FLOW NETWORKS

Module 4

Network Flow

Maximum Flow

- A directed graph is interpreted as a flow network:

  - A material coursing through a system from a source, where the material is produced, to a sink, where it is consumed.

  - The source produces the material at some steady rate, and the sink consumes the material at the same rate.

- Maximum problem: to compute the greatest rate at which material can be shipped from the source to the sink.

- Applications which can be modeled by the maximum flow

  - Liquids flowing through pipes
  - Parts through assembly lines
  - current through electrical network
  - information through communication network

■ Definition – flow networks and flows

- A flow network $G = (V, E)$ is a directed graph in which each edge $(u, v) \in E$ has a nonnegative capacity $c(u, v) \geq 0$.
- source: s; sink: t
- For every vertex $v \in V$, there is a path:

  $s \rightsquigarrow v \rightsquigarrow t$

- A flow in G is a real-valued function f: $V \times V \rightarrow \mathbf{R}$ that satisfies the following properties:

  Capacity constraint: For all $u, v \in V$, $f(u, v) \leq c(u, v)$.

  Skew symmetry: For all $u, v \in V$, $f(u, v) = - f(v, u)$.

Flow conservation: For all $u \in V - \{s, t\}$, we require

$$\sum_{v \in V} f(v, u) = \sum_{v \in V} f(u, v) .$$

When $(u, v) \notin E$, there can be no flow from u to v, and $f(u, v) = 0$.

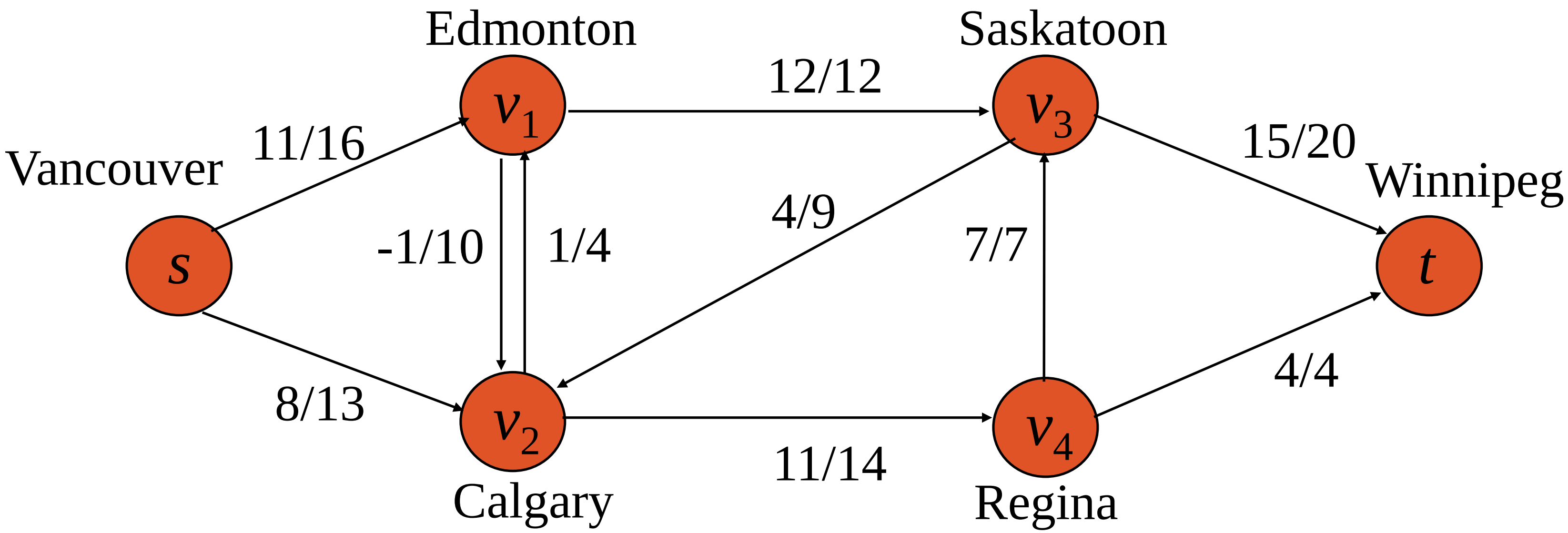
We call the nonnegative quantity $f(u, v)$ the flow from vertex u to vertex v. The ***value*** $|f|$ of a flow f is defined as

