Lecture #7

<u>Last Time</u>: nearest neighbor search and LSH

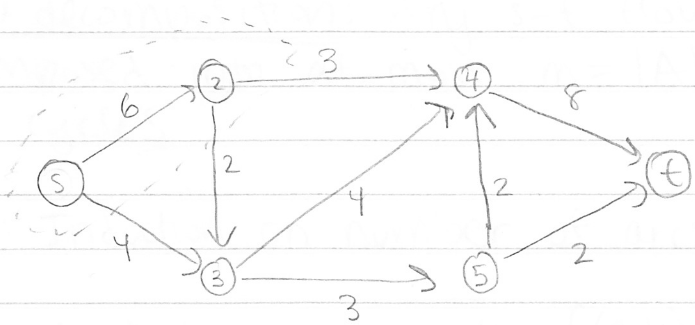
i.e. how to exploit collisions, to get sublinear search time

<u>Today</u>: Maximum Flow (seen before?)

<u>Input</u>: (1) directed graph $G=(V,A)$ ← ordered pairs

(2) capacity fctn $u: A \to \mathbb{R}^+$

(3) source s, sink t $(s,t \in V)$



<u>Goal</u>: Send as much flow as possible from s to t, without violating constraints

$$\max |f| \equiv \sum_a f(s,a) - \sum_a f(a,s)$$

s.t. (1) $\sum_a f(v,a) - \sum_a f(a,v) = 0 \quad \forall v \neq s,t$

(2) $0 \le f(a,b) \le u(a,b) \quad \forall (a,b) \in A$

(1) is called conservation, (2) is called capacity constraints

Examples? packets in computer network,
cars/trains in transportation network

Generalizations? multiple sources/sinks } we will study these
costs

Many applications in combinatorial optimization e.g. matching, scheduling, partitioning, etc

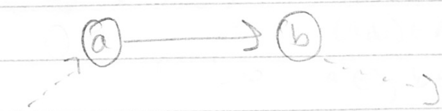
---

Powerful tool for reasoning about flows.
* Assume no arc into s has flow, no arc out of t has flow

<u>Flow decomposition</u>: Any s-t flow can be decomposed into at most $m = |A|$ s-t paths and cycles } simplest s-t flows

<u>Proof</u>: Induction on number of arcs with non-zero flow.

Let (a,b) be any arc with $f(a,b) > 0$. Then trace backwards from a and forwards from b



flow out of a + conservation of flow $\Rightarrow$ flow into a

flow into b + " $\Rightarrow$ flow out of b

Thus we can trace backwards from a and forwards from b until we (1) reach a cycle or
(2) reach both s and t

Let W be the arcs in the cycle, or in the s-t path

$$\Delta(W) = \min_{(a,b) \in W} f(a,b)$$

Decrease flow on cycle/path by $\Delta(W)$ on each arc; add cycle/path with $\Delta(W)$ flow to flow decomp.

still get valid s-t flow but

The # of edges with positive flow has strictly decreased ☒

Question: If s-t flow $\xrightarrow{\text{decomp}}$ s-t paths + cycles, what if we zero out cycles? would we get an s-t flow? And of what value?

Now let's apply flow decomposition

def: An s-t cut is a set $S \subseteq V$ where $s \in S$ and $t \in V \setminus S$. The capacity is

$$\mathrm{cap}(S, V \setminus S) = \sum_{\substack{(a,b) \in A \\ a \in S,\, b \in V \setminus S}} u(a,b)$$

only forward edges

Lemma: $\underset{\substack{(\max |f|) \\ \text{s-t flows}}}{\text{max flow}} \leq \underset{\substack{(\min \mathrm{cap}(S, V \setminus S)) \\ \text{s-t cuts}}}{\text{min cut}}$

Proof: Take any max flow, and decompose it into paths and cycles.

