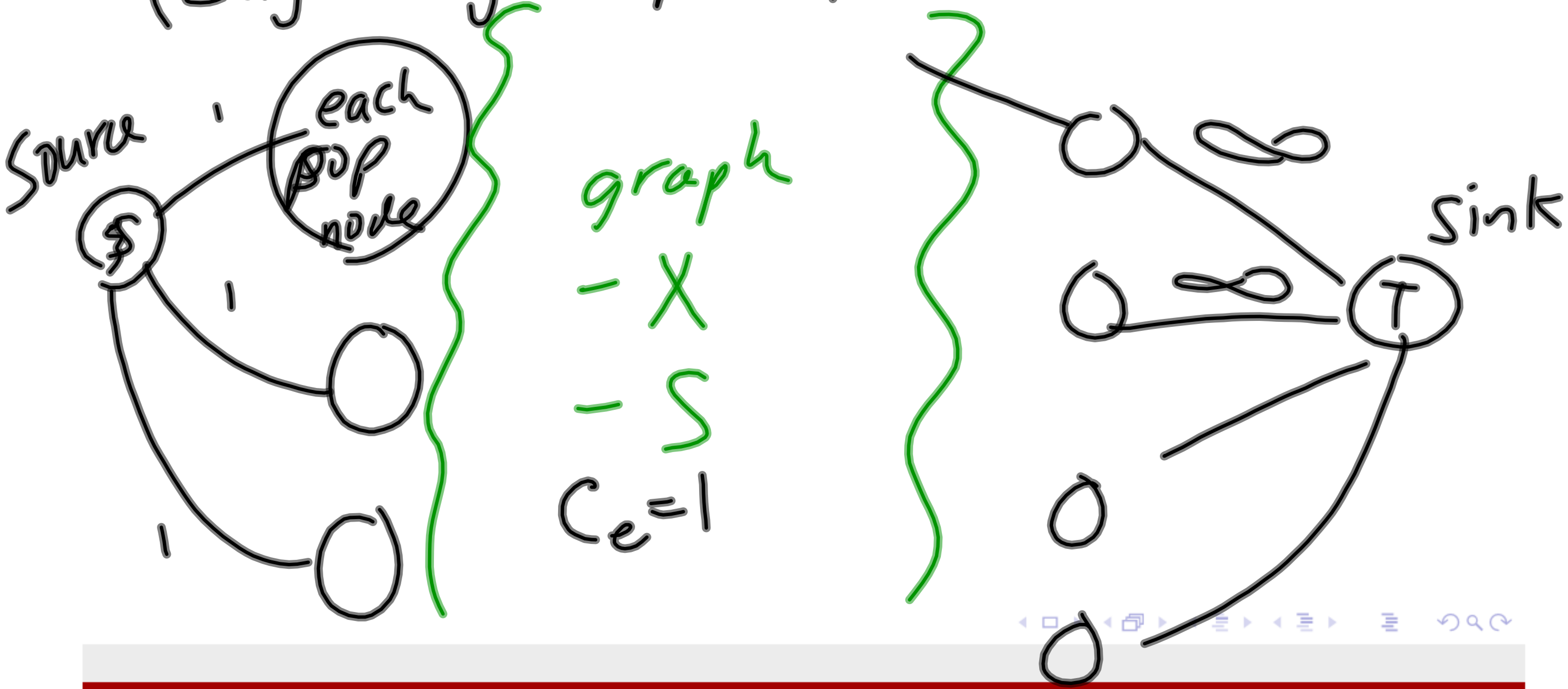
Edge Disjoint Paths

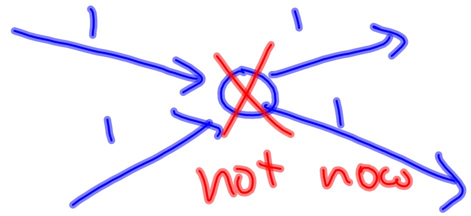
Given G, X, and S, show how to decide in polynomial time whether such a set of evacuation routes exists.

