ where $C = \sum_u c(s,u)$.

Suppose* $|X| = |Y| = n$

Then, $C = |X| = n$

assume $> n$

and $|E'| = |E| + 2n = m + 2n$

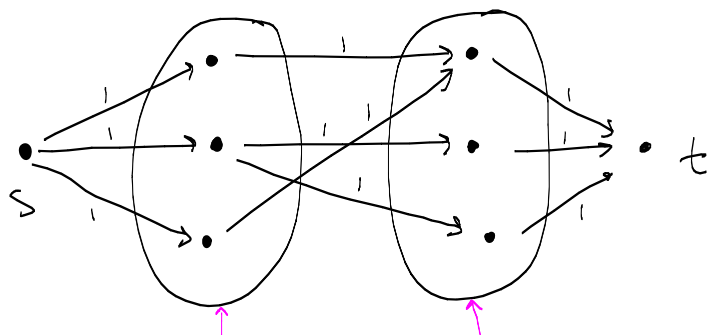
Thus

$C \cdot |E'| = n \cdot (m + 2n)$, so time is $O(nm)$.

* In example, $|X| = 4$, $|Y| = 6$

Example

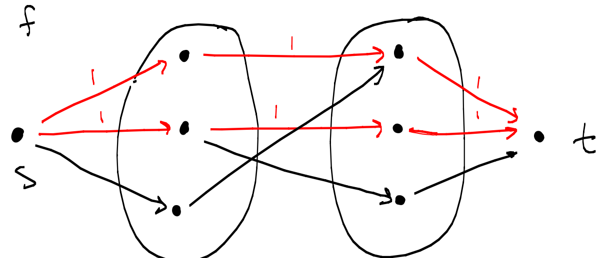
What is max flow? What is min cut?



These play no role once flow network is defined but I'll keep them there anyhow.

e.g. Max flow $|f| = 2$

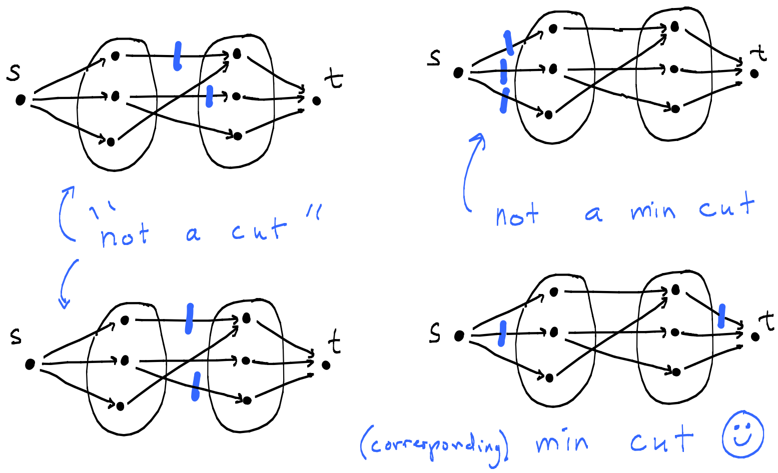
G, f



There are other flows having $|f| = 2$.

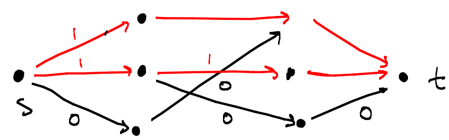
What is the corresponding min cut?

Find any min cut with capacity 2. (All edges have capacity = 1.)

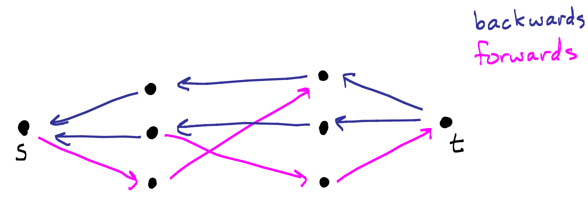


To find a min cut, compute a max flow.

flow f

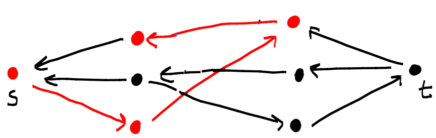


residual graph G_f

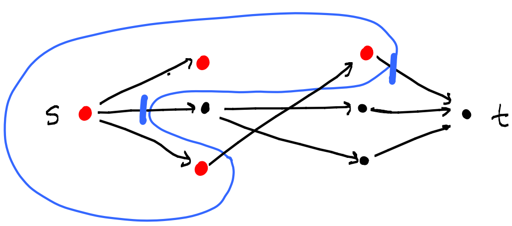


To find the cut, run BFS(G_f', s).
i.e. find vertices reachable from s.

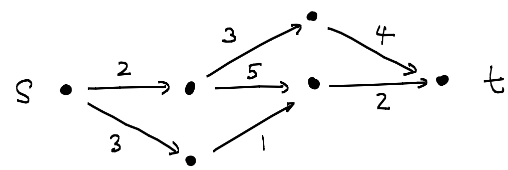
residual graph G_f'



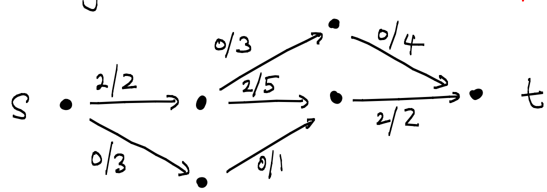
min cut (in G)



Exercise: Given a flow network G, c



and given flow [Note the notation $f(e)/c(e)$]



Find an augmenting path.

Have a great weekend

Happy Valentine's Day ♡ .