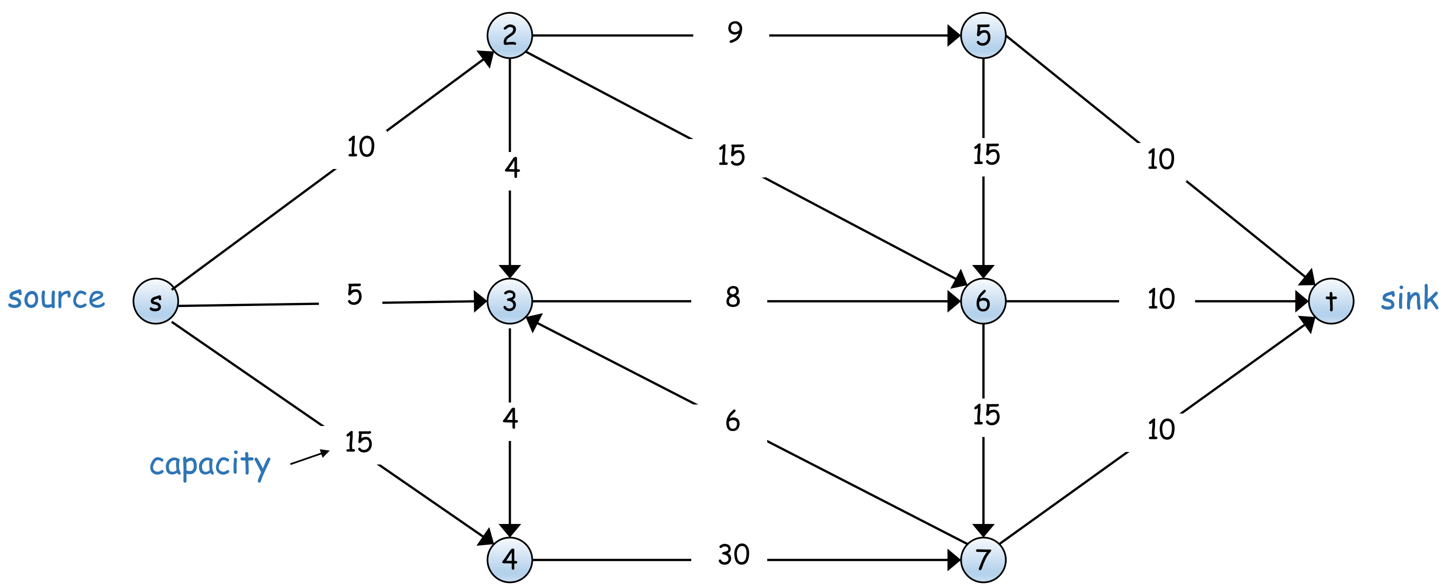
CS4800: Algorithms & Data
Jonathan Ullman

Lecture 17:
- Network Flow
  - Choosing Good Augmenting Paths

Mar 20, 2018

Recap

- Directed graph $G = (V, E)$
- Two special nodes: source s and sink = t
- Edge capacities $c(e)$



Recap

non-negativity

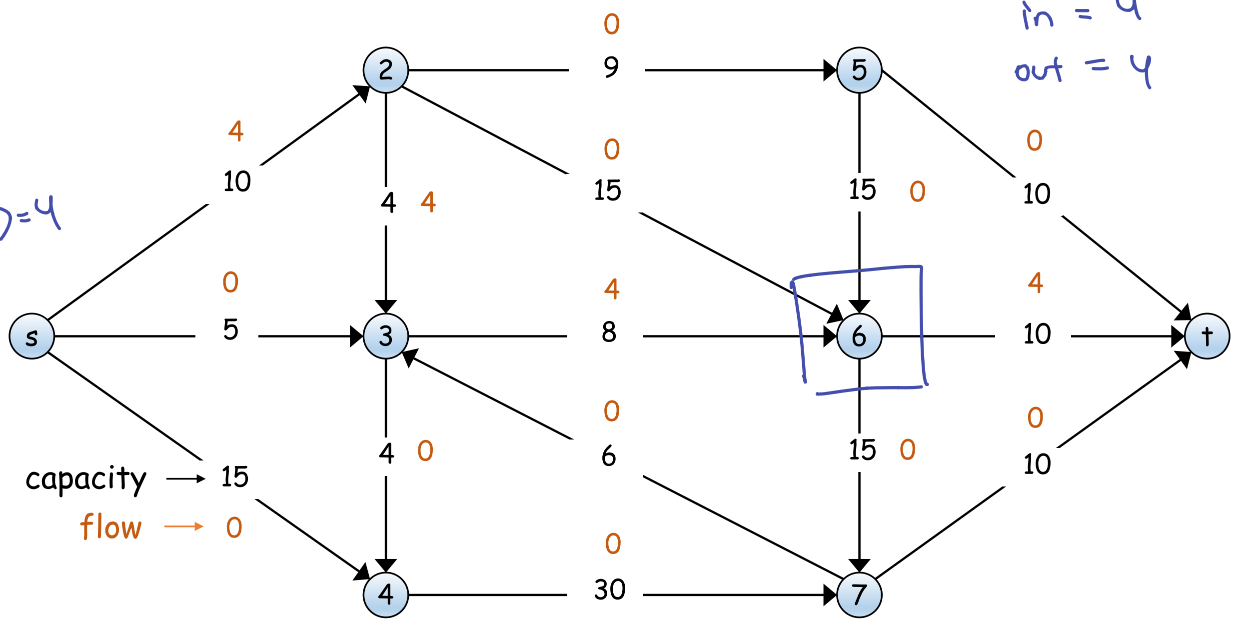
- An s-t flow is a function $f(e)$ such that
  - For every $e \in E$, $\boxed{0 \leq f(e)} \leq c(e)$ (capacity)
  - For every $v \in E$, $\sum_{e \text{ in to } v} f(e) = \sum_{e \text{ out of } v} f(e)$ (conservation)

    (except s,t)