(Edge-disjoint paths) Safe



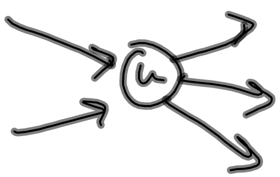
Suppose we have exactly the same problem as in (a), but we want to enforce an even stronger version of the "no congestion" condition (iii). Thus we change (iii) to say "the paths do not share any *nodes*."

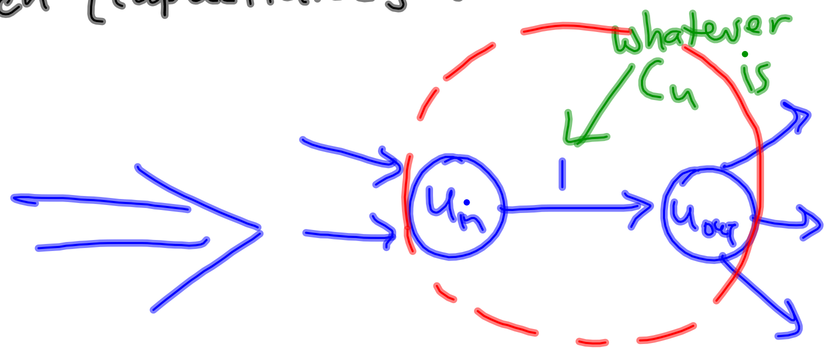
old okay:



for each [capacitated] vertex u

old:

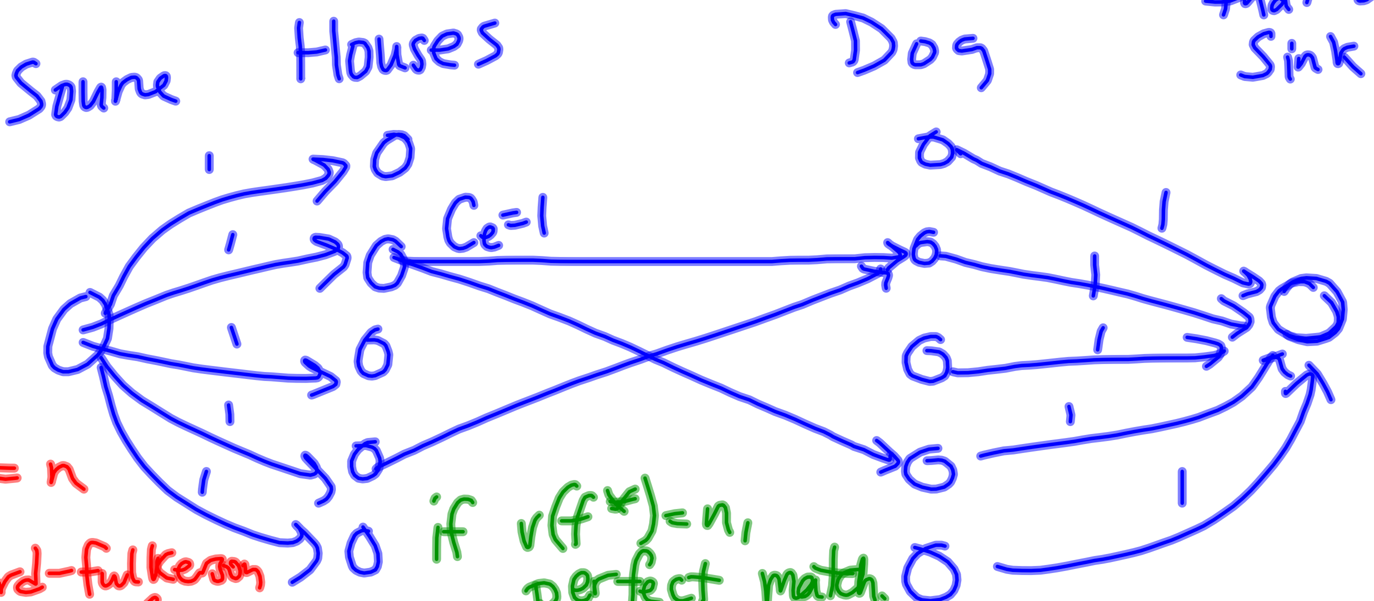




Dog Adoption...

edge House → Dog iff that house wants that dog

(can you follow this idea to solve the tennis problem?)