(1) Each s-t path must cross the cut $(S, V \setminus S)$ at least once

Hence a path that contributes p to total flow uses up at least p units of capacity in $(S, V \setminus S)$

(2) cycles do not contribute to $|f|$

(if flow > cut, we used up more capacity than is available) $\boxtimes$

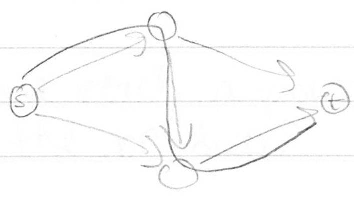
Max-Flow Attempt #1

If W is a path, let $\delta(W) = \min_{(a,b)\in W} u(a,b) - f(a,b)$

while $\exists$ s-t path W with $\delta(W) > 0$

increase flow on each arc in W by $\delta(W)$

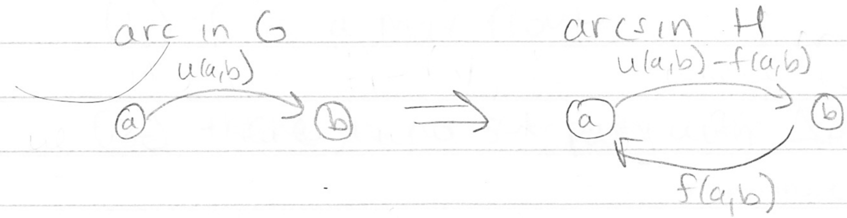
Does this algorithm succeed in finding the max flow? What can go wrong?



augment along one path, but when we're done no more s-t paths left

This motivates the notion of residual graph

Residual Graph: Given instance of max flow problem on $G=(V,A)$ and s-t flow f



The meaning of arc (b,a) is that $f(a,b)$ units of flow in f can be undone.

Max Flow Attempt #2:

Let $\Delta_H(W) = \min_{(a,b)\in W} \underbrace{u_H}_{\text{capacity in } H}(a,b)$

While $\exists$ s-t path W in H with $\Delta_H(W) > 0$

augment along W by $\Delta_H(W)$ (could mean undoing flow in G on some edges)

update residual

What if we reach a point where there is no path in the residual?

Let S be the set of nodes reachable (in residual) from s. Then

(1) every arc (a,b) with $a \in S$, $b \in V \setminus S$ has $\underbrace{f(a,b) = u(a,b)}_{\text{saturated}}$ (else we could reach b in residual)

(2) every arc (b,a) with $a \in S$, $b \in V \setminus S$ has $f(b,a) = 0$ (else there would be an arc (a,b) in residual)

Thus $cap(S, V\setminus S) = \sum_{\substack{(a,b) \\ a\in S,\, b\in V\setminus S}} f(a,b) \overset{\text{claim}}{=} |f|$

This proves max flow = min cut

---

the following conditions are equivalent

(1) f is a max flow

(2) there is no s-t path with $\Delta_H(u) > 0$ in residual

often ignore zero capacity edges

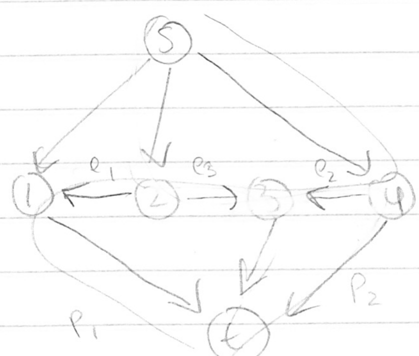
(3) $|f| = cap(S, V/S)$ for some s-t cut

Clearly $(3) \Rightarrow (1)$

$\neg(2) \Rightarrow \neg(1)$

and $(2) \Rightarrow (3)$ by claim

There are many complications to how you choose augmenting path



$u(e_1) = 1$

$u(e_2) = r = \frac{\sqrt{5}-1}{2}$

$u(e_3) = 1$

$u(\text{else}) = M \geq 2$ integer

Path	sent flow	residual capacities after e_1	e_2	e_3
$s \to 2 \to 3 \to t$	1	1	r	0
P_1	r	r^2	0	r
P_2	r	r^2	r	0
P_1	r^2	0	r^3	r^2
P_3	r^2	r^2	r^3	0

$P_1 = s \to 4 \to 3 \to 2 \to 1 \to t$

$P_2 = s \to 2 \to 3 \to 4 \to t$

$P_3 = s \to 1 \to 2 \to 3 \to t$

Flow converges to

$1 + 2\sum_{i=1}^{\infty} r^i = 3 + 2r < 2M+1$

For irrational capacities, augmenting paths need not converge.

But for integer capacities, it always finds an integer-valued max flow, and this is important in many settings