$$|f| = \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s) , \tag{26.1}$$

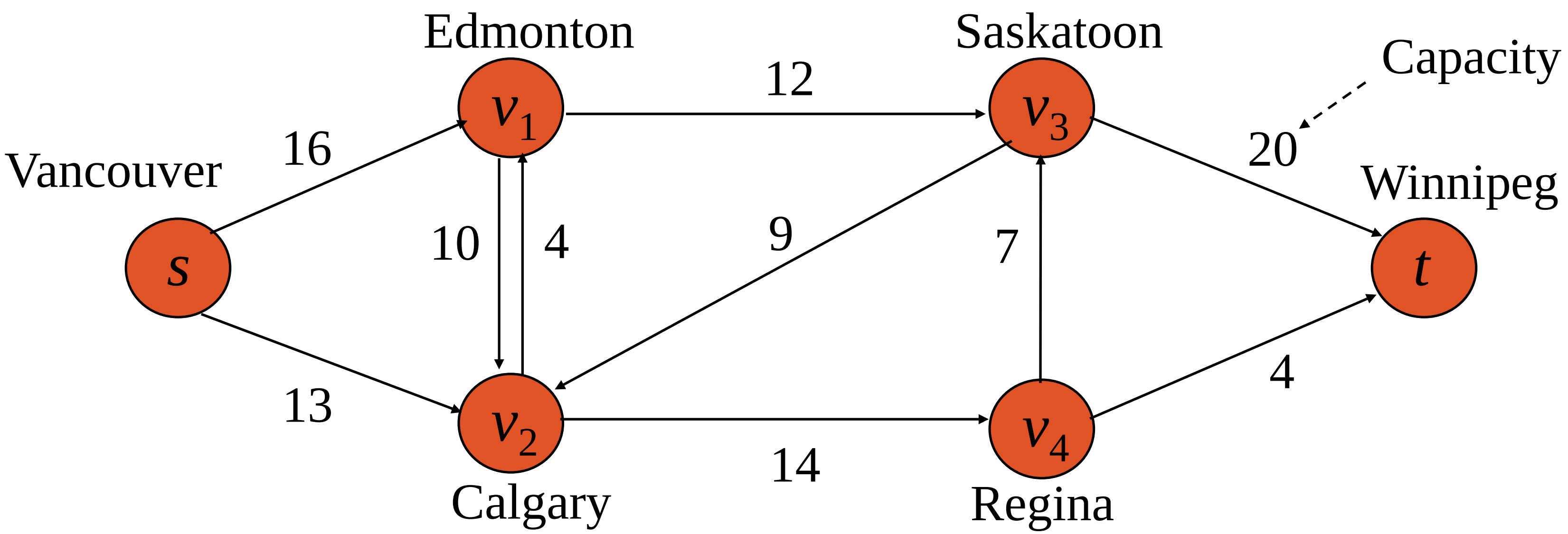
The quantity $f(u, v)$, which can be positive, zero, or negative, is called the ***flow*** from vertex u to vertex v. The value of a flow f is defined as the total flow out of the source

$$|f| = \sum_{v \in V} f(s, v)$$

- Example



- Example


Edmonton
Saskatoon
Capacity
Vancouver
Winnipeg
Calgary
Regina
s
v_1
v_2
v_3
v_4
t
16
13
10
4
12
9
7
14
20
4

