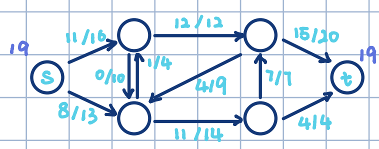
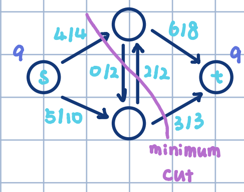
Definition (Informal)

We are given a directed positively weighted graph with a source S and a sink t. Think of it like water pipes with capacity: we want maximize water quantity from source s to sink t.

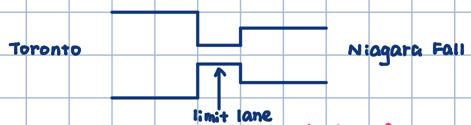




capacity using / capacity

NO Leakage!

MAX FLOW / MIN CUT Theorem



limit lane is gonna heavily decide how fast to arrive

Definition (formal)

Given positively weighted, connected, directed graph G where for every edge $u \to v$ there is a capacity $c(u,v) \geq 0$. If $(u,v) \notin E$ then $c(u,v) = 0$. We distinguish two vertices as source S and sink t. A flow is a function $f: V^2 \to \mathbb{R}$ with properties:

Capacity Constraint

$\forall (u,v) \in V$ we have $f(u,v) \leq c(u,v)$

Skew Symmetry

$\forall (u,v) \in V$ we got $f(u,v) = -f(v,u)$

Flow Conservation

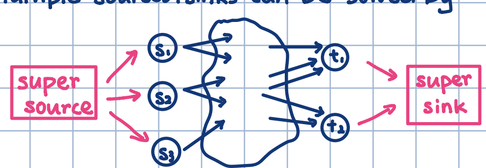
$\forall u \in V - \{s,t\}$ we got $\sum_{v \in V} f(u,v) = 0$

Problem

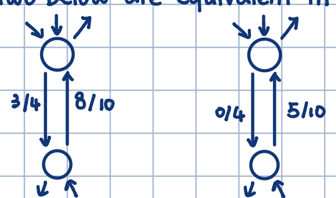
Maximize value of flow in G $|f| = \sum_{v \in V} f(s,v)$

Observation

Multiple sources/sinks can be solved by



The two below are equivalent in terms of flow mathematics



Flow Mathmatics

Notation: $f(X,Y) = \sum_{x \in X} \sum_{y \in Y} f(x,y)$ (X, Y are sets of x, y)

Lemma (properties)

1) $f(X,X) = 0$
2) $f(X,Y) = -f(Y,X)$
3) $f(X \cup Y, W) = f(X,W) + f(Y,W)$ if $X \cap Y = 0$
4) $f(W, X \cup Y) = f(W,X) + f(W,Y)$ if $X \cap Y = 0$

Proof

2). $f(X,Y) = \sum_{x \in X} \sum_{y \in Y} f(x,y) = \sum_{x \in X} \sum_{y \in Y} -f(y,x) = -\sum_{x \in X} \sum_{y \in Y} f(y,x) = -f(Y,X)$

Example

Prove that all units of flow leaving S, they enter t.

$|f| = f(S,V) = f(V,V) - f(V-S,V)$ (property 3)

$= 0 - f(V-s, V)$ (prop 1)

$= f(V, V-s)$ (prop 2)

$= f(V,t) + f(V, V-s-t)$

$= f(V,t) - f(\cancel{V-s-t}^{\,0}, V)$

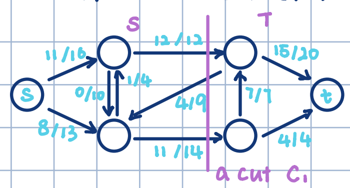
$= f(V,t)$

Definition

A cut (S,T) of a flow network is a partition of the vertices in a disjoint sets (S,T) st $S \cup T = V$, $s \in S$ and $t \in T$.

cut capacity: $C(S,T) = \sum_{S \to T}$ edges

flow of cut: it is $f(S,T)$



$C_1(S,T) = 26$

$f(S,T) = 19$

Residual capacity of edge $C_f(u,v)$

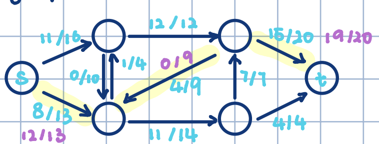
It is the amount of additional flow that can pass through edge (u,v)