- The value of a flow is $val(f) = \sum_{e \text{ out of } s} f(e)$

in = 4
out = 4

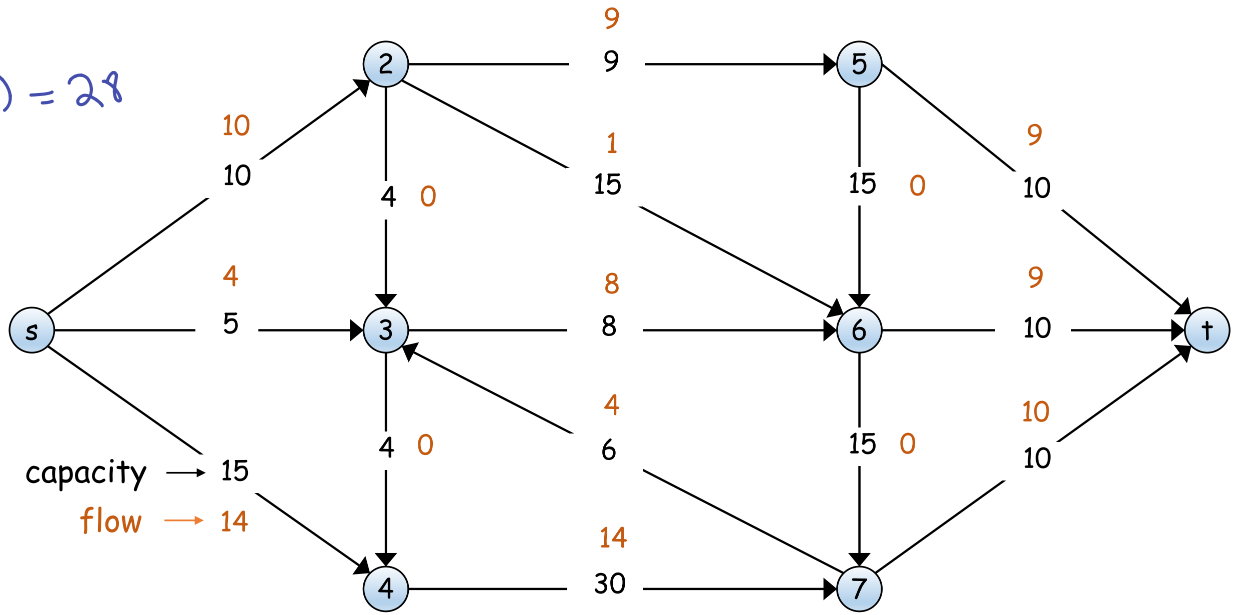
val(f)=4



Recap

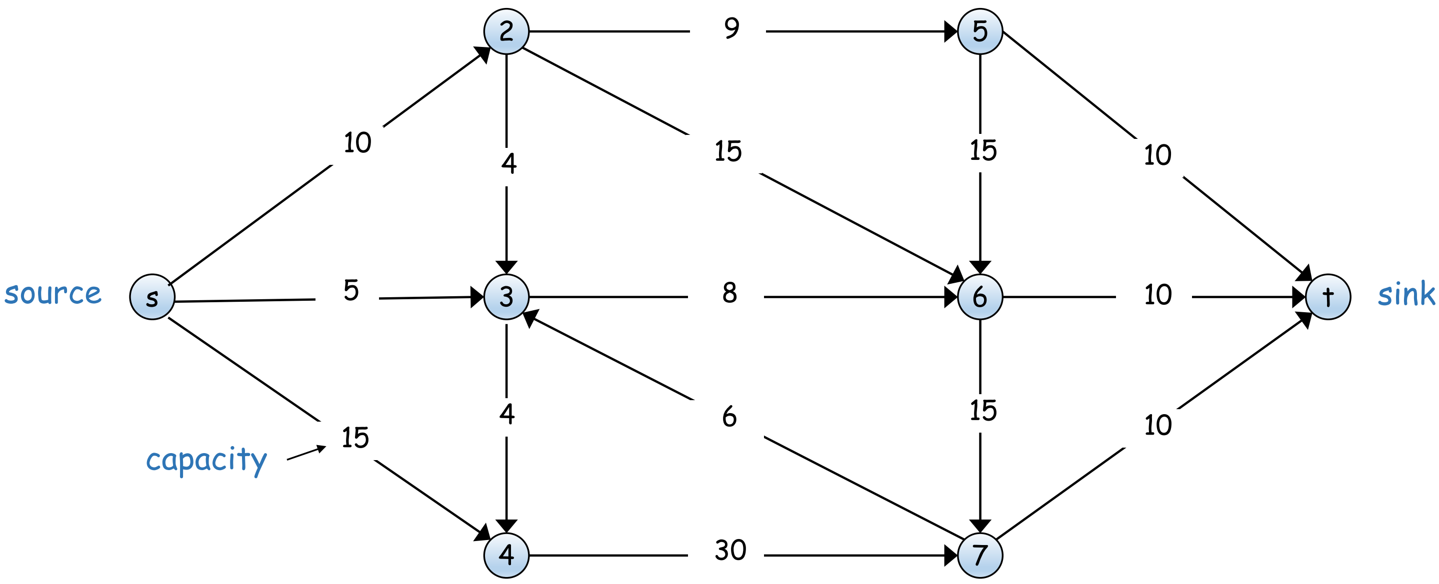
- MaxFlow = Find an s-t flow of maximum value

val(f) = 28



Recap

- An s-t cut is a partition (A, B) of V with $s \in A$ and $t \in B$
- The capacity of a cut (A,B) is $cap(A, B) = \sum_{e \text{ out of } A} c(e)$



