OntarioTech Science

Kourosh Davoudi
kourosh@ontariotechu.ca

Lecture 12: Flow Graphs

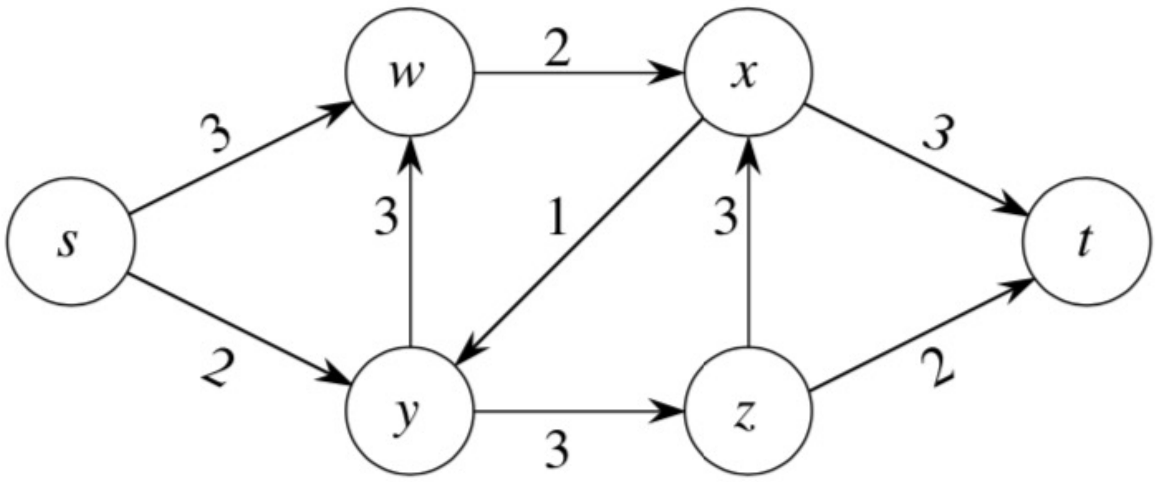
CSCI 3070U: Design and Analysis of Algorithms

Learning Outcomes

- Flow Networks:
  - Concepts and Foundations
  - Algorithms

Flow Network

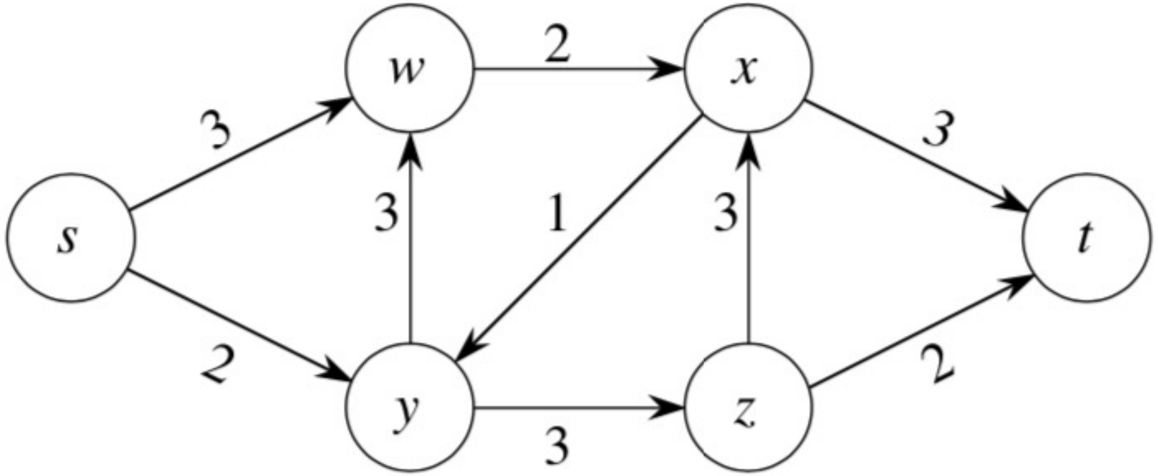
- A flow network $G = (V, E)$ is a directed graph in which each edge $(u, v) \in E$ has a nonnegative capacity $c(u, v) \geq 0$



Flow Network

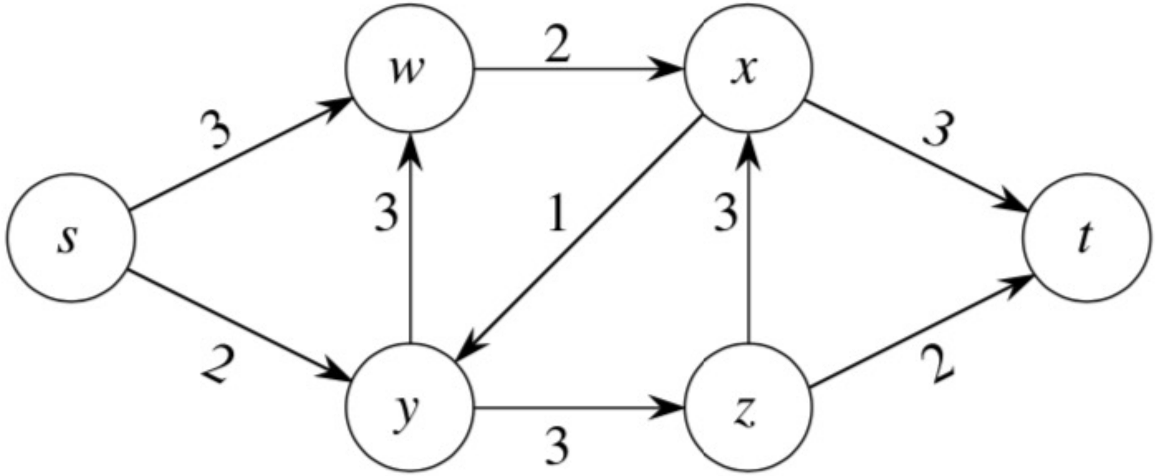
- We distinguish two vertices in a flow network:
  - Source s
  - Sink t
- We assume that each vertex lies on a path from source to sink
  - For all $v \in V$, we have a path

$$s \rightsquigarrow v \rightsquigarrow t$$



Flow Networks Properties

- A flow network $G = (V, E)$ is a directed graph in which each edge $u \in E$ has a nonnegative capacity $c(u, v) \geq 0$
- If $(u, v) \in E$, then $(v, u) \notin E$
- If $(v, u) \notin E$, then $c(v, u) = 0$



