CS7800: Advanced Algorithms

Network Flow II

- Heavy augmenting paths
- Short augmenting paths

Jonathan Ullman

09-30-22

Recap

① Defined max s-t flow and min s-t cut problems

② Ford-Fulkerson Algorithm for solving max flow

- add augmenting paths in the residual graph

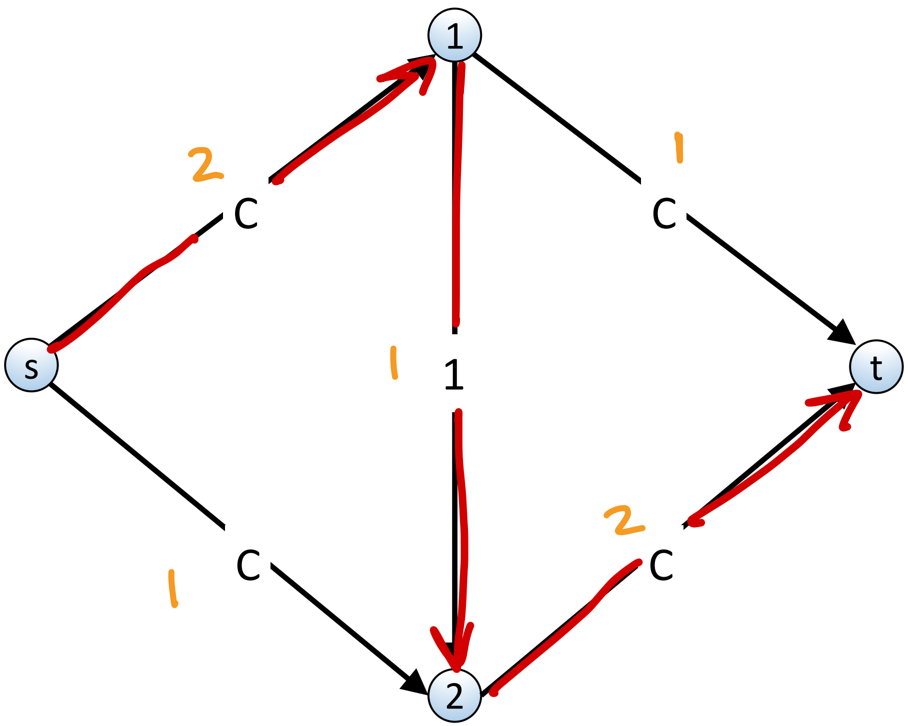
③ Max Flow - Min Cut Theorem

- an $O(m)$ time for finding a min cut given G, f^*

④ In graphs with integer capacities $c(e) \in \{1, 2, \dots, C\}$
Ford-Fulkerson terminates after $\leq val(f^*) \leq nC$ paths

Choosing (bad) augmenting paths

If we dont specify anything about the paths,
then we might need $val(f^*)$ paths
$(= \Omega(nC))$



$val(f^*) = 2C$

# of iterations can be $2C$

The Capacity Scaling Algorithm

"scale parameter"

```
CapacityScaling(G,s,t,{c(e)})
    C ← max_e {c(e)}, L ← C
    for e ∈ E: f(e) ← 0
    G_f is the residual graph

    while (L ≥ 1)
       while (there is an s-t path P in G_f
       consisting of edges with c(e) ≥ L)
           f ← Augment(G_f,P)
           update G_f
       L ← L/2

    return f
```

Let $G_f(L)$ be the residual w/ any edge of cap $< L$ removed.

Running time of capacity scaling

$L=C$ $\quad P_1$
$\quad P_2$ } epoch

$L=C/2$ $\quad P_3$
$\quad P_4$
$\quad P_5$ } epoch

$L=C/4$ $\quad P_6$

$\vdots$

$L=4$ $\quad P_{73}$

$L=2$ $\quad P_{74}$
$\quad P_{75}$

$L=1$ $\quad P_{76}$

$\#$ of epochs $= \log_2 C$

$\boxed{\text{\underline{Goal}: Prove that each epoch has } O(m) \text{ augmenting paths}}$