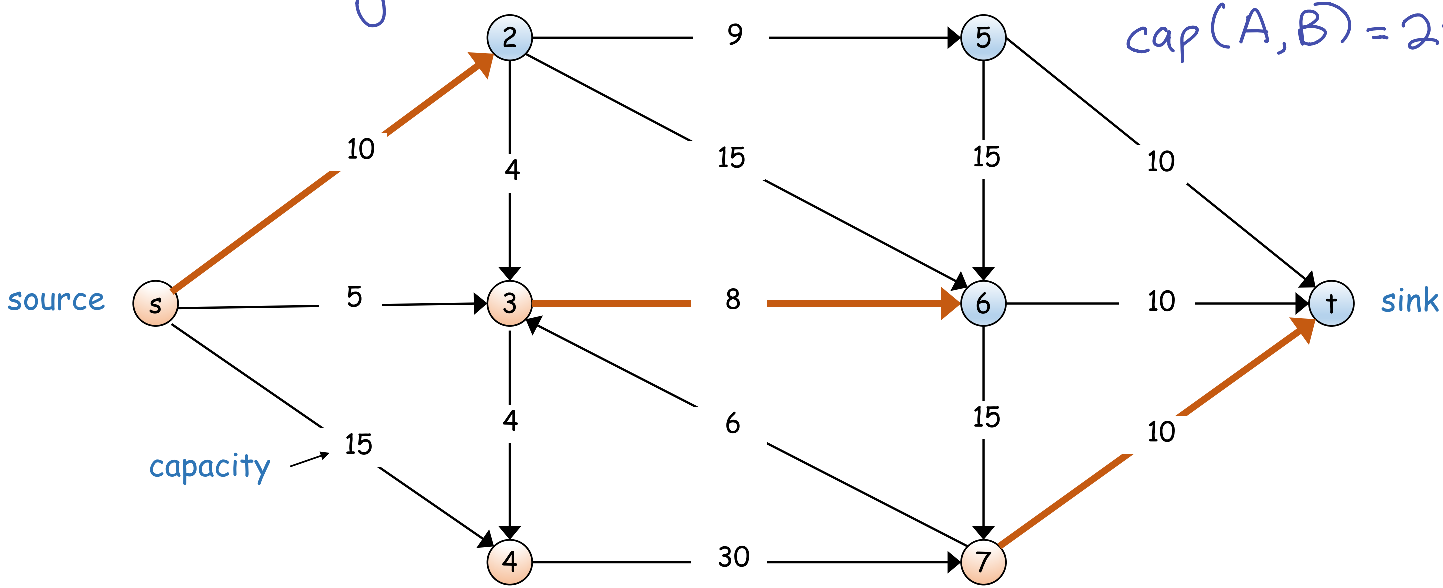
Recap

- MinCut: Find an s-t cut of minimum capacity

$$cap(A,B) = \sum_{e \text{ from } A \text{ to } B} c(e)$$

- Weak Duality Theorem: For any flow f, cut (A,B), $val(f) \le cap(A,B)$

$cap(A,B) = 28$

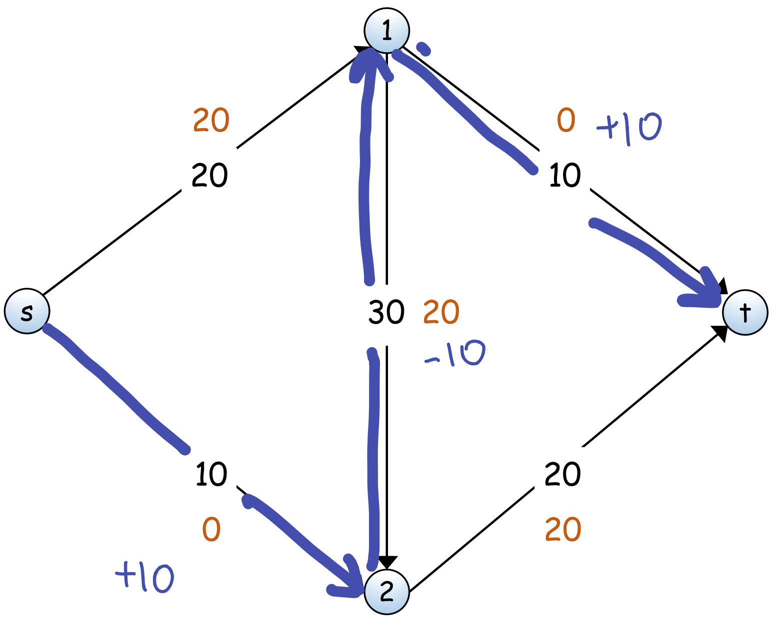


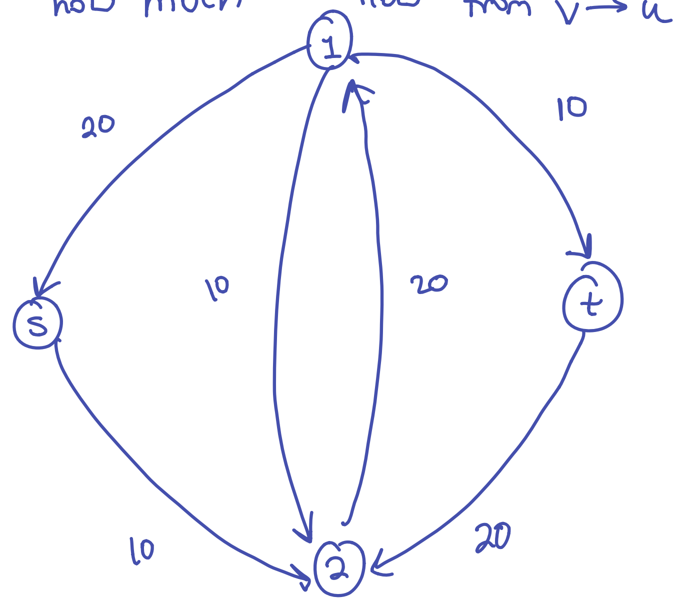
Ford-Fulkerson Algorithm

An s-t path s.t. you can send more flow along the path.