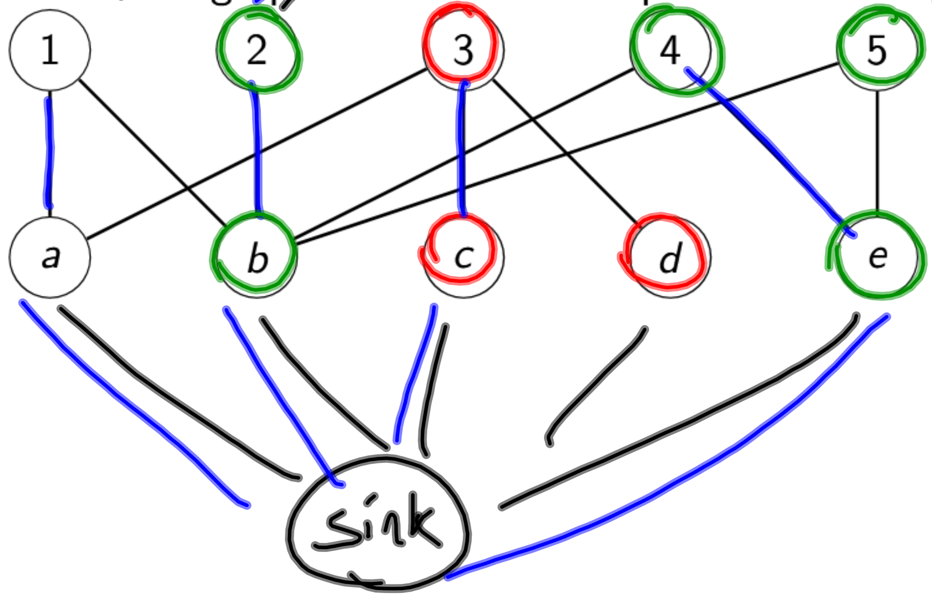
$C = n$

Ford-Fulkerson $O(mn)$ (on this graph)

if $v(f^*) = n$, perfect match. Every house gets a dog!

Hall's Theorem

Prove this graph *does not* have a perfect matching:

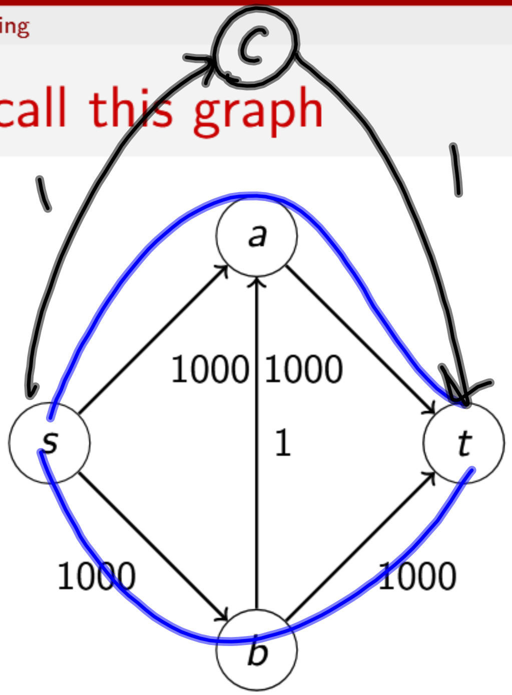


A in (A,B) min cut

{2, 4, 5 / b, e}

How can we convince someone that it isn't possible to adopt every dog?