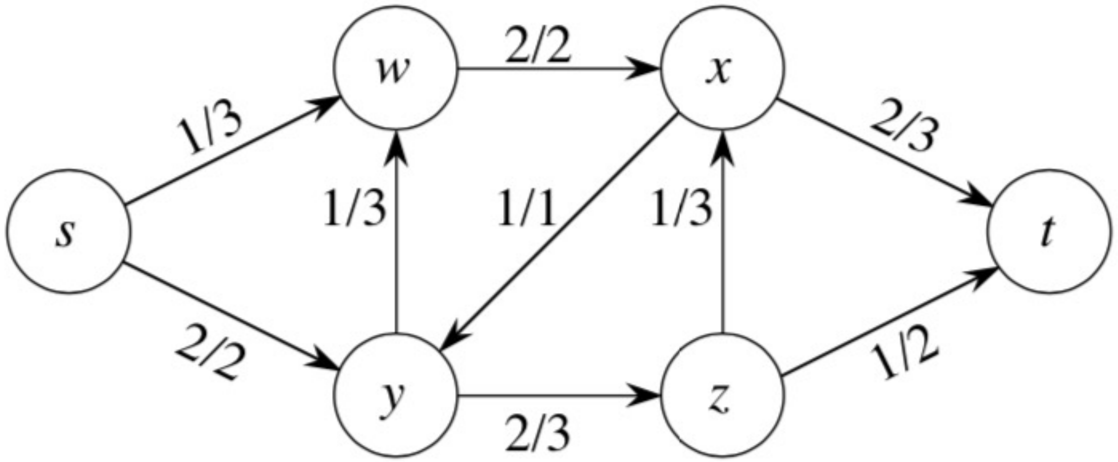
Flow Definition

- Flow is a function $f : V \times V \to \mathbb{R}$ satisfying:
  - Capacity Constraint:

    For all $u, v \in V, 0 \leq f(u, v) \leq c(u, v)$.

  - Flow Conservation:

    For all $u \in V - \{s, t\}, \underbrace{\sum_{v \in V} f(v, u)}_{\text{flow into } u} = \underbrace{\sum_{v \in V} f(u, v)}_{\text{flow out of } u}$.



Value of Flow

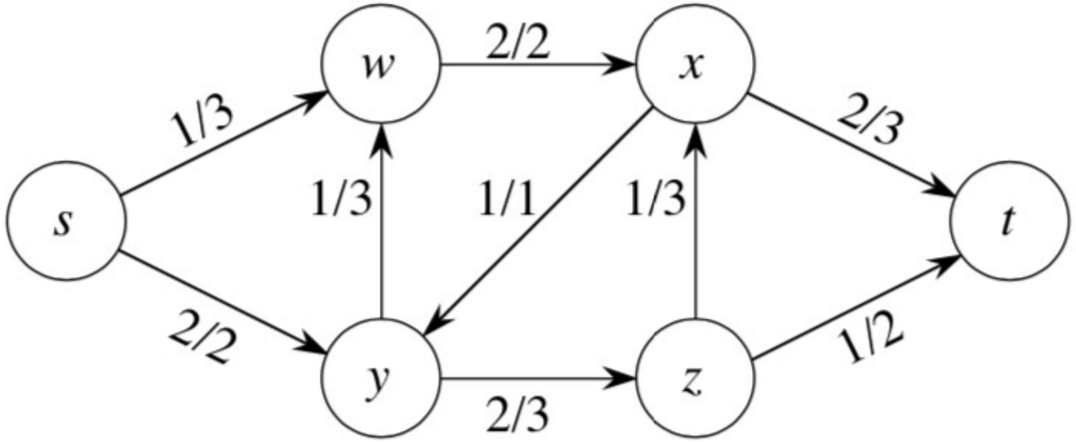
- Value of flow f is defined as follows:

$$\begin{aligned} \textit{Value of flow } f &= |f| \\ &= \sum_{v \in V} f(s,v) - \sum_{v \in V} f(v,s) \\ &= \text{flow out of source} - \text{flow into source} \end{aligned}$$

- Example:

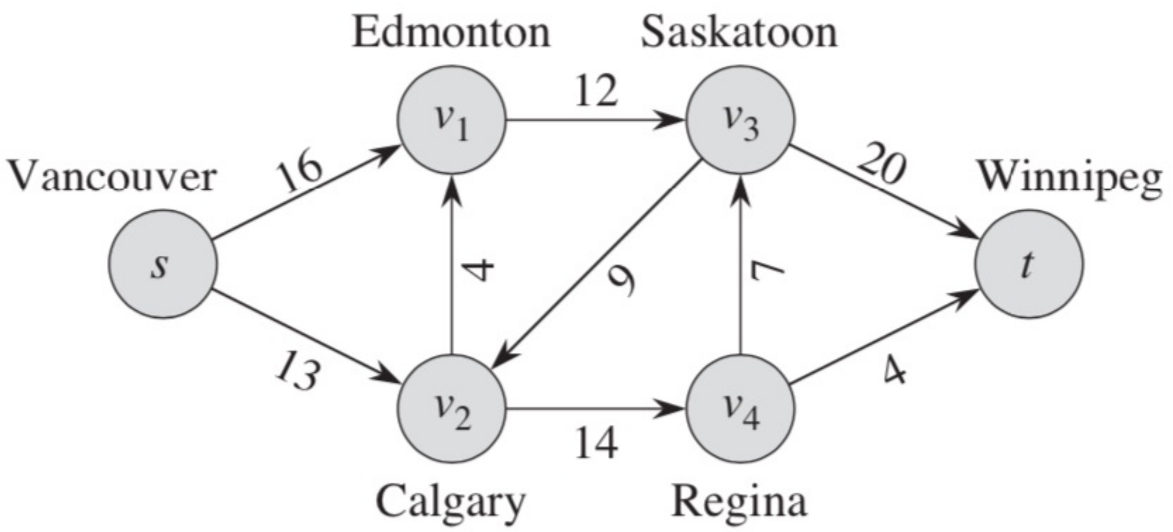
$|f| = ?$

Answer = 3



Maximum Flow Problem

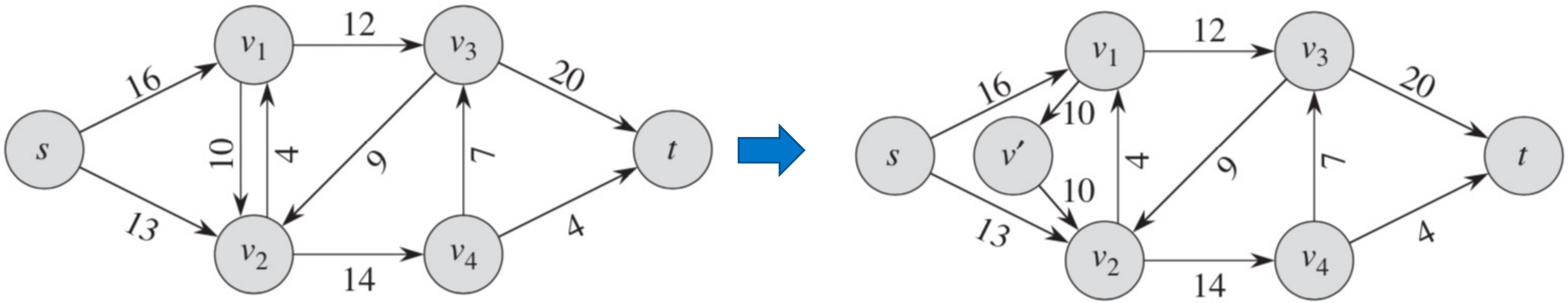
- Given G, s, t, and c, find a flow whose value is maximum