- Start with $f(e) = 0$ for all $e \in E$
- While
  - There is an $\boxed{\text{augmenting path}}$ P in the $\boxed{\text{residual graph}}$ G_f
- Augment flow along the path P

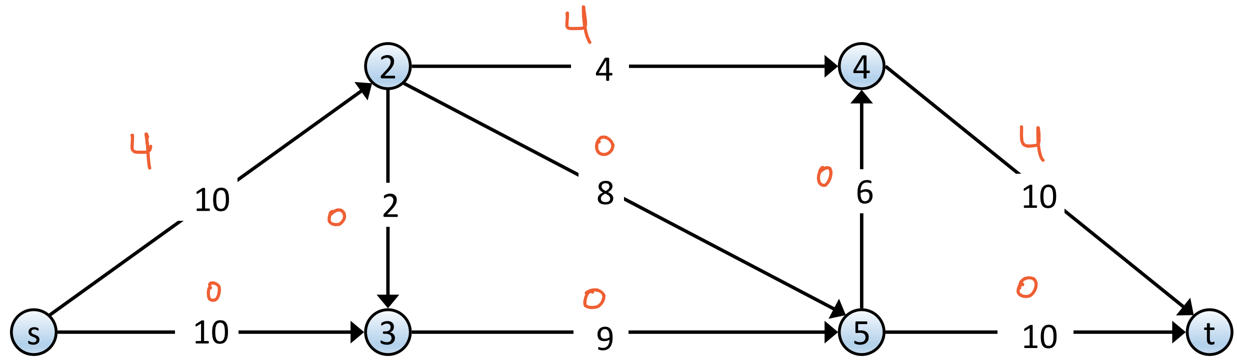
records how much +flow from u→v
how much − flow from v→u



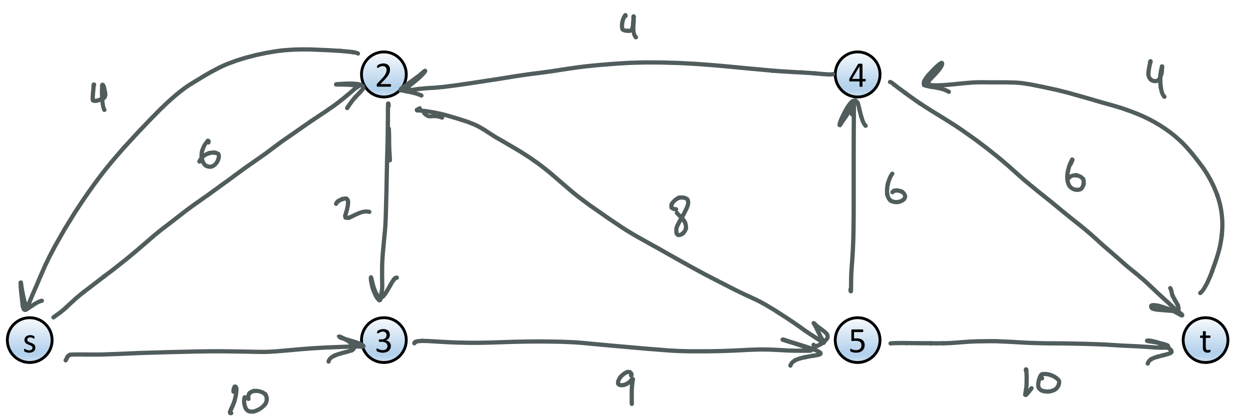


Ford-Fulkerson Demo

G:

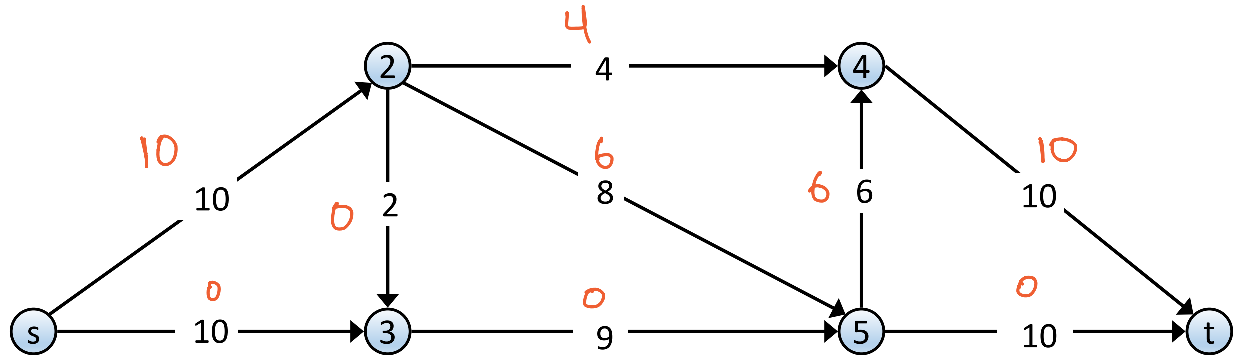


G_f:

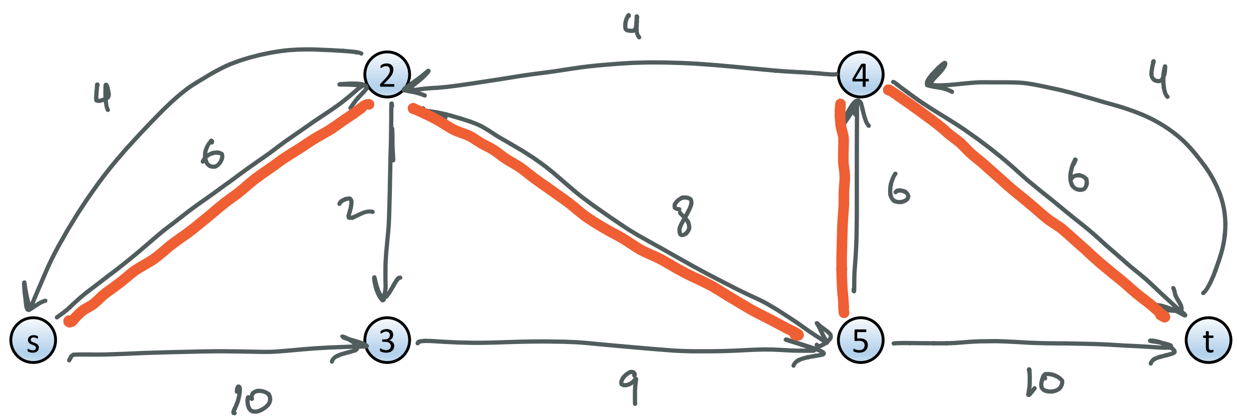


Ford-Fulkerson Demo

G:

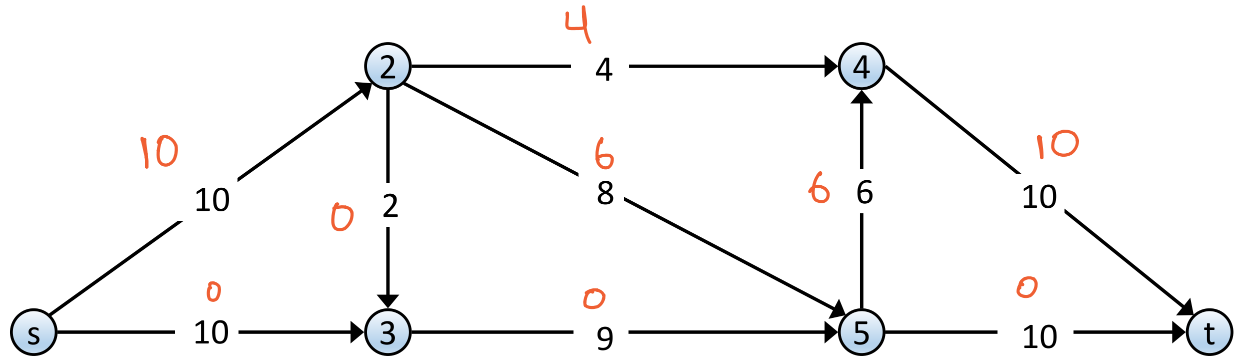


G_f:

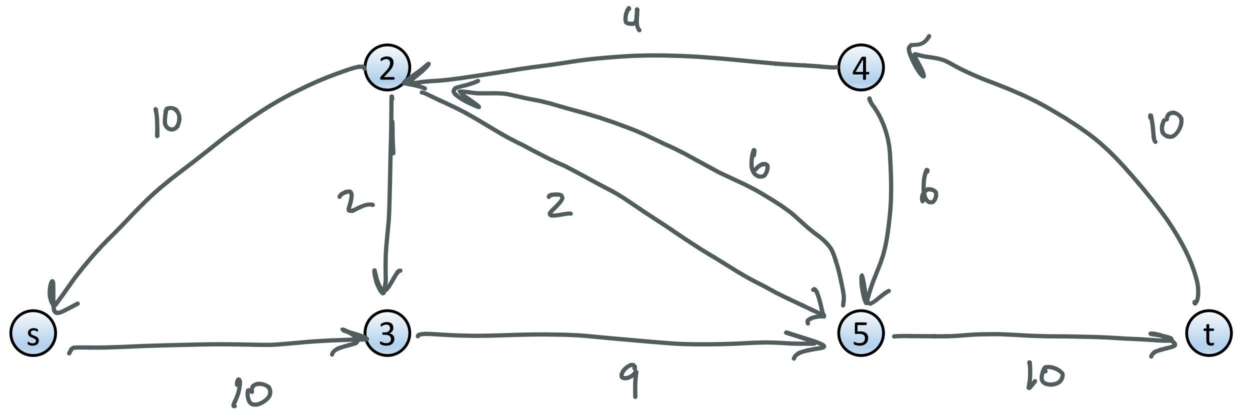


Ford-Fulkerson Demo

G:

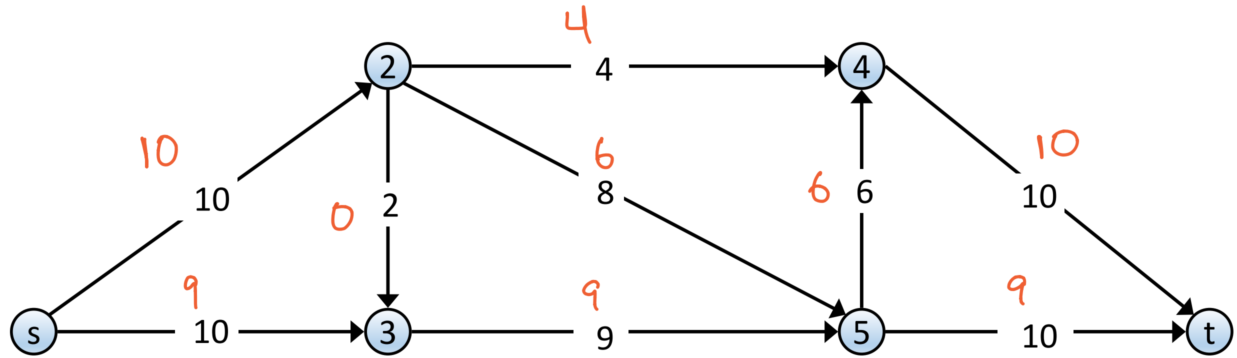


G_f:

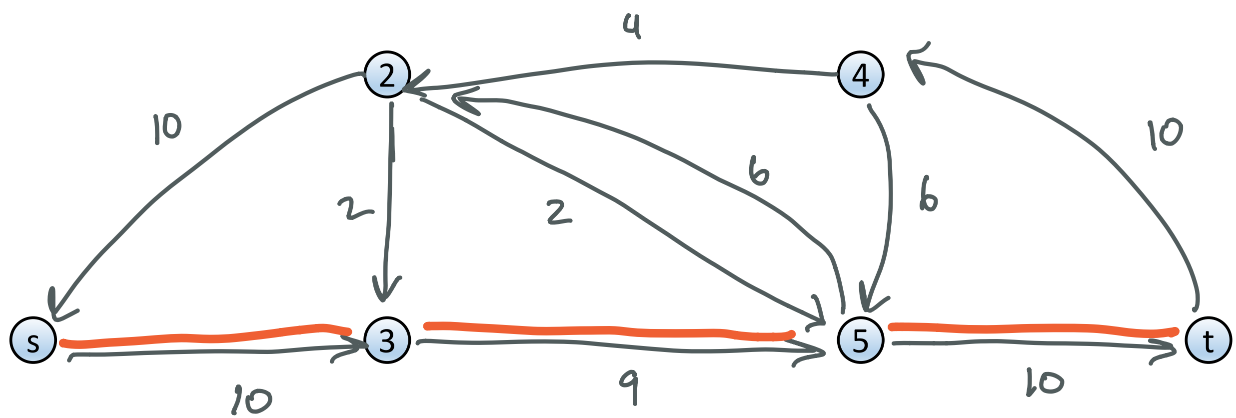


Ford-Fulkerson Demo

G:



G_f:



Ford-Fulkerson Demo

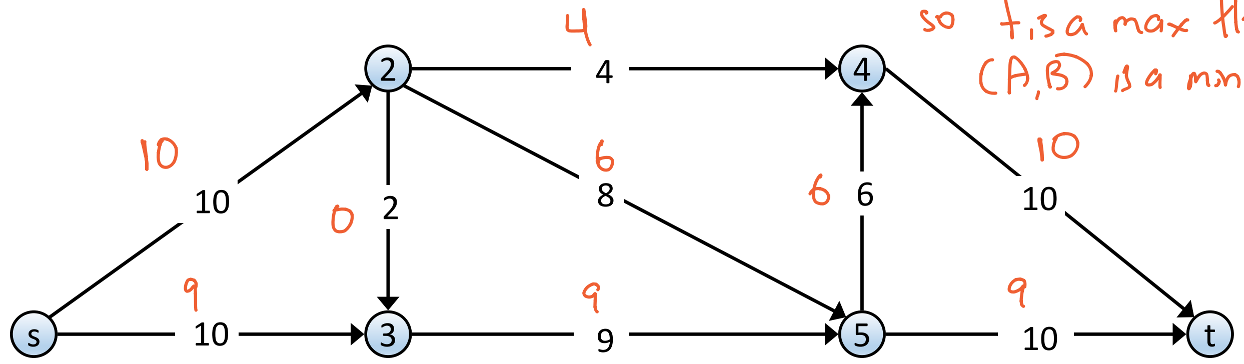
$A = \{s, 3\}$ $B = \{2, 4, 5, t\}$

$val(f) = 19$ $cap(A, B) = 19$

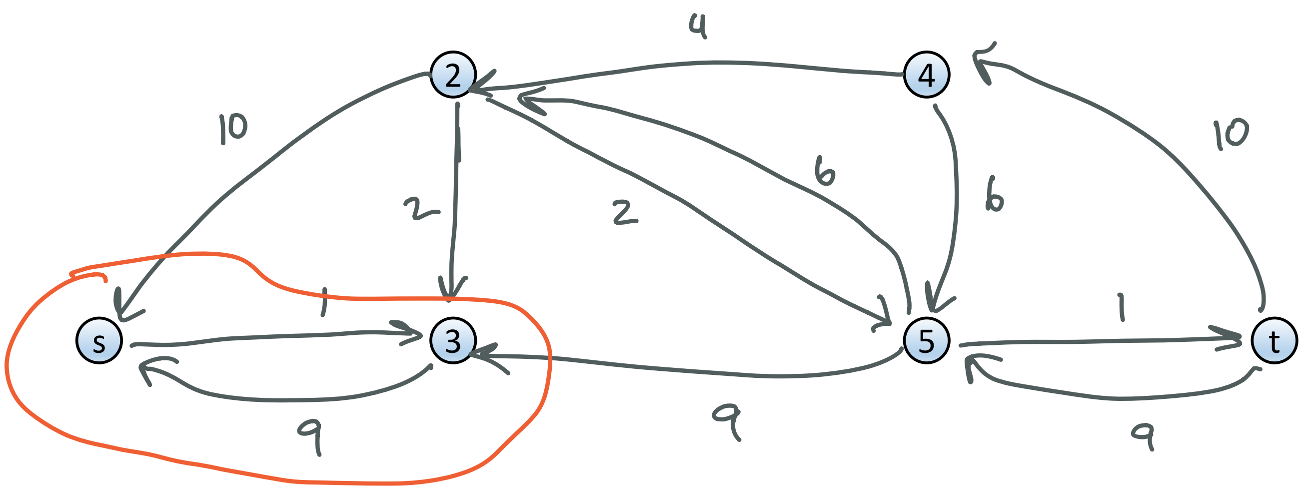
so f is a max flow

(A, B) is a min cut

G:



G_f:



Summary

- The Ford-Fulkerson Algorithm solves maximum s-t flow
  - Running time is O(m) per augmentation step
  - val(f*) ~~O(val(f*))~~ augmentinations in any graph with integer capacities
  - O(n) augmentations in graphs with unit capacities

    (total time O(mn))

- MaxFlow-MinCut Theorem: The value of the max s-t flow equals the capacity of the min s-t cut
  - If f^* is a max flow, the nodes reachable from s in G_{f^*} are a min cut
  - Given a max flow, can find a min cut in time O(n+m) via BFS

Recap

- Two Problems:
  - MaxFlow: given a network G=(V,E), capacities c(e), source s, sink t, find the s-t flow of maximum value
  - MinCut: given a network G=(V,E), capacities c(e), source s, sink t, find the s-t cut of minimum value