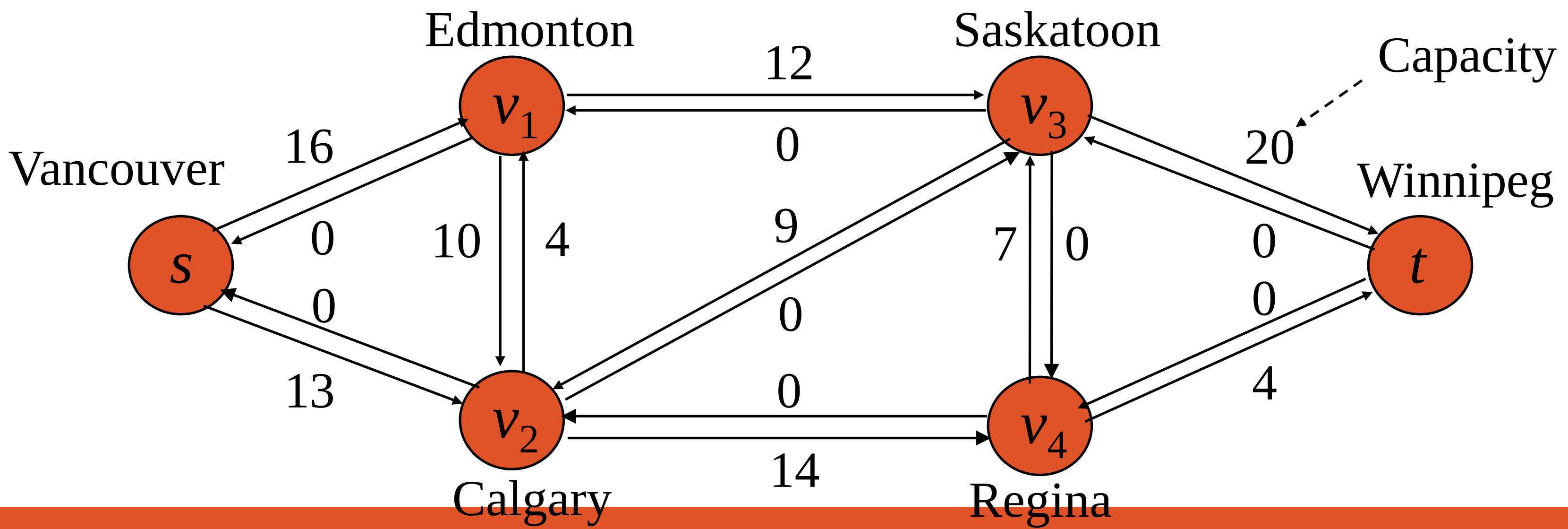
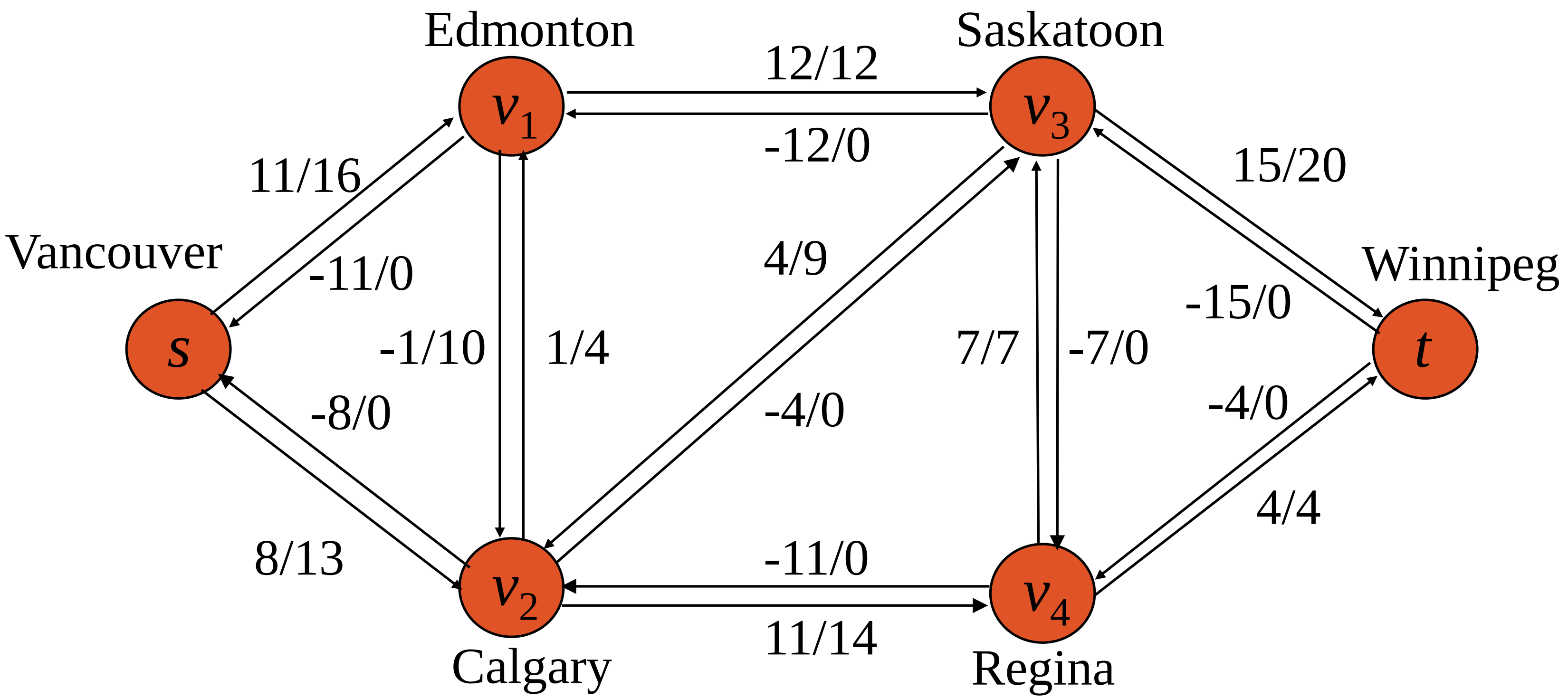

