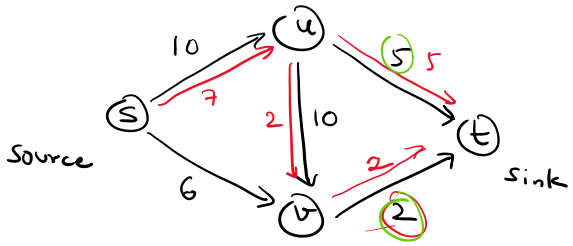
Network Flows



$|f| = 7$

maximum flow

$\forall$ edges $e = (a, b)$

there is a capacity (integer)

$c(e) \geq 0$

goal: maximize the "flow" from s to t

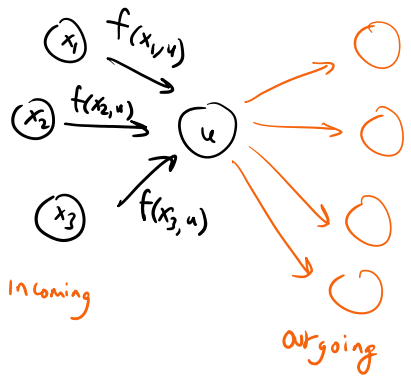
?

1) flow along an edge $\equiv f(e) \geq 0$

$f(e)$ is also integer

$$\underset{\text{capacity}}{c(e)} \geq \underset{\text{flow}}{f(e)} \geq 0$$

2) flow must be <u>conserved</u> at each vertex u



incoming flow

$$f^{in}(u) = \sum_{e=(v,u)} f(v, u)$$

outgoing flow

$$f^{our}(u) = \sum_{e=(u,v)} f(u, v)$$

$$\boxed{f^{in}(u) = f^{our}(u)}$$

3) Value of the flow

$$|f| = \boxed{f^{our}(s)} = f^{in}(t)$$

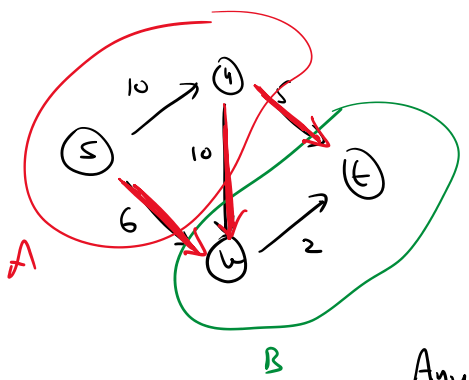
$$|f| = \boxed{f^{out}(s)} = \underline{\underline{f^{in}(t)}}$$

<u>Problem / Task:</u> Maximize the flow (find value of $f(e)$ $\forall e$)

$$\left[\begin{array}{l} \text{Max } f^{out}(s) \\ \text{such that } c(e) \geq f(e) \geq 0 \\ \quad \text{and flow is conserved } \forall u, \text{ i.e. } f^{in}(u) = f^{out}(u) \\ \qquad\qquad \text{(except for } s \text{ and } t\text{)} \end{array}\right.$$

---

duality : $\boxed{\text{Maximum flow} = \text{minimum cut}}$ in the graph



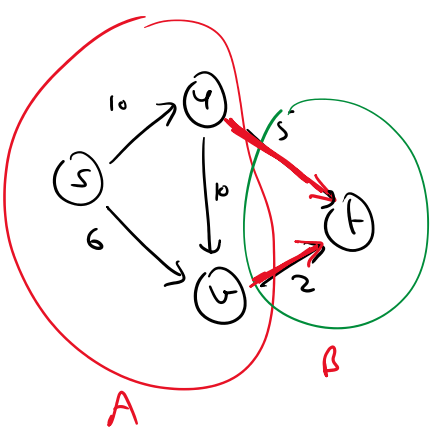
$A = \{s, u\}$

$B = \{t, v\}$

Capacity Cut $(A, B) = 5 + 10 + 6 = 21$

Any flow must cross the cut

$\therefore$ $|f| \leq$ capacity of cut ≤ 21



$\text{Cut}(A, B) = 5 + 2 = \underline{7}$

$|f| \leq \underline{7}$

$|f| = \underline{7}$

Maximum flow = min cut

Ford - Fulkerson Algorithm

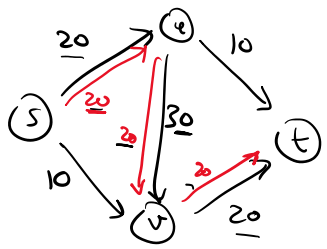
given a graph G, and given some flow along the edge,

we will define a <u>residual graph</u> G^f

The algo finds paths from s to t in G^f and this to increase the flow along that path