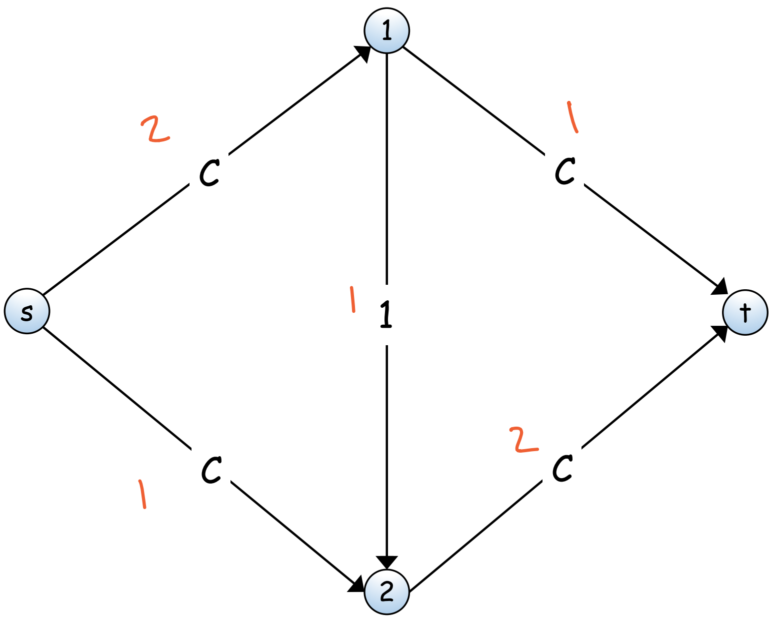
- Ford-Fulkerson Algorithm:
  - Start with the empty flow f(e) = 0
  - While there is an augmenting path P in the residual graph G_f, increase f along the path

Ford-Fulkerson Algorithm

$\text{val}(f^*) = 2C$

$\#$ of augmentations $= 2C$

- Start with $f(e) = 0$ for all $e \in E$
- While
  - There is an augmenting path P in the residual graph G_f
- Augment flow along the path P



$s \xrightarrow{+1} 1 \xrightarrow{+1} 2 \xrightarrow{+1} t \quad +1$

$s \xrightarrow{+1} 2 \xrightarrow{-1} 1 \xrightarrow{+1} t \quad +1$

$s \xrightarrow{+1} 1 \xrightarrow{+1} 2 \xrightarrow{+1} t$

Ideas:
- Choose the "fattest path"
- Choose the "shortest path"

Choosing Good Paths

- Last time: arbitrary augmenting paths
  - If FF terminates, it outputs a maximum flow
- Today: clever augmenting paths
  - Maximum-capacity augmenting path (“fattest path”)
  - Shortest augmenting paths (“shortest path”)

Fattest Augmenting Path

Fattest Augmenting Path

- Maximum-capacity augmenting path

$$\max_{s\text{-}t \text{ paths } P \text{ in } G_f} \ \min_{e \in P} \ c(e)$$

- Can find the maximum-capacity augmenting path in time $O(m \log n)$ using a variant of Prim's or Kruskal's MST
  - Exercise for the reader

Fattest 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:

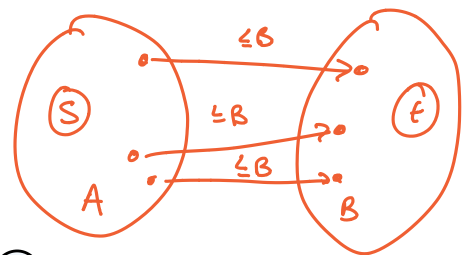
Fattest Augmenting Path

capacity of fattest path

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

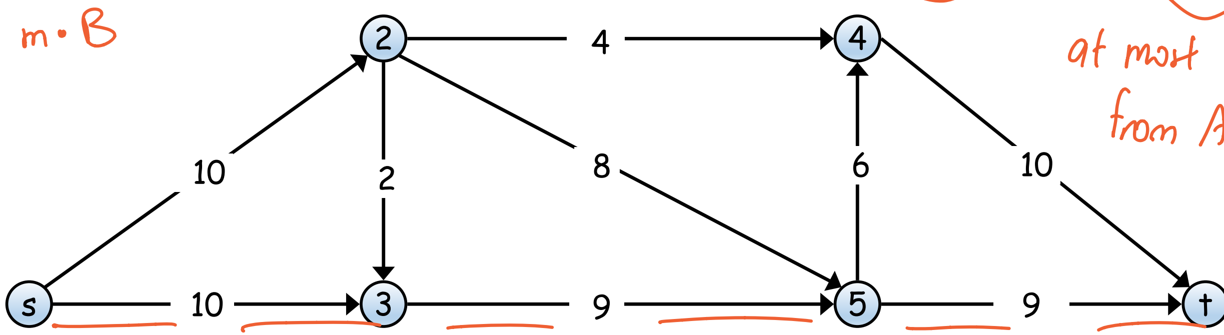
If there is no path w/ cap > B then $v^* \leq B \cdot m$

- Consider G' containing only edges w/ capacity > B
- s-t must be disconnected in this graph



- $cap(A,B) \leq m \cdot B$
- $v^* \leq m \cdot B$

at most m edges from A to B



Fattest 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: $\geq v^*/m$
- Flow remaining in G_f: $\leq v^* - \frac{v^*}{m} = \left(1 - \frac{1}{m}\right) \cdot v^*$
- # of aug paths: $\leq m \cdot \ln(v^*)$

After T augmentations there is $v^* \cdot \left(1 - \frac{1}{m}\right)^T \leq v^* \cdot e^{-T/m}$

If $T > m \cdot \ln(v^*)$ then flow remaining is < 1

Choosing Good Paths

- Last time: arbitrary augmenting paths
  - If FF terminates, it outputs a maximum flow

- Today: clever augmenting paths
  - Maximum-capacity augmenting path ("fattest path")
    - $\leq m \ln v^*$ augmenting paths (assuming integer capacities)
    - $O(m^2 \ln n \ln v^*)$ total running time
    - See KT for a slightly faster variant ("fat enough path")

  - Shortest augmenting paths ("shortest path")