Edmonton
Saskatoon
Capacity
Vancouver
Winnipeg
Calgary
Regina
s
v_1
v_2
v_3
v_4
t
16
0
0
13
10
4
12
0
9
0
7
0
0
14
20
0
0
4

Figure 26.1 **(a)** A flow network $G = (V, E)$ for the Lucky Puck Company's trucking problem. The Vancouver factory is the source s, and the Winnipeg warehouse is the sink t. The company ships pucks through intermediate cities, but only $c(u, v)$ crates per day can go from city u to city v. Each edge is labeled with its capacity. **(b)** A flow f in G with value $|f| = 19$. Each edge (u, v) is labeled by $f(u, v)/c(u, v)$. The slash notation merely separates the flow and capacity; it does not indicate division.

■ Example



$\sum_{v \in V} f(u,v) = 0$. The total flow out of a vertex is 0.

$\sum_{u \in V} f(u,v) = 0$. The total flow into a vertex is 0.

Network Flow Definitions

- Capacity
- Source, Sink
- Capacity Condition
- Conservation Condition
- Value of a flow

Network Flow Definitions

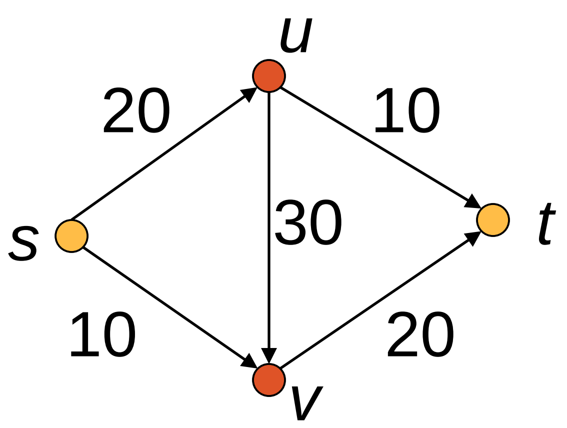
- Flowgraph: Directed graph with distinguished vertices s (source) and t (sink)
- Capacities on the edges, c(e) >= 0
- Problem, assign flows f(e) to the edges such that:
  - 0 <= f(e) <= c(e)
  - Flow is conserved at vertices other than s and t
    - Flow conservation: flow going into a vertex equals the flow going out
  - The flow leaving the source is a large as possible

