CS 7800: Advanced Algorithms

Class 8: Network Flow II

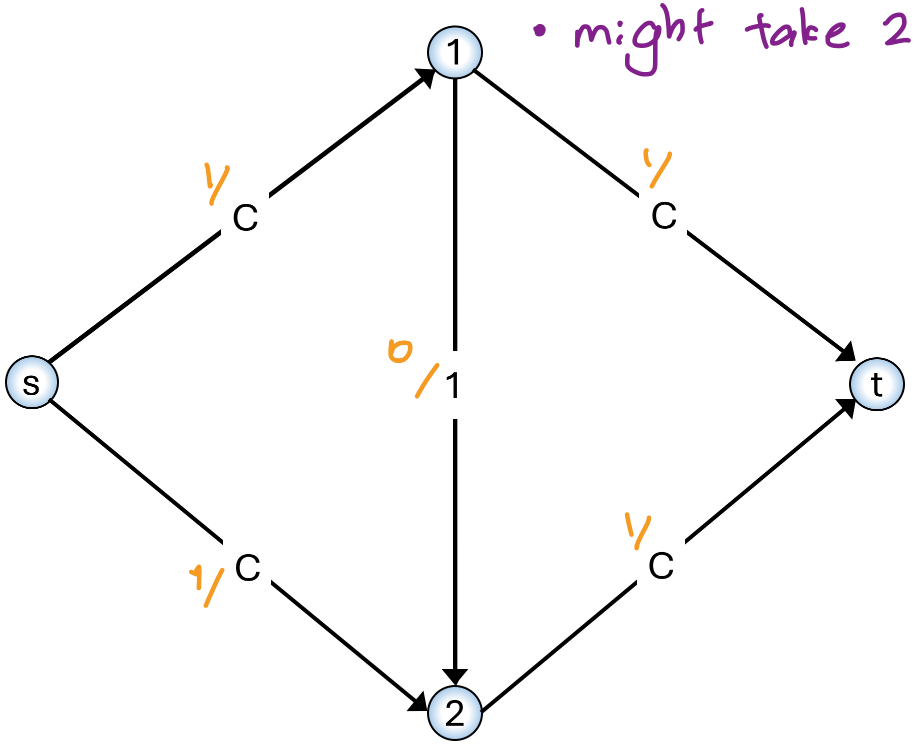
- Choosing Good Augmenting Paths

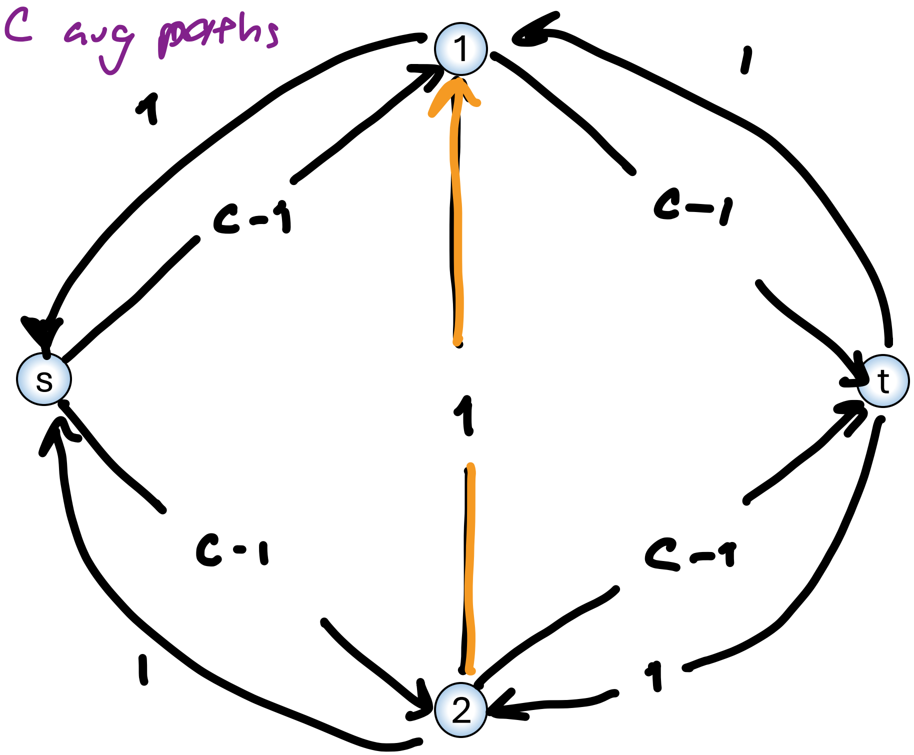
Jonathan Ullman

September 30, 2025

Ford-Fulkerson can be Slow

- Start with $f(e) = 0$ for all edges $e \in E$
- Find an **augmenting path** P in the **residual graph** G_f
- Repeat until you get stuck
  - $val(f^*) = 2C$
  - might take $2C$ aug paths





Choosing Good Augmenting Paths

Last Time: Ford Fulkerson (integer capacities) needs at most $\mathrm{val}(f^*)$ augmenting paths

- Might be tight for <u>some</u> choice of paths

Today: How to make better choices of paths

- widest augmenting path
- shortest augmenting path

Widest Augmenting Path

- Widest augmenting path

$$\max_{s,t \text{ paths } P} \left(\min_{e \in P} c(e) \right) \text{ in the residual graphs}$$

"Maximum bottleneck path"

- Can find the widest augmenting path in time $O(m \log n)$ in several different ways
  - BFS + binary search
  - Variants of Prim's or Kruskal's MST algorithm

Widest Augmenting Path

Arbitrary Paths

- **Assume integer capacities**

- Value of maxflow: v^*
- Value of aug path: ≥ 1
- Flow remaining in G_f: $\leq v^* - 1$
- # of aug paths: $\leq v^*$

Widest Path

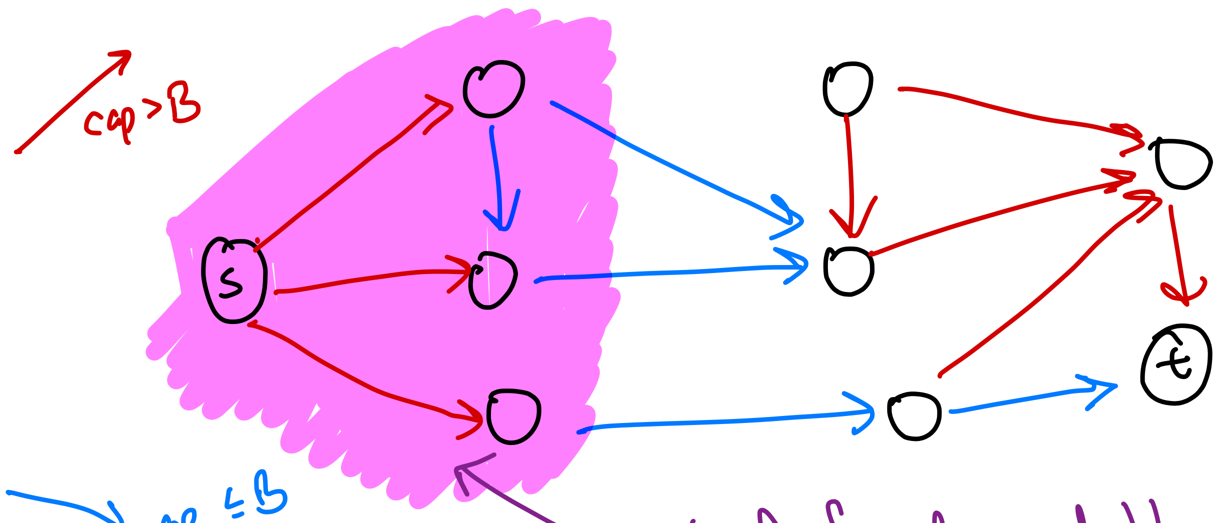
Maximum-Capacity Path

- **Assume integer capacities**

- Value of maxflow: v^*
- Value of aug path:
- Flow remaining in G_f:
- # of aug paths:

Widest Augmenting Path

- f^* is a maximum flow with value $v^* = val(f^*)$
- P is a widest augmenting s-t path with capacity B
- **Key Claim:** $B \geq \frac{v^*}{m}$ widest path $\geq \frac{\text{value of max flow}}{\text{\# of edges}}$
- **Proof:**
  - If widest path has capacity B then s and t are disconnected in the graph of only edges with capacity >B



- $v^* \leq cap(A, A^c)$
  $\leq B \cdot m$
- $\Rightarrow B \geq \frac{v^*}{m}$

cut A of nodes reachable via edges with cap >B

Widest Augmenting Path

- f^* is a maximum flow with value $v^* = val(f^*)$
- P is a widest augmenting s-t path with capacity B
- **Key Claim:** $B \geq \frac{v^*}{m}$
- **Proof:**

Key Observation about Flows:

$$\boxed{\text{① For every } f, \quad val(f^*) = val(f) + \text{value of max flow in } G_f}$$

② For every f, " $f^* = f +$ max flow in G_f "

[Not exactly a true statement]

Widest Augmenting Path

Arbitrary Paths

- **Assume integer capacities**
- Value of maxflow: v^*
- Value of aug path: ≥ 1
- Flow remaining in G_f: $\leq v^* - 1$
- # of aug paths: $\leq v^*$

Maximum-Capacity Path

- **Assume integer capacities**
- Value of maxflow: v^*
- Value of aug path: $\geq \frac{v^*}{m}$
- Flow remaining in G_f: $\leq \left(1-\frac{1}{m}\right)\cdot v^*$
- # of aug paths: $\leq m\cdot\ln(v^*)+1$

① If there exists an aug path, it has value ≥ 1

② After t augmentations, flow remaining is $\leq v^*\cdot\left(1-\frac{1}{m}\right)^t \leq v^*\cdot e^{-t/m}$

$$v^*\cdot e^{-\frac{(t-1)}{m}} \geq 1 \iff \boxed{t \leq m\cdot\ln(v^*)+1}$$

Choosing Good Augmenting Paths

Last Time: Ford Fulkerson (integer capacities) needs at most $val(f^*)$ augmenting paths

- Might be tight for <u>some</u> choice of paths

Today: How to make better choices of paths

- widest augmenting path
  - at most $O(m \cdot \ln(v^*))$ paths
  - at most $O(m^2 \cdot \log n \cdot \log v^*)$ time
- shortest augmenting path

Shortest Augmenting Path

- Shortest augmenting path

  path with fewest edges
  (independent of capacities)

- Can find the shortest augmenting path in time $O(m)$ using breadth-first search

Shortest Augmenting Path

- **Theorem:** Shortest augmenting path terminates after at most $mn/2$ augmenting paths
  - No dependence on capacities! Even for irrational capacities
- **Proof Sketch (Full Proof is Challenging):**

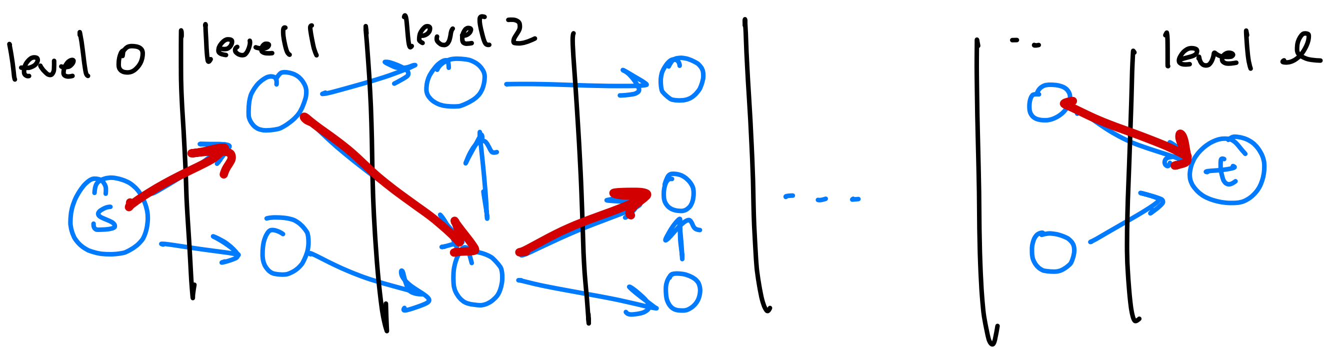
Key Claim

- No edge can be saturated and then unsaturated more than $n/2$ times

Shortest Augmenting Path

- **Key Claim:** No edge reappears more than $n/2$ times
- **Proof Sketch (Full Proof is Challenging):**

- Let G_i be the residual graph after i paths ($G_0 = G$)
- "Run breadth-first search on G_i"



- $\text{level}_i(v)$ = length of the shortest path from s to v in G_i
  1. No edge from level k to $k+2$ or higher
  2. Shortest paths always go forward in level

Shortest Augmenting Path

- **Key Claim:** No edge reappears more than $n/2$ times
- **Proof Sketch (Full Proof is Challenging):**

- Let G_i be the residual graph after i paths $(G_0 = G)$
- $\text{level}_i(v)$ = length of the shortest path from s to v in G_i

Claim 1: For all v and all i $\text{level}_{i-1}(v) \leq \text{level}_i(v)$

Shortest Augmenting Path

- **Key Claim:** No edge reappears more than $n/2$ times
- **Proof Sketch (Full Proof is Challenging):**

- Let G_i be the residual graph after i paths $(G_0 = G)$
- $\text{level}_i(v)$ = length of the shortest path from s to v in G_i

Claim 2: If edge (u,v) "disappears" from G_i but "reappears" in graph G_{j+1} then $\text{level}_{j+1}(u) \geq \text{level}_i(u) + 2$

Shortest Augmenting Path

- **Key Claim:** No edge reappears more than $n/2$ times
- **Proof Sketch (Full Proof is Challenging):**

- Let G_i be the residual graph after i paths $(G_0 = G)$
- $\text{level}_i(v)$ = length of the shortest path from s to v in G_i

Claim 2: If edge (u,v) "disappears" from G_i but "reappears" in graph G_{j+1} then $\text{level}_{j+1}(u) \geq \text{level}_i(u) + 2$

- (u,v) disappeared because it was on the i^{th} aug path

$$s \rightsquigarrow u \to v \rightsquigarrow t$$

$\text{level}_i(v) = \text{level}_i(u) + 1$ (because we used a shortest path)

- (u,v) reappeared because (v,u) was on the $(j+1)^{st}$ aug path

$$s \rightsquigarrow v \to u \rightsquigarrow t$$

$$\text{level}_j(u) = \text{level}_j(v) + 1 \overset{(\text{Clm } 1)}{\geq} \text{level}_i(v) + 1 = \text{level}_i(u) + 2$$

Choosing Good Augmenting Paths

Summary

- **Last Class:** Can solve maximum flow in time $O(m \cdot v^*)$
  - Can be very slow when capacities are large
  - Cannot be improved if we allow arbitrary augmenting paths

- **Today:** Improving running time by choosing better paths
  - **Widest Augmenting Path:** $O(m \cdot \log v^*)$
  - **Shortest Augmenting Path:** $O(m^2 n)$

- **Still actively studied!**
  - Can solve maximum flow in $O(mn)$ using augmenting path* algos
  - **Recent Breakthrough:** Can solve maximum flow in time* $m^{1+o(1)}$

- **Later On:** Using maximum-flow as a building block for solving many more problems