G : graph

p: flow

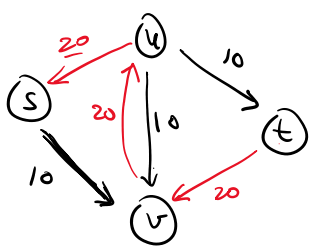


path: $s \to u \to v \to t$

find the minimum capacity for any edge on the path

$\Downarrow$ Residual graph

G^f



1) for any edge $(a,b) \in G$ such that $f(a,b) > 0$

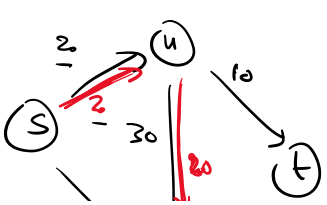
In the residual graph G^f add edge (a,b) with "residual" capacity

$$c(e) - f(e)$$

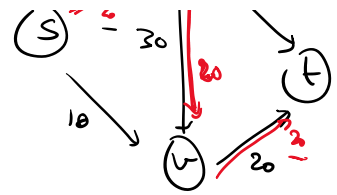
2) we also add a backward edge in G^f

$$(b,a) = \underline{\underline{f(e)}}$$

$G^f = G$



find a path

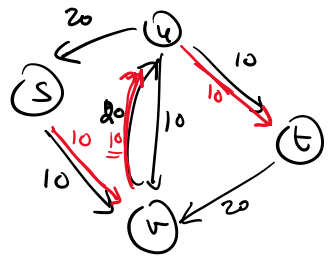


find a path

$s \to u \to v \to t$: 20

sending 20 units from $u \to v$

G^f

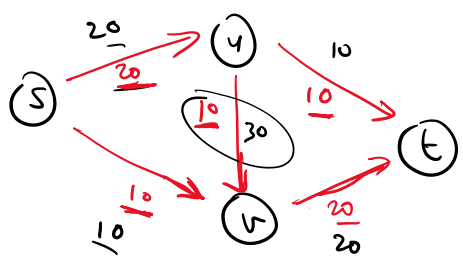


find another path

$s \to v \to u \to t$: 10

send 10 units from v to u

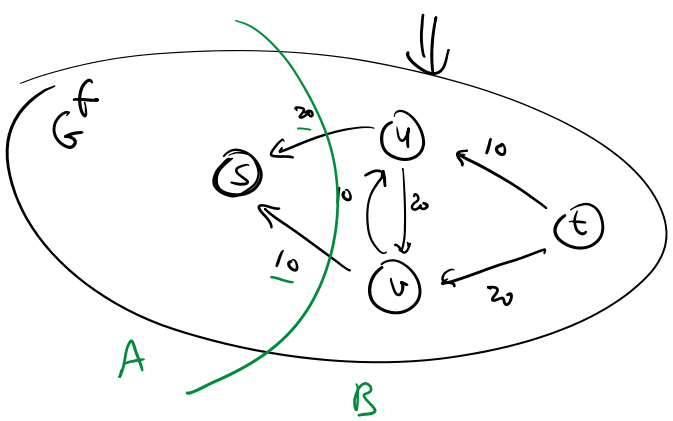
G



$|f| = 30$

$f^{in}(u) = 20 \quad f^{out}(u) = 20$

$f^{in}(v) = 20 \quad f^{out}(v) = 20$



find a path from s to t

There is no such path!

done!

$cut(A,B) = 30$

Ford-Fulkerson Algorithm

$f(e) = 0 \quad \forall e \in G$

while $\exists$ a $\boxed{\text{path from } s \text{ to } t}$ in G^f

shortest path from s to t using edge cost 1

$p: s \rightsquigarrow t$ ← use DFS/BFS $O(|V| + |E|)$

?

residual

?