$\Downarrow$

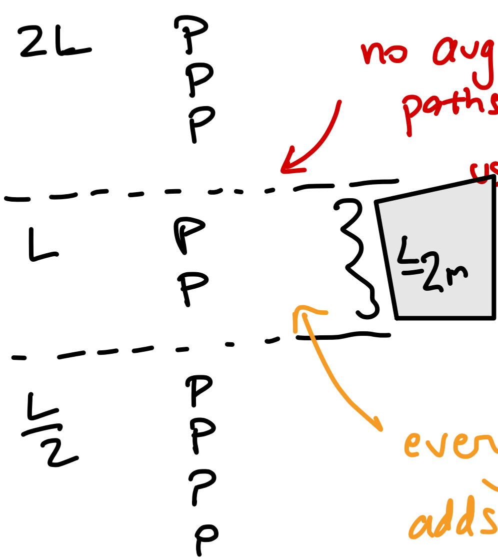
Running Time:

$$\#\text{epochs} \times \frac{\#\text{paths}}{\text{epoch}} \times \frac{\text{time}}{\text{path}}$$

$$O(\log C) \times O(m) \times O(m)$$
$$= O(m^2 \log C)$$

Running time of capacity scaling

fix some epoch L



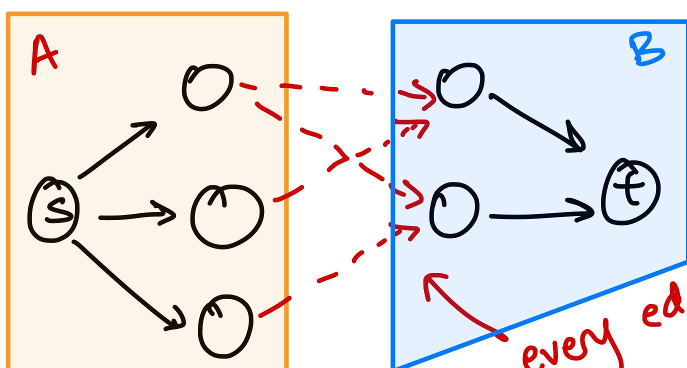
no augmenting paths in G_f using edges of capacity $\geq 2L$

every augmentation adds flow $\geq L$

- At the end of epoch $2L$, $G_f(2L)$ has no paths
- At the end of epoch $2L$ there is $\leq$??? flow left in G_f