Recall this graph



What would it be useful to do to help someone find flow quickly here?

How to find a “good” path?

How can I find out if a path with residual capacity *at least* Δ (for some given integer Δ) remains in G_f?

$\exists$? path $s \rightsquigarrow t$ in G_f w/ residual cap $\geq \Delta$?

BFS w/ ignore edges w/ res. cap $< \Delta$

(last lecture/F-F alg: $\Delta = 1$, always)

$G_f(\Delta) = G_f$ w/o $< \Delta$ edge caps

$\forall_e \;\; f_e = 0$
$X =$ maximum c_e out of s
for $\Delta = 2^{\lfloor \log_2 X \rfloor}$; $\Delta \geq 1$; $\Delta = \Delta/2$ **do**
  while $\exists$ path p from s to t in $G_f(\Delta)$ **do**
    $p =$ any simple s to t path in $G_f(\Delta)$
    $b =$ min residual capacity edge on p (the "bottleneck" edge)
    for all edges $e = (u, v)$ in p **do**
      if e is forward **then**
        $f_e = f_e + b$
      else
        $e' = (v, u)$
        $f_{e'} = f_{e'} - b$

Total: $O(m^2 \log C)$

What is that algorithm's running time?

- How many iterations of the outer loop will happen?

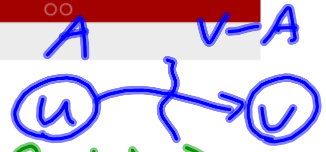
  $O(\log C)$ = # bits to rep. caps

- How long will each iteration of the inner **while** loop take?

  $O(m)$

- How many iterations of the inner **while** loop will happen during each iteration of the outer loop?

<u>After</u> some value Δ // A = BFS from s in $G_f(\Delta)$



$f_e + \Delta > c_e$

$c_e - f_e < \Delta$

$f_e > c_e - \Delta$

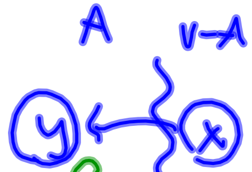
So $O(m)$ iterations

$$v(f) = \sum_{e \text{ out of } A} f_e - \sum_{e \text{ into } A} f_e$$

$$\geq \sum (c_e - \Delta) - \sum \Delta$$

$$= \sum_{e \text{ out}} c_e - m\Delta - m\Delta$$

$$v(f) \geq \text{min-cut} - 2m\Delta$$



$f_e < \Delta$

Does the Δ-scaling flow algorithm correctly find maximum flow?

Yes. base: $\forall e\ f_e = 0$ is valid flow
each augment retains validity

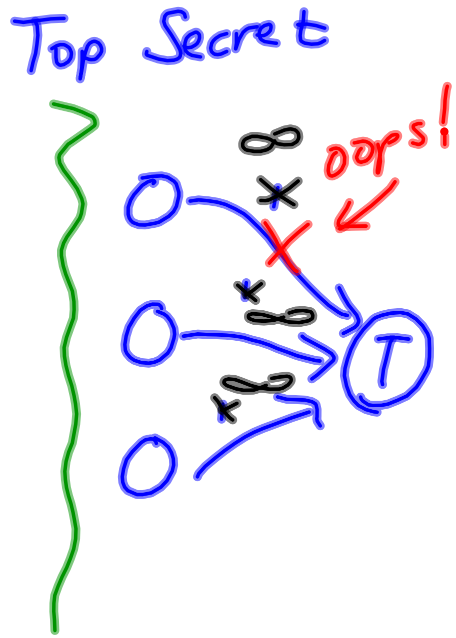
eventually $\Delta = 1$,
F-F from any valid flow
finds max flow

Graph

$c_e = 1$

Call min-cut.

Place traps on cut edges



$C_e = 1$ all e, compute edge-disjoint paths in orig

Each path has $O(n)$ edges

$O(\log n)$ pings per path,

find cut edge by binary search.

$O(k \log n)$ pings

Build bipartite: