CS 7800: Advanced Algorithms

Class 8: Network Flow II

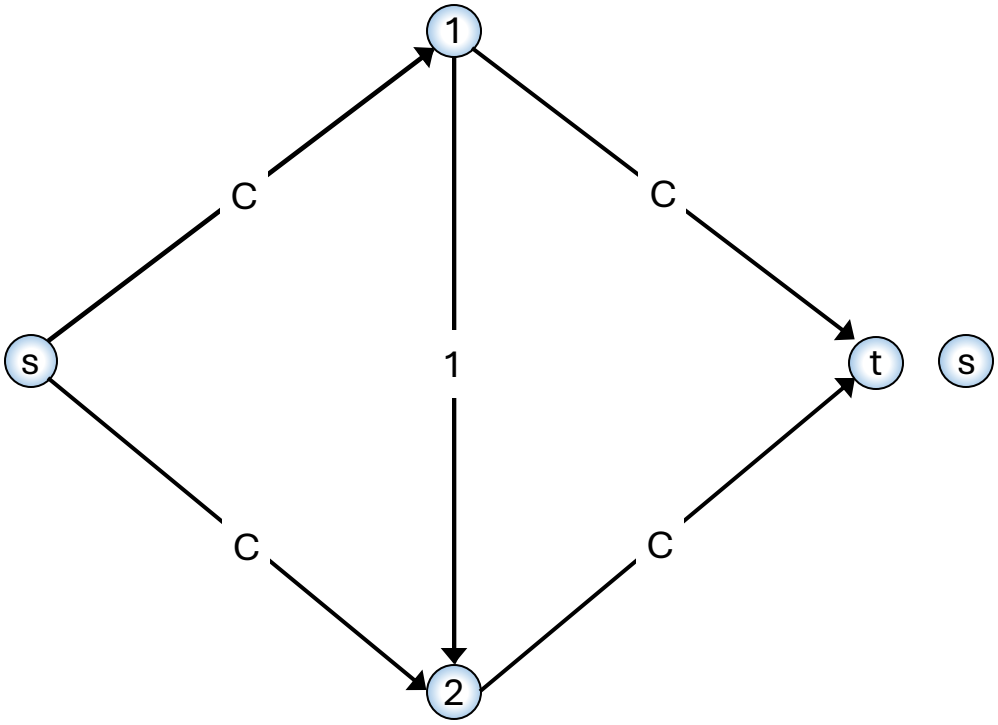
- Choosing Good Augmenting Paths

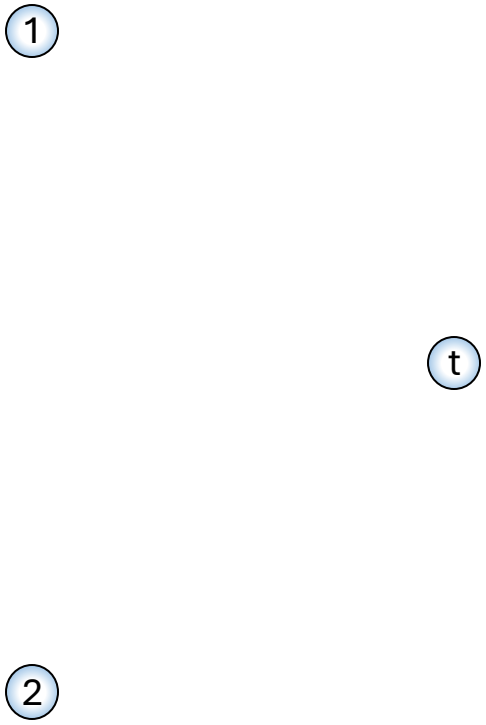
Jonathan Ullman

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





Choosing Good Augmenting Paths

Widest Augmenting Path

- Widest augmenting 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: $\leq v^*$

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:** $B \geq \frac{v^*}{m}$
- **Proof:**

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:
- Flow remaining in G_f:
- # of aug paths:

Choosing Good Augmenting Paths

Shortest Augmenting Path

- Shortest augmenting path

- 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!
- **Proof Sketch (Full Proof is Challenging):**

Shortest Augmenting Path

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

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