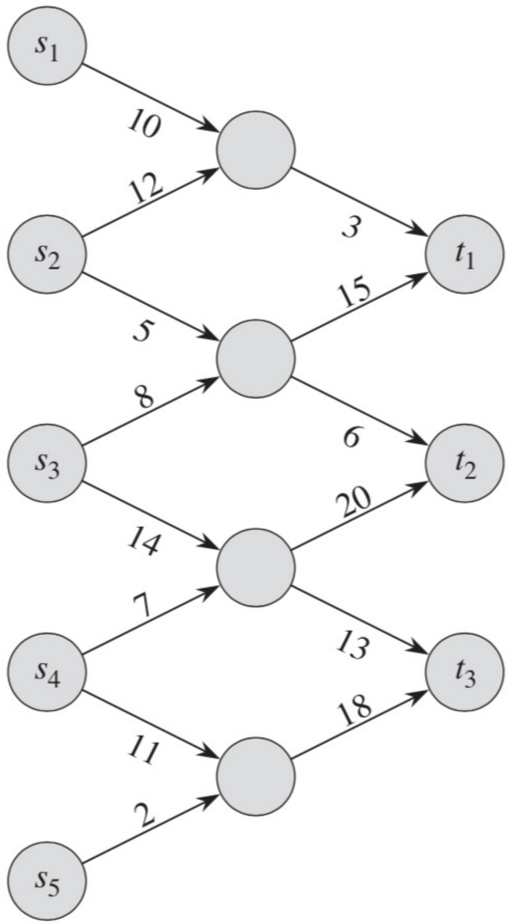
  - Maximum rate of shipping product from Vancouver to Winnipeg through intermediate cities

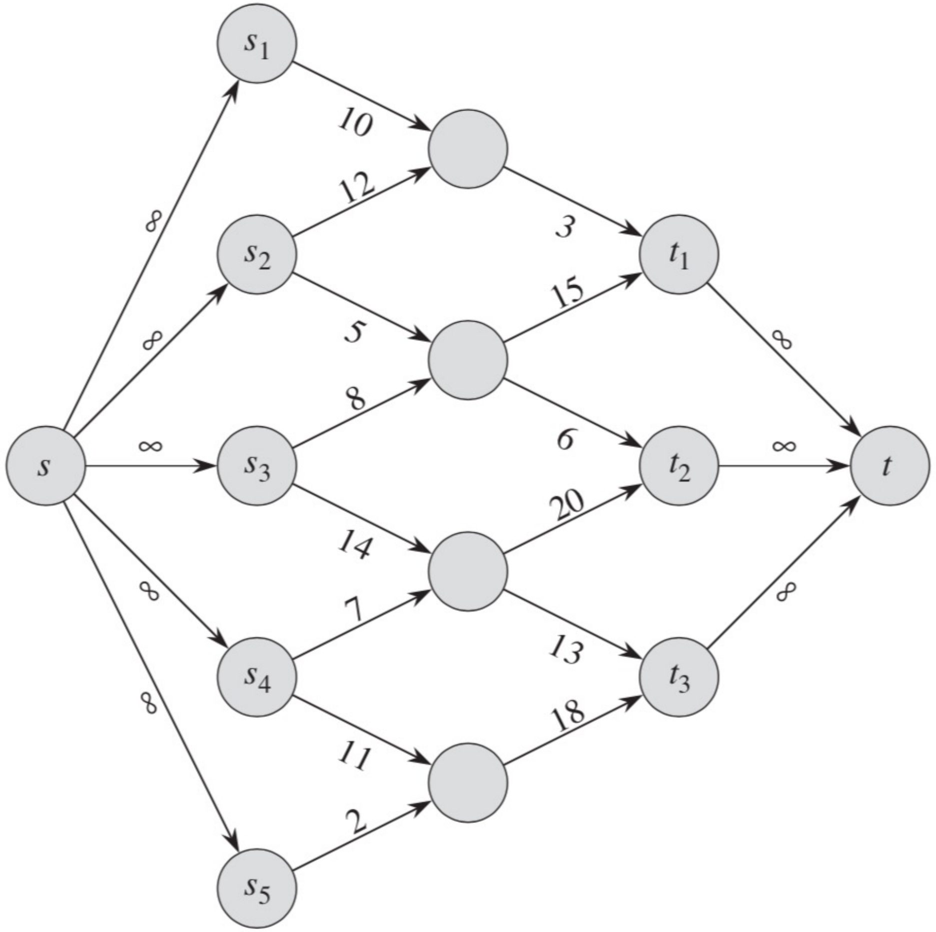
Antiparallel Edges

- Definition of flow network does not allow both (u, v) and (v, u) to be edges. These edges would be antiparallel
- What if we really need antiparallel edges?



Networks with Multiple Sources and Sinks



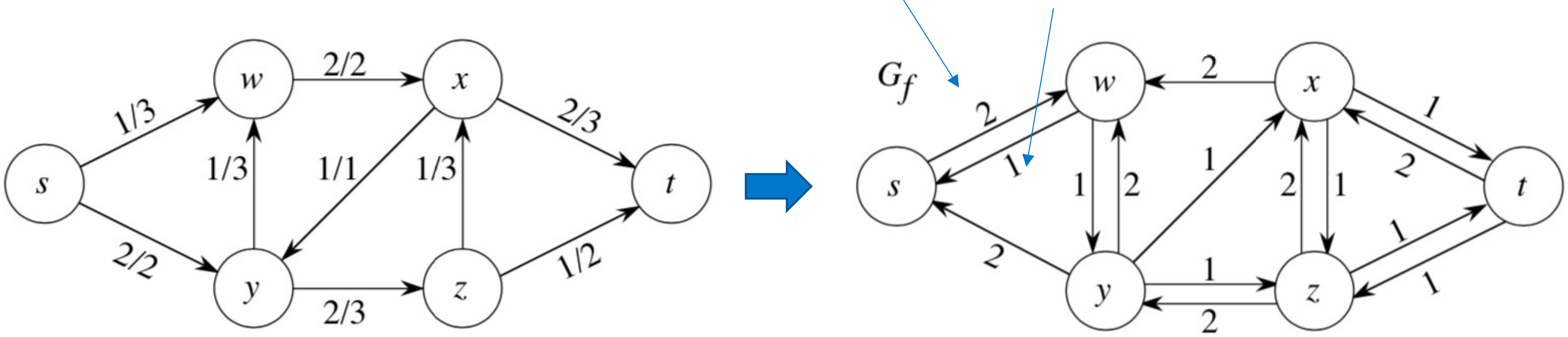


Residual Network

- Given a flow f in network $G = (V, E)$, the residual network of G is G_f

$$c_f(u, v) = c(u, v) - f(u, v)$$

$$c_f(v, u) = f(u, v)$$



$$c_f(u, v) = \begin{cases} c(u, v) - f(u, v) & \text{if } (u, v) \in E, \\ f(v, u) & \text{if } (v, u) \in E, \\ 0 & \text{otherwise}. \end{cases}$$

Augmenting Paths

- Given a flow network $G = (V, E)$ and a flow f, an augmenting path p is a simple path from s to t in the residual network G_f.
    - A simple path is a path without cycle.