Shortest Augmenting Path

Shortest Augmenting Path

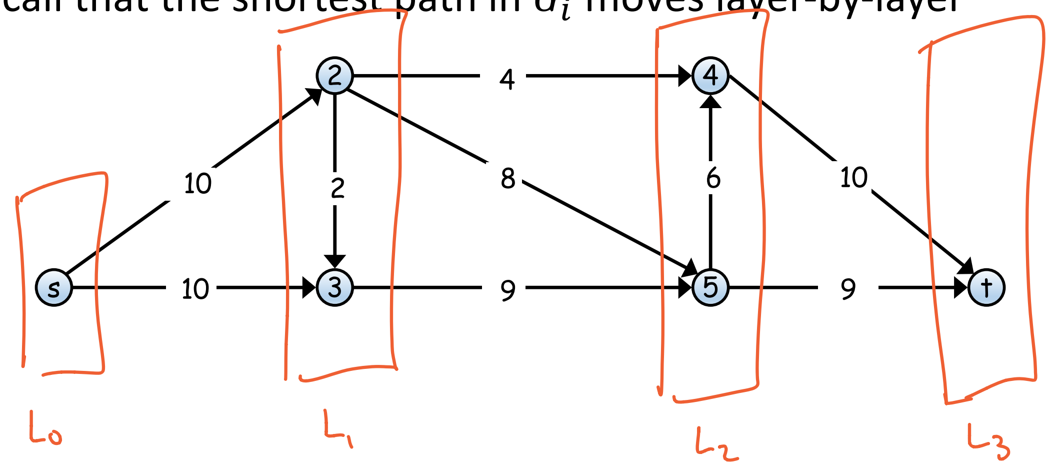
- Find the augmenting path with the fewest hops
  - Can find shortest augmenting path in O(m) time using BFS

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

- Warning: proof is very tricky (you will not be tested on it)

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
  - $f_0 = 0$ and $G_0 = G$
- 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

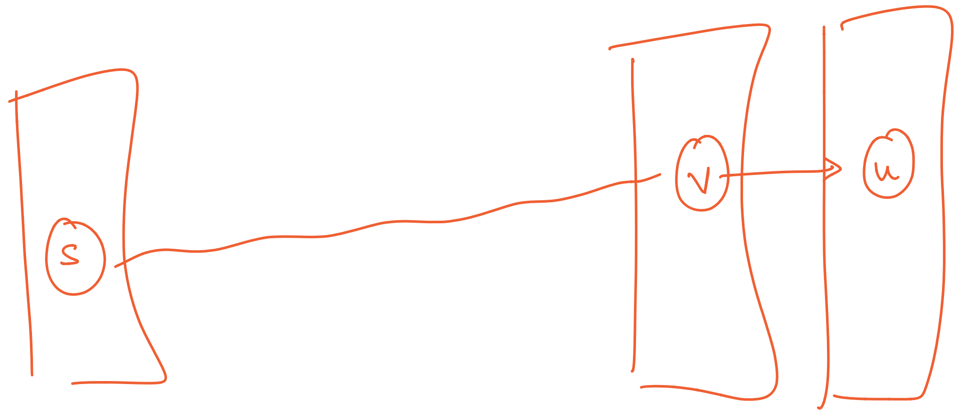
- Claim 1: for every $v \in V$ and every i, $L_{i+1}(v) \geq L_i(v)$
  - Obvious for $v = s$ because $L_i(s) = 0$
  - Suppose for the sake of contradiction that $L_{i+1}(v) < L_i(v)$
    - Let v be the smallest such node

    after aug i, v gets closer to s
  - Let $s \rightsquigarrow u \to v$ be a shortest path in G_{i+1}
    - layer-by-layer: By optimality of the path, $L_{i+1}(v) = L_{i+1}(u) + 1$
    - by choice of v: By assumption, $L_{i+1}(u) \geq L_i(u)$

    $L_{i+1}(v) = L_{i+1}(u) + 1$
    $\geq L_i(u) + 1$
  - Two Cases:
    - $(u, v) \in G_i$, so $L_i(v) \leq L_i(u) + 1$

      $L_{i+1}(v) = L_{i+1}(u) + 1 \geq L_i(u) + 1 \geq L_i(v)$
    - $(u, v) \notin G_i$, so (v, u) was in the i-th path, so $L_i(v) = L_i(u) - 1$

      $L_{i+1}(v) = L_{i+1}(u) + 1 \geq L_i(u) + 1 = L_i(v) + 2$



$L_i(s)=0$ $L_i(v)$ $L_i(u)=L_i(v)+1$

Because augmenting path is the shortest path.

Shortest Augmenting Path

- Claim 2: If an edge $u \rightarrow v$ disappears from G_i and reappears in G_{j+1} then $L_j(u) \geq L_i(u) + 2$ shortest paths go layer-by-layer
  - $u \rightarrow v$ is on the i-th augmenting path, $L_i(v) = L_i(u) + 1$
  - $v \rightarrow u$ is on the j-th augmenting path, $L_j(u) = L_j(v) + 1$
  - By Claim 1: $L_j(v) \geq L_i(v)$

$$L_j(u) = L_j(v) + 1 \geq L_i(v) + 1 = L_i(u) + 1 + 1$$

- Every augmentation causes $\geq$ one edge to disappear
- At most m edges, so at most $\frac{mn}{2}$ disappearances
- At most $\frac{mn}{2}$ augmenting paths

Choosing Good Paths

- Last time: arbitrary augmenting paths
  - If FF terminates, it outputs a maximum flow

- Today: clever augmenting paths
  - Maximum-capacity augmenting path ("fattest path")
    - $\leq m \ln v^*$ augmenting paths (assuming integer capacities)
    - $O(m^2 \ln n \ln v^*)$ total running time
    - See KT for a slightly faster variant ("fat enough path")

  - Shortest augmenting paths ("shortest path")
    - $\leq \frac{mn}{2}$ augmenting paths (for any capacities)
    - $O(m^2 n)$ total running time

Summary

- The Ford-Fulkerson Algorithm solves maximum s-t flow
  - Different choices of augmenting paths give different running times

- Still an active area of research!

$O(m^{2/5} n^{4/5} \log^2 v)$ $O(m^{10/7})$

Can solve maxflow
in time
$O(mn)$

- Can solve many problems efficiently via reductions to the maximum flow or minimum cut problems