#Q iterations

$p: s \rightsquigarrow t$

$c(p)$: capacity of p (least residual capacity edge in p)

augment the flow along p

recompute G ← $O(|E|+|V|)$

recompute G^f ← $O(2|E|+|V|)$

possibly G^f has both forward and backward edge

Computational complexity?

each iteration of while

$O(|V|+|E|)$ time

how many iterations before we stop?

each time we find a path $p: s \rightsquigarrow t$, the flow increase by $c(p)$

In the worst case, the flow increase by $\underline{1}$ unit

any flow $|f| \le \boxed{\sum_{e \in G} c(e) = C}$

upper bound on the value of the flow

$O(C)$ iterations in the worst case!

$$O\left(C \cdot (|V| + |E|)\right)$$

$$= O(C \cdot |E|)$$

$$= O(|f^*| \cdot |E|)$$

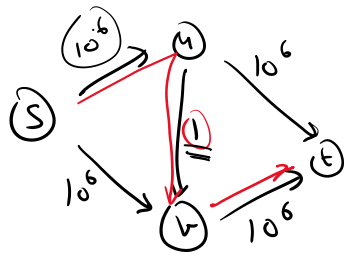
pseudo polynomial running time in worst case

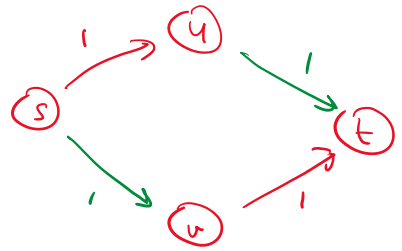
maximum flow value

flow so far

max flow value

G

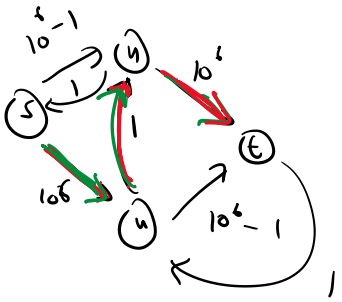


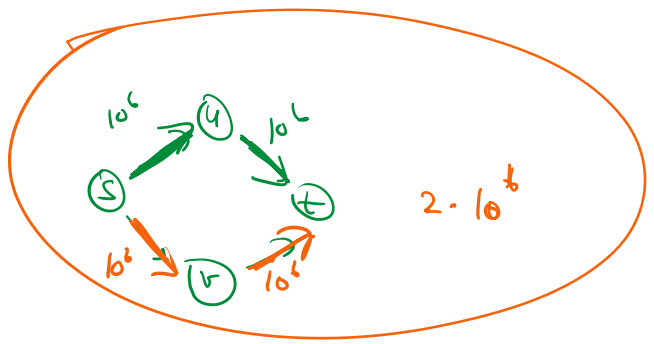


worst case

it takes $2 \cdot 10^6$ steps to get to max flow

G^f





best case

Edmonds Karp Algorithm

In FF (Ford Fulkerson)

Choose the paths "wisely"

always find shortest paths (via BFS), in terms of the number of edges, between s and t.

→ then the # of iterations is bounded by $O(|V| \cdot |E|)$

proof is somewhat involved

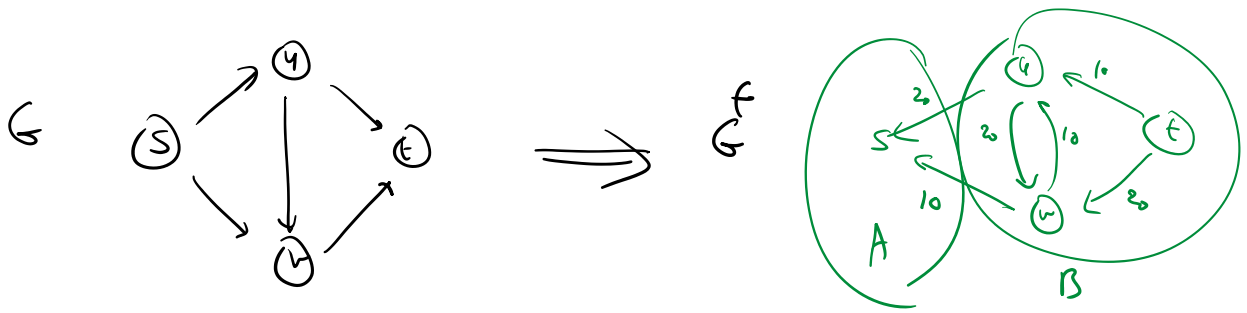
And therefore the total time

$$O(|V| \cdot |E| \cdot |E|) = O(|V| \cdot |E|^2)$$

$$O(\underbrace{|V|\cdot|E|}_{\#\text{ of iterations}} \cdot \underbrace{|E|}_{\text{per iter cost}}) = O(|V|\cdot|E|^2)$$

When FF/EK stops, we can extract both the flow & cut info from G^f

i.e. $\boxed{\text{maximum flow} = \text{min cut}}$

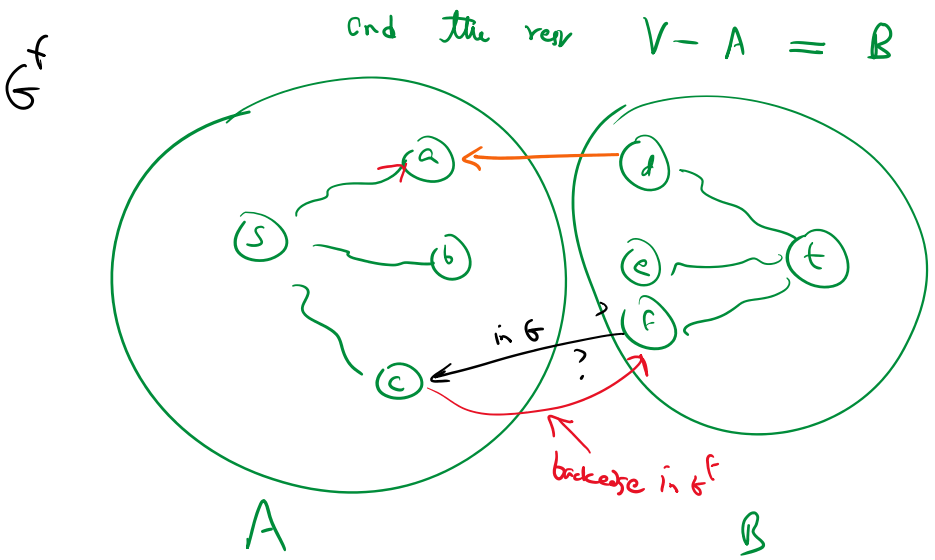


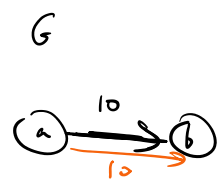
In G^f: find all vertice that are reachable from s;

they comprise one side of the cut,

A

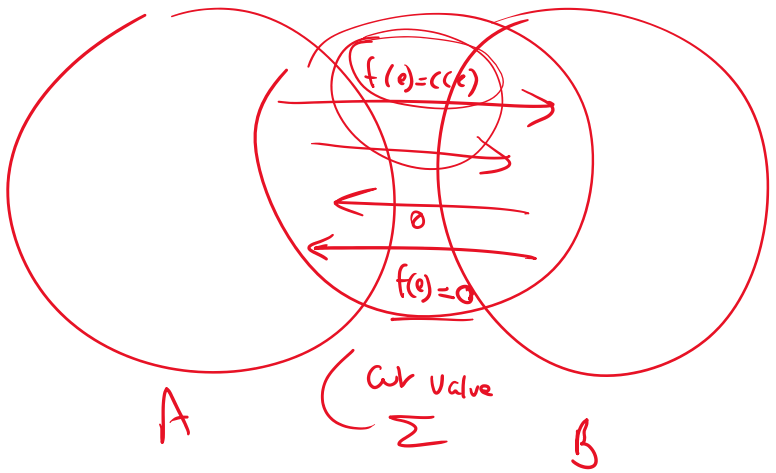
and the rest $V - A = B$





c) Any forward edge is an edge in G is saturated

If there an edge from B vertex to A vertex then it must have flow 0



cut value

Σ

$$\Rightarrow \min \text{cut}(A, B) = \max \text{flow in } G$$