$C_f(u,v) = c(u,v) - f(u,v)$

Residual Network

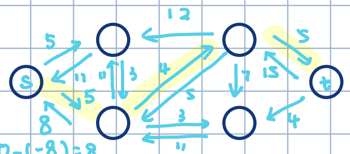
depends on flow

A graph (network) $G_f = (V, E_f)$ where $E_f = \{(u,v) \in V^2 : C_f(u,v) > 0\}$



flow before: 11+8

after: 11+12



$0-(-8)=8$

$13-8=5$

Augmenting Path (AP)

It is a simple path $s \overset{AP}{\rightsquigarrow} t$ in G_f

Residual capacity of AP p:

$$c_f(p) = \min\{c_f(u,v) : (u,v) \in p\}$$

Intuition: AP is a sequence of edges where capacity exceeds existing flow and more flow can be pushed in

Ford Fulkerson (generic)

- Initialize $f(u,v) = \emptyset \quad \forall (u,v) \in E$
- while $\exists$ Augmenting Path $s \overset{AP}{\rightsquigarrow} t$ in residual network G_f

  Increase flow in G by adding the residual capacity of AP

Ford Fulkerson (official, 1956)

$\forall$ edge (u,v) do

$\quad f(u,v) = f(v,u) = \emptyset$

while $\exists$ augmenting path AP

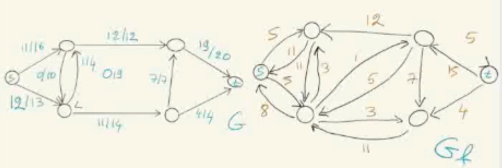
$\quad p : s \overset{AP}{\rightsquigarrow} t$ in G_f do

$\qquad c_f(AP) = \min\{c_f(u,v) \in AP\}$

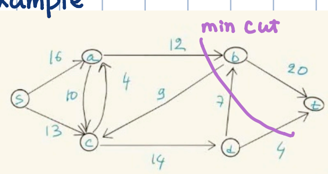
$\qquad$ For every $(u,v) \in AP$ do

$\qquad\quad f(u,v) = f(u,v) + c_f(u,v)$

$\qquad\quad f(v,u) = -f(u,v)$



Example



Augmenting Paths	Flow added
s→a→b→c→d→t	4
s→a→c→d→b→t	7
s→c→a→b→t	8
s→c→b→t	4

Max Flow is 23

why Ford-Fulkerson terminates?
Because of Max Flow/Min Cut theorem

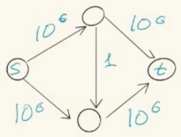
Max Flow – Min Cut theorem

∀ Flow f on graph G, the following statements are equivalent:

- f is a max-flow for G
- G_f has no augmenting path
- $|f| = c(S,T)$ for some cut ⊛

⊛ Another theorem proves that this is the minimum cut

Run time of Ford Fulkerson?



Suppose we use BFS to look for an AP $O(E)$

$Max\ Flow = 2 \cdot 10^6$

Runtime $= O(|f_{MAX}| \cdot E)$

Solution?

(1972)

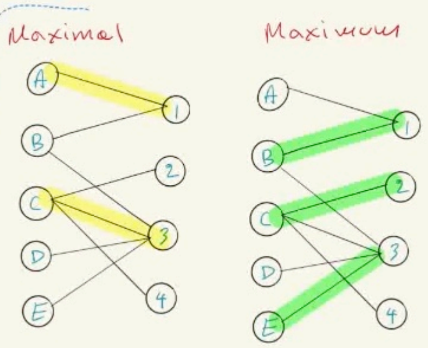
Edmond-Karp's algorithm:

Always choose the augmenting path with the minimum # of edges (shortest path)

Runtime: $O(VE^2)$

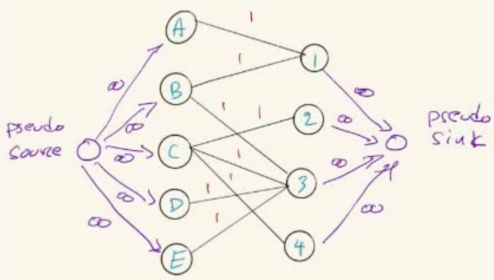
MAXIMUM BIPARTITE MATCHING

Matching:
A subset $M \subseteq E$ in G s.t
$\forall v \in V$ at most one edge from M is incident on v



interested in a maximum matching M (i.e. how many # of edges) in the bipartite graph

We reduce the problem to Max Flow



- Introduce pseudo source/sink connected to the two portitions of the biportite graph
- Put ∞ capacity on new edges
  Capacity 1 on original edges
- Run Edmond-Karp
  $|F| = |M_{max}|$