$\forall f, \; val(f^*) = val(f) + val(\text{max in } G_f)$

$G_f(2L)$



every edge from $A \to B$ has capacity $\leq 2L$

s, t are disconnected

$val(\text{max in } G_f)$

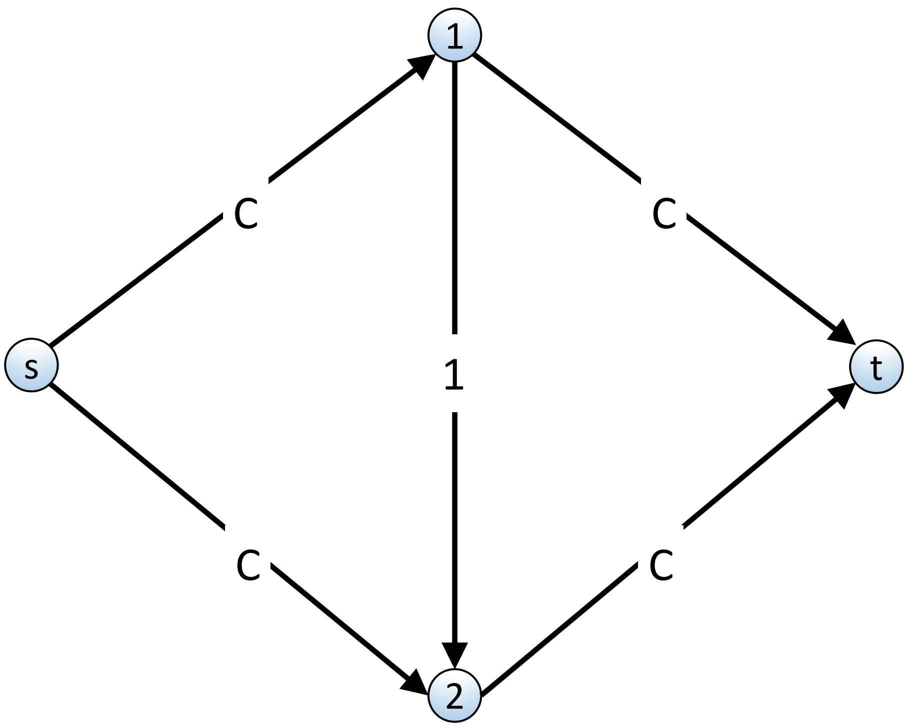
$\leq cap(A, B)$

$\leq m \cdot (2L)$

$\Rightarrow \leq 2m$ augmentations

Choosing (bad) augmenting paths

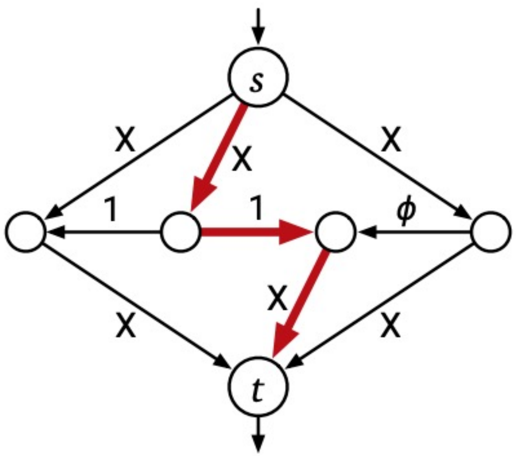
Candidate Rule: Choose the avg path with fewest hops