$$s \leadsto t$$

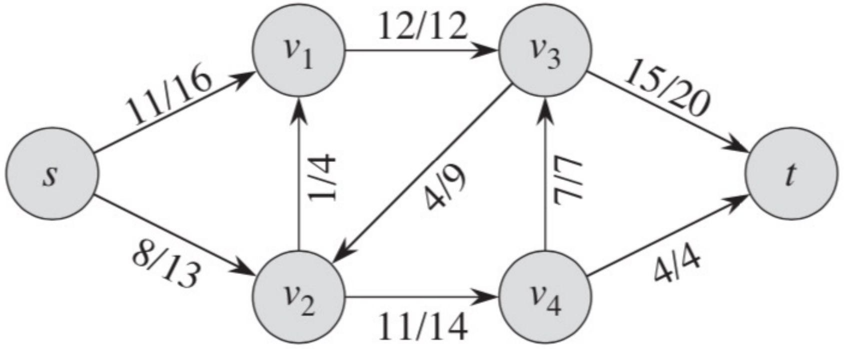
- How much more flow can we push from s to t along augmenting path p?

$$c_f(p) = \min\{c_f(u, v) : (u, v) \text{ is on } p\}$$

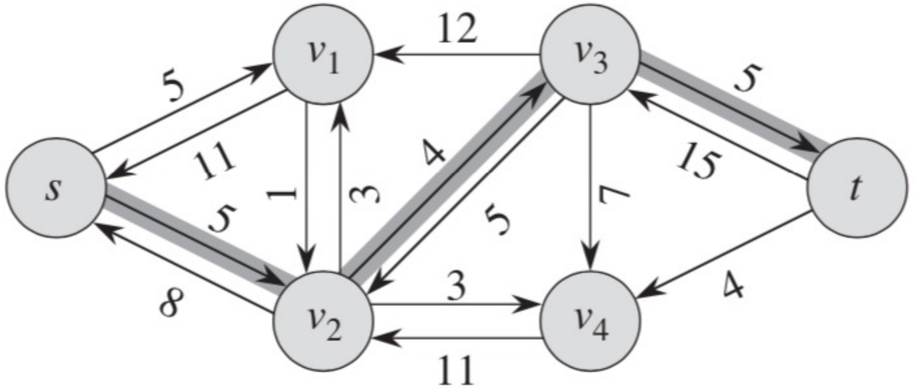
Residual Capacity

Augmenting Paths

- The shaded path is an augmenting path



G_f

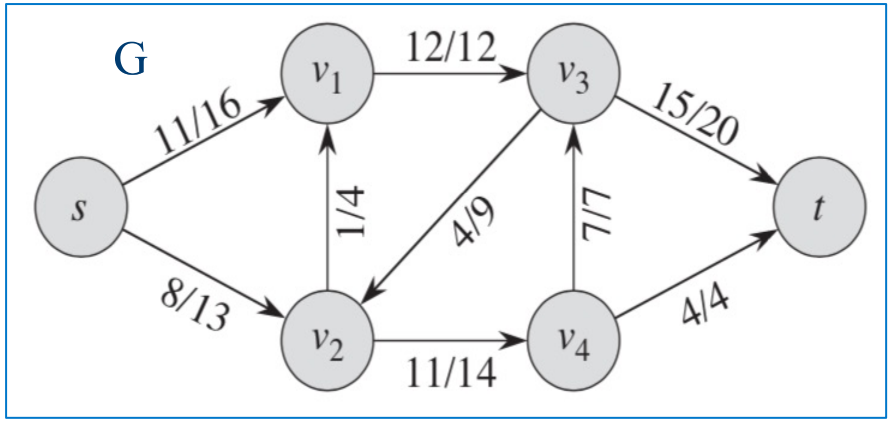


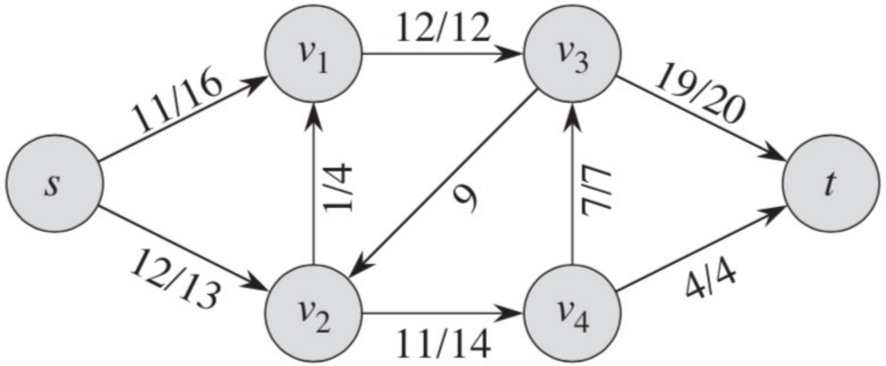
G_f

$$c_f(p) = \min\{c_f(u, v) : (u, v) \text{ is on } p\} = \min\{5, 4, 5\} = 4$$

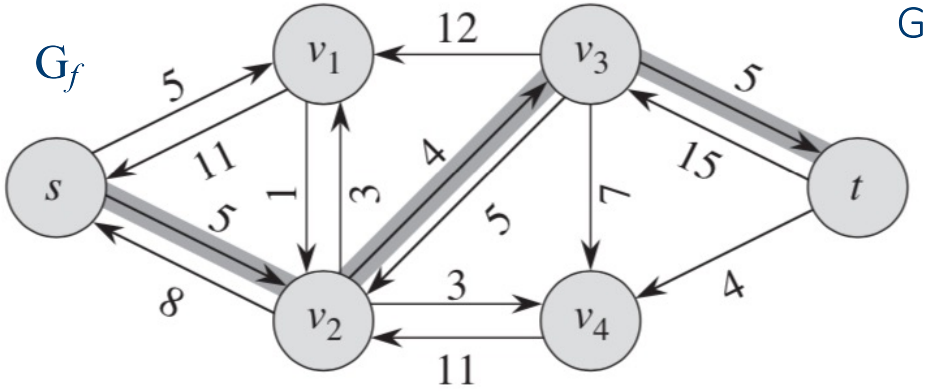
Augmenting Paths

- Insight: We can increase the flow through each edge of this path by up to 4 units without violating the capacity constraint:





G with augmented flow based on path p



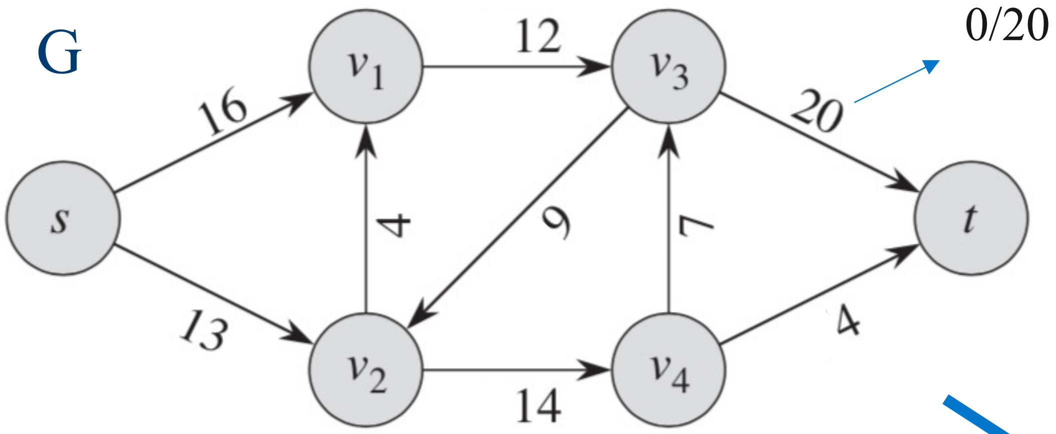
Ford-Fulkerson Algorithm

- Given G, s, t, and c, it finds a flow whose value is maximum

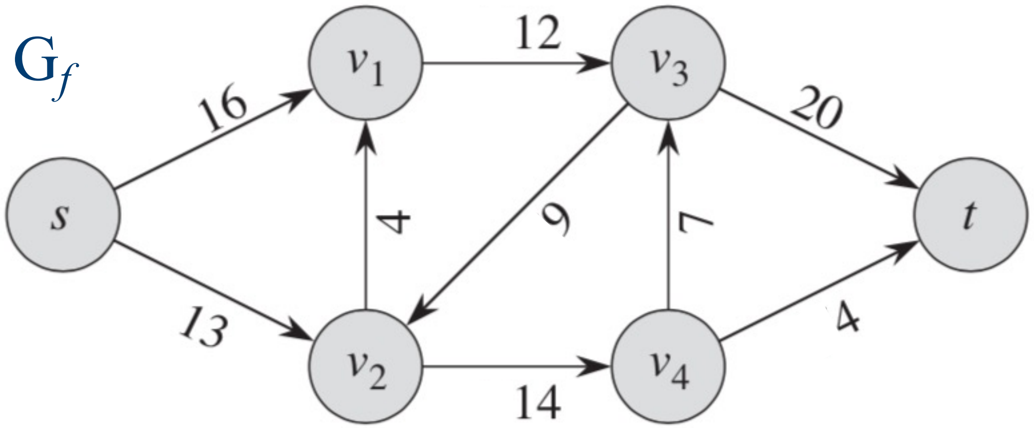
- General ideas:
  - In each iteration of the Ford-Fulkerson method, we find some augmenting path p and use p to modify the flow f
  - That is, adding flow when the residual edge in p is an original edge and subtracting it otherwise
  - When no augmenting paths exist, the flow f is a maximum flow.

Ford-Fulkerson Algorithm Example

G

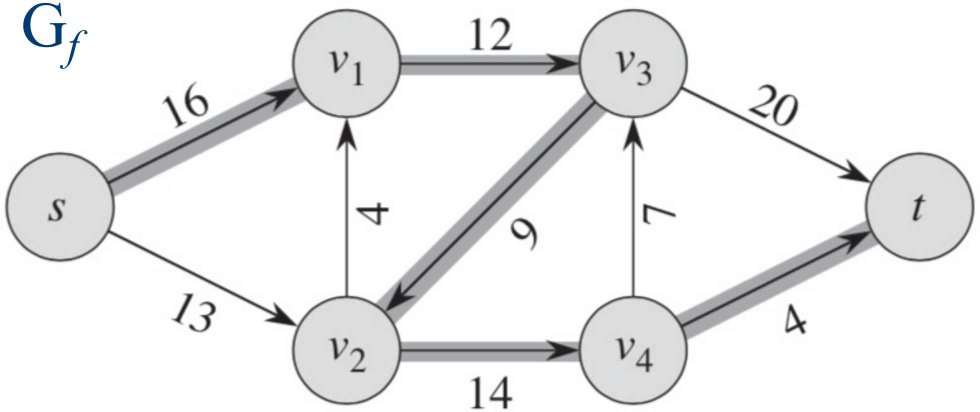


G_f



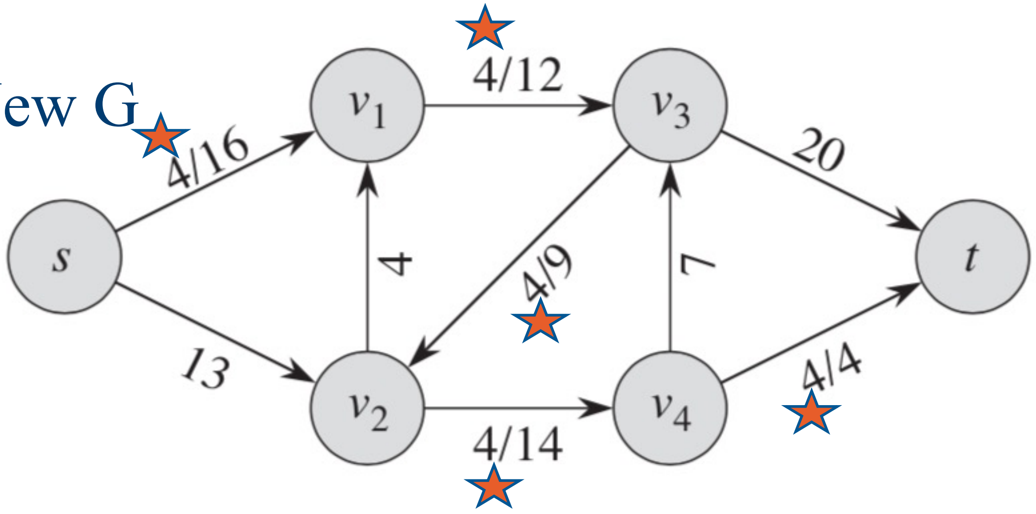
Are there any augmenting path?

G_f



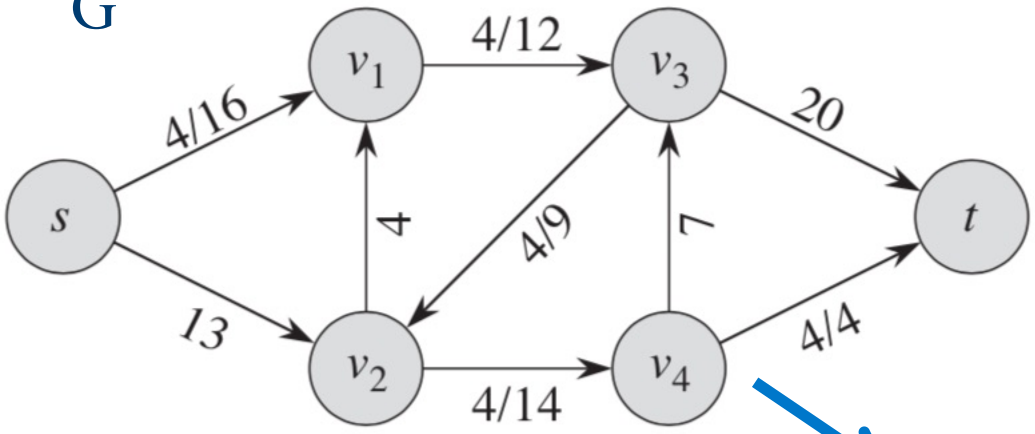
$c_f(p) = 4$

New G

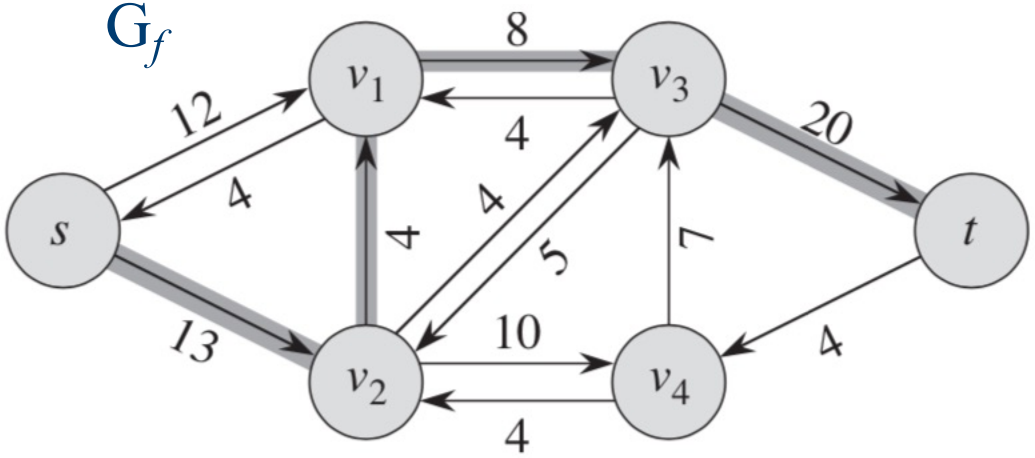


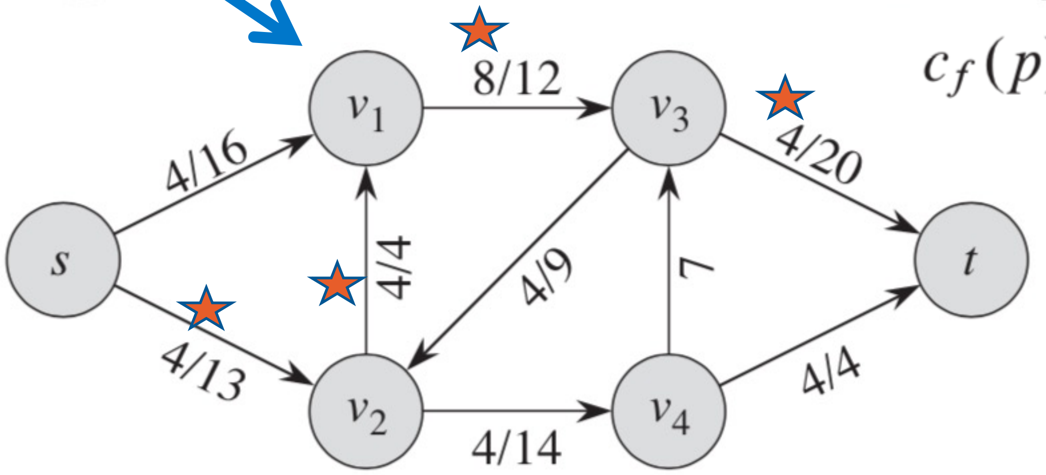
Ford-Fulkerson Algorithm Example

G



G_f



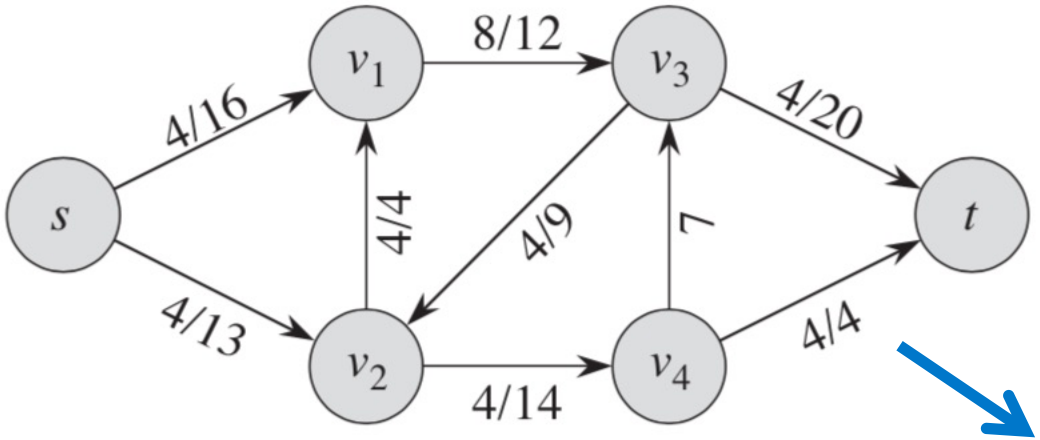


$c_f(p) = 4$

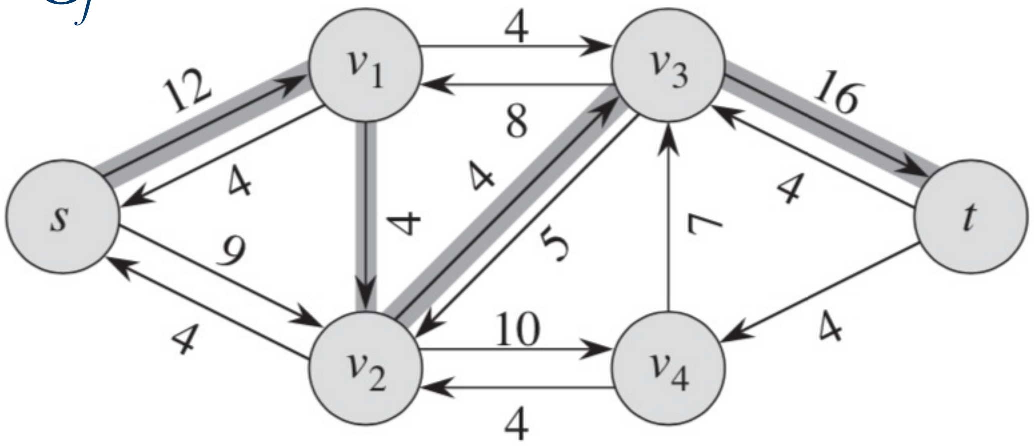
New G

Ford-Fulkerson Algorithm Example

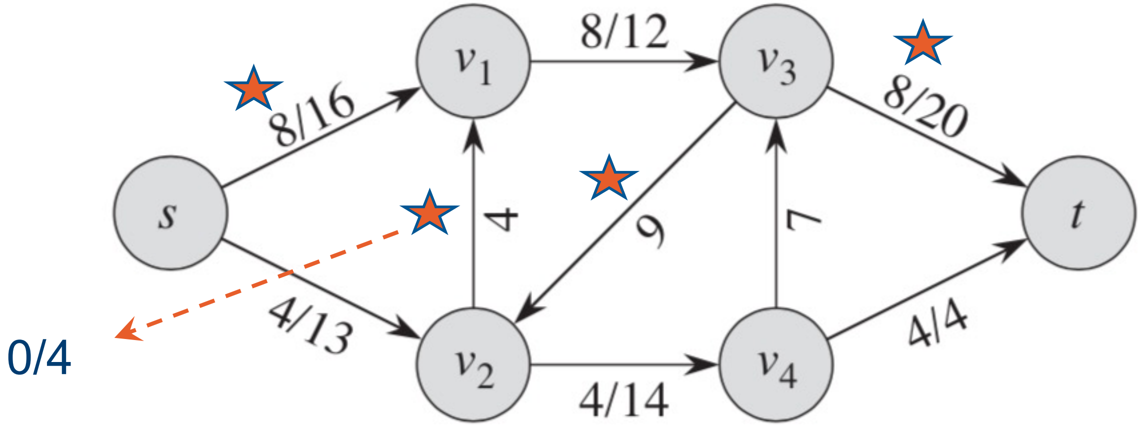
G



G_f



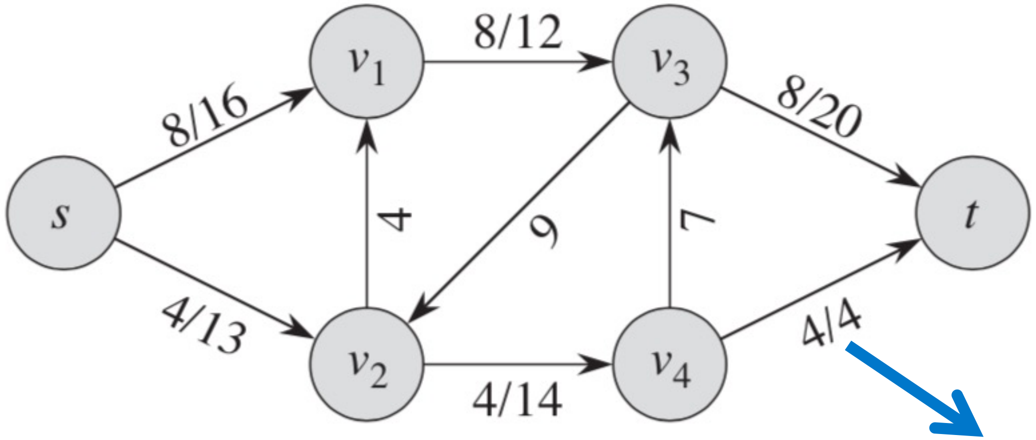
$c_f(p) = 4$



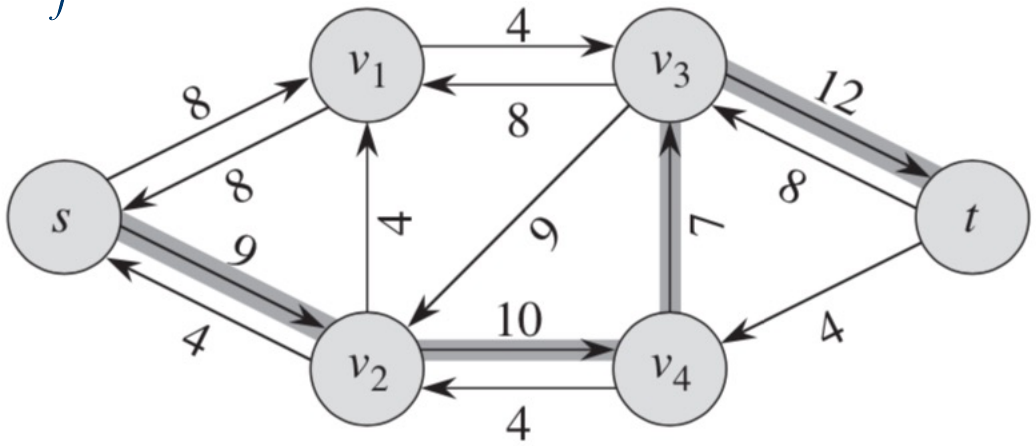
New G

Ford-Fulkerson Algorithm Example

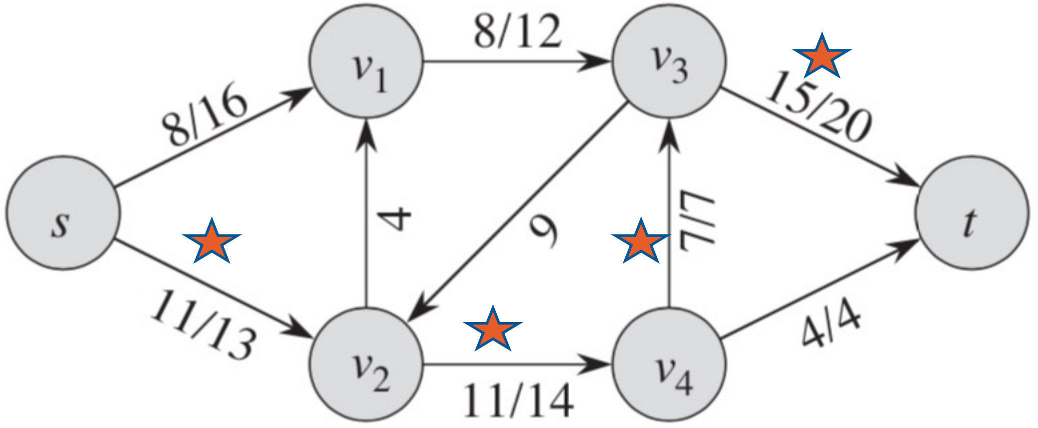
G



G_f

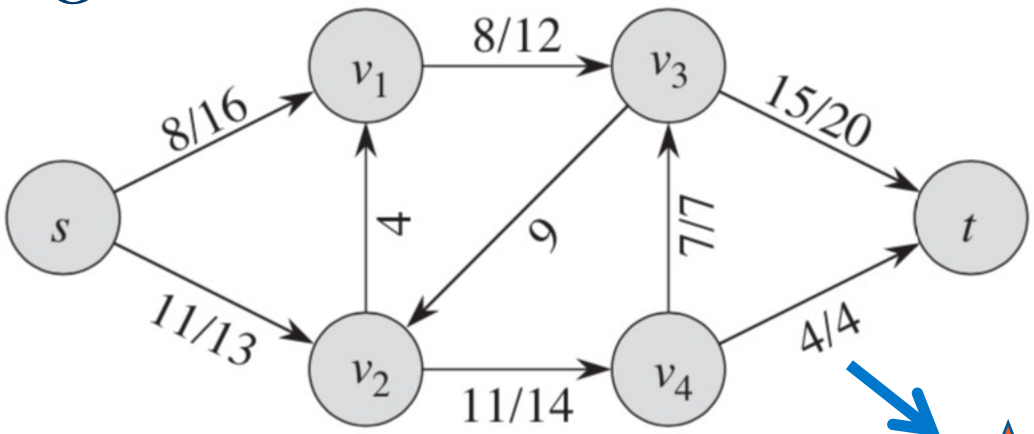


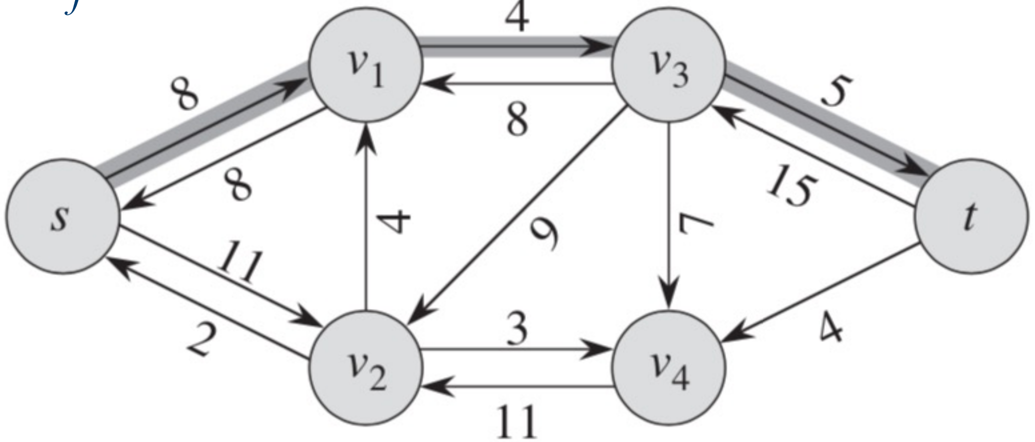
$c_f(p) = 7$



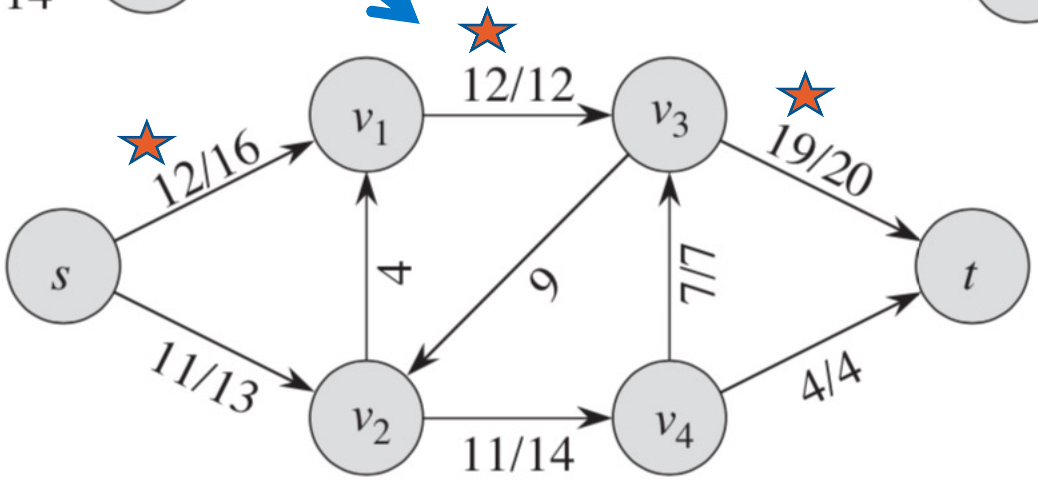
New G

Ford-Fulkerson Algorithm Example





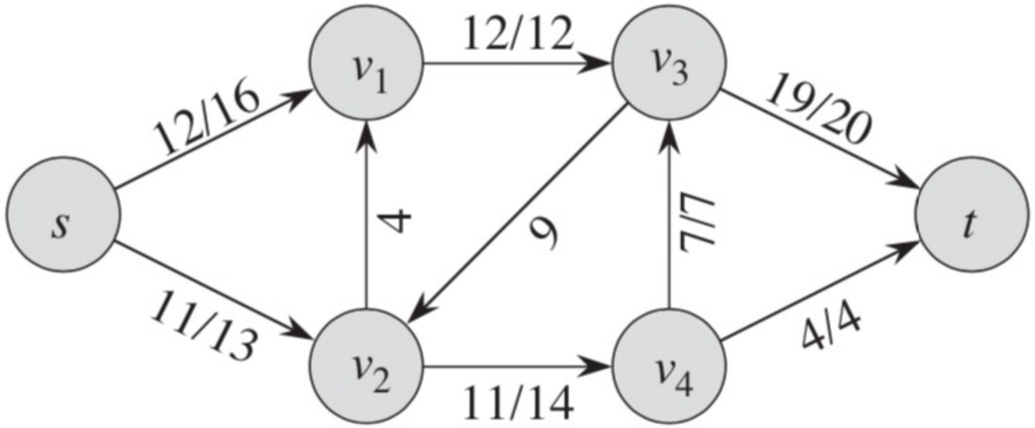
$c_f(p) = 4$



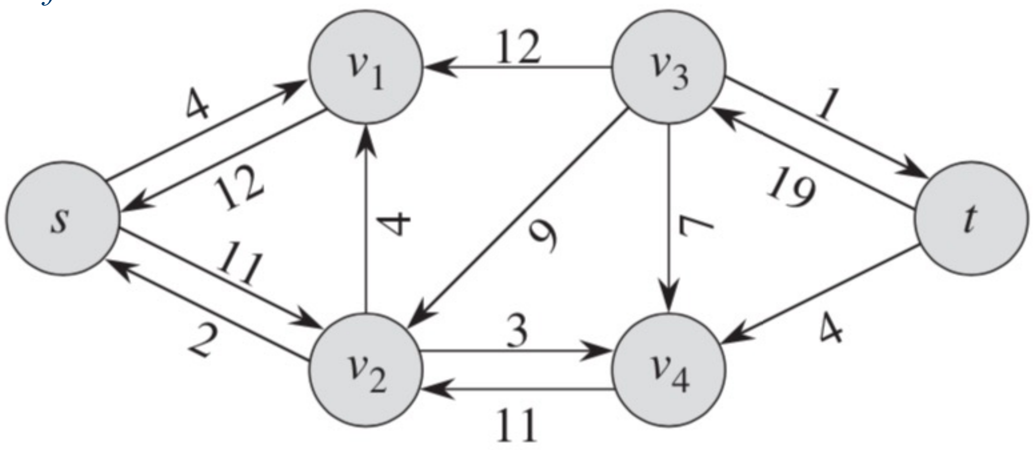
New G

Ford-Fulkerson Algorithm Example

G



G_f



Answer = 23

No Augmenting Path !

Ford-Fulkerson Algorithm

FORD-FULKERSON(G, s, t)

for each edge $(u, v) \in G.E$
    $(u, v).f = 0$
while there exists a path p from s to t in the residual network G_f
    $c_f(p) = \min\{c_f(u, v) : (u, v) \text{ is in } p\}$
    for each edge (u, v) in p
        if $(u, v) \in E$
            $(u, v).f = (u, v).f + c_f(p)$
        else $(v, u).f = (v, u).f - c_f(p)$

Ford-Fulkerson Algorithm

FORD-FULKERSON(G, s, t)

for each edge $(u, v) \in G.E$
    $(u, v).f = 0$ $O(E)$
while there exists a path p from s to t in the residual network G_f $O(|f^*|)$
    $c_f(p) = \min\{c_f(u, v) : (u, v) \text{ is in } p\}$
    for each edge (u, v) in p
        if $(u, v) \in E$
            $(u, v).f = (u, v).f + c_f(p)$
        else $(v, u).f = (v, u).f - c_f(p)$ $O(E|f^*|)$

Time Complexity:

$$O(E|f^*|)$$

$|f^*|$ denotes a maximum flow in the network

Ford-Fulkerson Algorithm

- Why we come up with maximum flow when there is no augmentation path in G_f ?

- **Theorem (Max-flow min-cut theorem):** The following are equivalent:
  - f is a maximum flow
  - G_f has no augmenting path

The Edmonds-Karp Algorithm

- We can improve the bound on Ford-Fulkerson by finding the augmenting path p in line 3 with a breadth-first search
- That is, we choose the augmenting path as a *shortest* path from s to t in the residual network, where each edge has unit distance (weight)
- The Edmonds-Karp algorithm runs in $O(VE^2)$

Wrap-up

- We learned about flow networks
  - Definitions
  - Properties
  - Max flow problem
  - Ford-Fulkerson
  - Edmonds-Karp