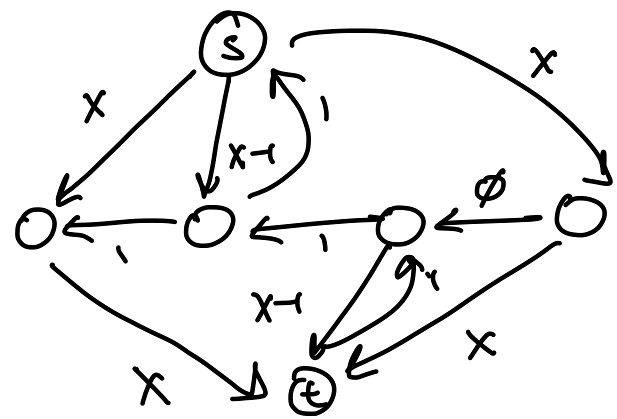


Choosing (bad) augmenting paths

X is some large integer

$\phi = \frac{(\sqrt{5}-1)}{2}$ satisfies $\phi^2 = 1 - \phi$

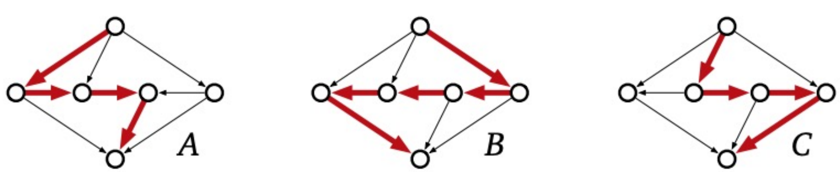




$1-\phi$ $\quad$ $1-\phi$ $\quad$ 0

$1-\phi$ $\quad$ 1 $\quad$ ϕ

Alternate paths BCBA BCBA BCBA BCBA



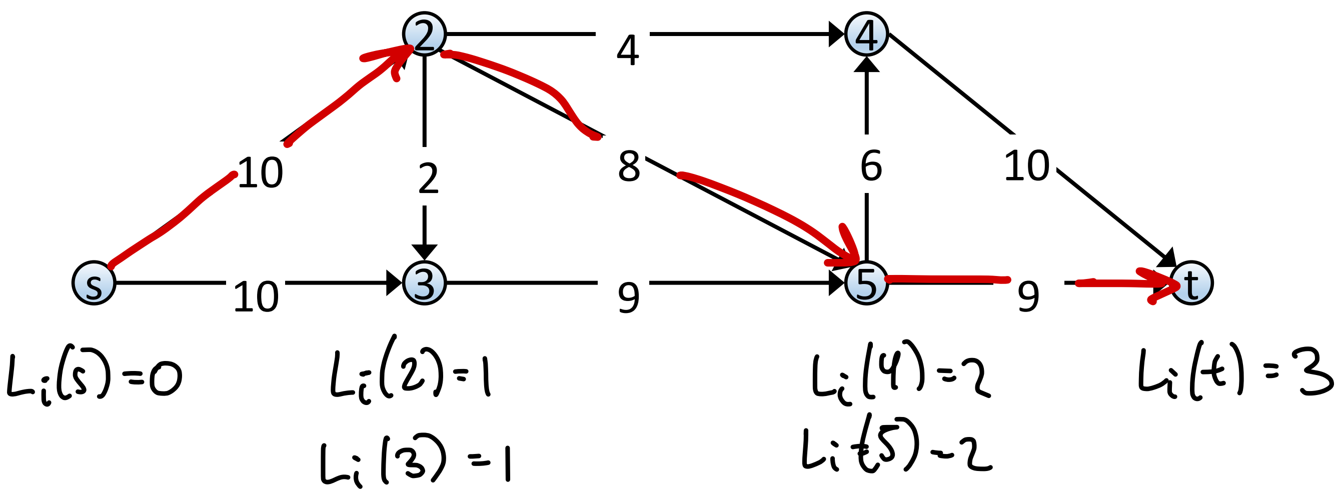
Shortest Augmenting Path (Edmunds-Karp)

- Find the augmenting path with the fewest hops
  - Can find shortest augmenting path in $O(m)$ time using BFS

- **Theorem:** for any capacities $\frac{nm}{2}$ augmentations suffice
  - Overall running time $O(m^2 n)$
  - Works for any capacities!

Shortest Augmenting Path

- Let f_i be the flow after the i-th augmenting path
- Let $G_i = G_{f_i}$ be the i-th residual graph
- Let $L_i(v)$ be the distance from s to v in G_i
  - Recall that the shortest path in G_i moves layer-by-layer



Shortest Augmenting Path

- Every augmentation causes at least one edge to disappear from the residual graph, may also cause an edge to appear

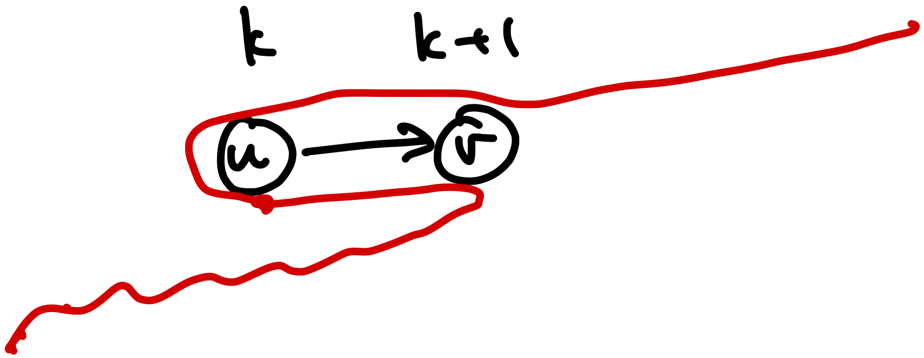
- **Key Property:** each edge disappears at most $\frac{n}{2}$ times
  - Means that there are at most $\frac{mn}{2}$ augmentaitons

Shortest Augmenting Path

- **Claim 1:** for every $v \in V$ and every i, $L_{i+1}(v) \geq L_i(v)$
  - Obvious for $v = s$ because $L_i(s) = 0$
  - Suppose for the sake of contradiction that $L_{i+1}(v) < L_i(v)$
    - Let v be the smallest such node
  - Let $s \rightsquigarrow u \to v$ be a shortest path in G_{i+1}
    - By optimality of the path, $L_{i+1}(v) = L_{i+1}(u) + 1$
    - By assumption, $L_{i+1}(u) \geq L_i(u)$

Shortest Augmenting Path

- **Claim 2:** If an edge $u \to v$ disappears from G_i and reappears in G_{j+1} then $L_j(u) \geq L_i(u) + 2$



- **Claim 3:** An edge (u, v) cannot reappear more than $\frac{n}{2}$ times
  - $0 \leq L_i(u) \leq n$
  - By Claim 2: length increases by 2 for each reappearance