Network

A directed graph $G = (V,E)$ such that

- each directed edge e has its nonnegative **capacity** denoted by c_e
- there is a node s (*source*) with no incoming edges
- there is a node t (*target*) with no outgoing edges

u,*v* - internal nodes



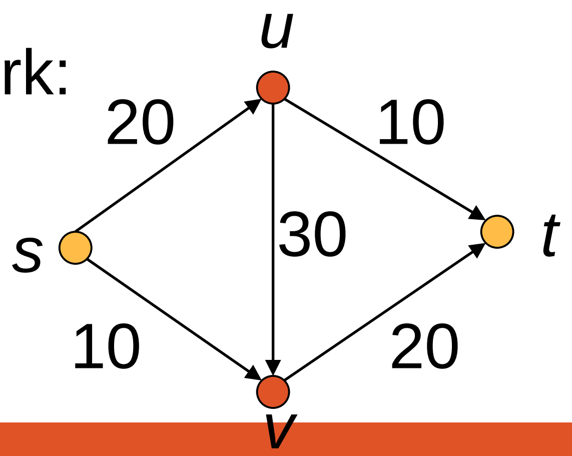
Flow

s-t flow in $G = (V,E)$ is a function f from E to $\mathbf{R}^+$

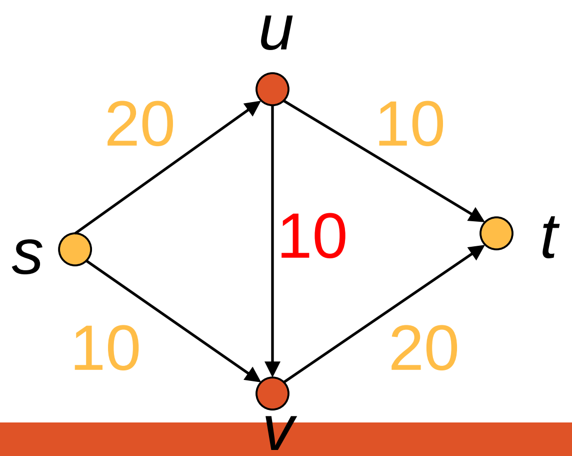
- **capacity condition:** for each e, $0 \leq f(e) \leq c_e$
- **conservation condition:** for each internal node v, $\sum_{e \text{ in } v} f(e) = \sum_{e \text{ out } v} f(e)$
- there is a node t (*target*) with no outgoing edges

Property: $\sum_{e \text{ in } t} f(e) = \sum_{e \text{ out } s} f(e)$

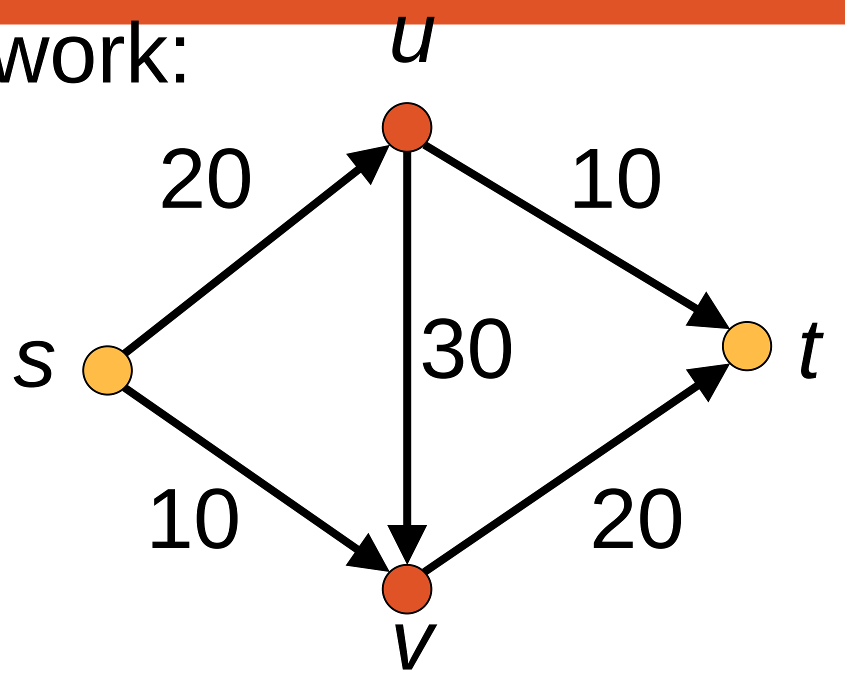
Network:



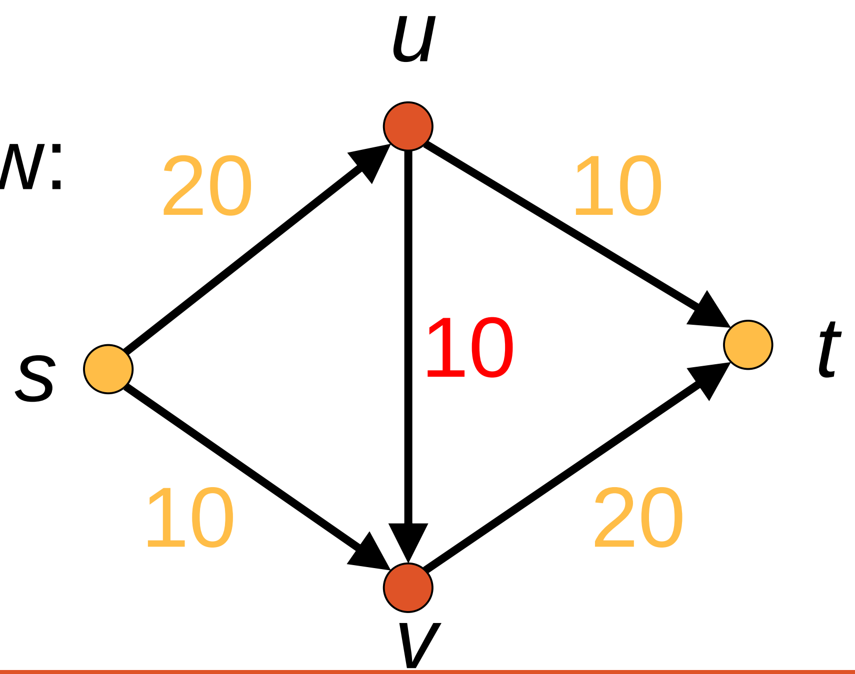
Flow:



Network:



Flow:



Useful definitions

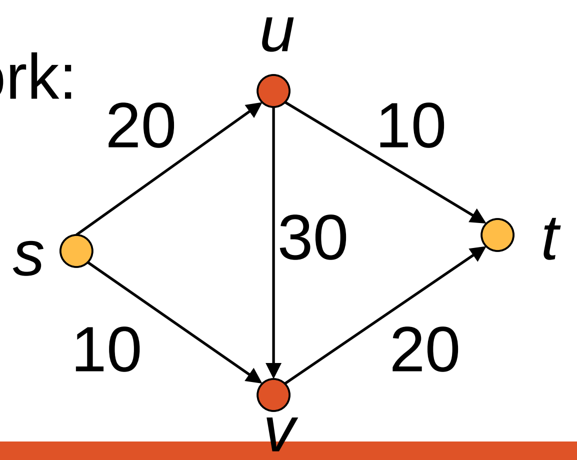
Given s-t flow f in G = (V,E) and any subset of nodes S

- $f^{\text{in}}(S) = \sum_{e \text{ in } S} f(e)$
- $f^{\text{out}}(S) = \sum_{e \text{ out } S} f(e)$

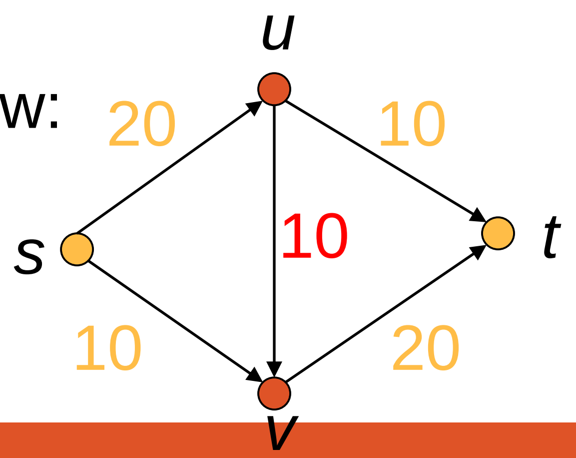
Property: $f^{\text{in}}(t) = f^{\text{out}}(s)$

Example: $f^{\text{in}}(u,v) = f^{\text{out}}(u,v) = 30$

Network:



Flow:



The *total positive flow* entering a vertex v is defined by

$$\sum_{u \in V, f(u,v)>0} f(u,v)$$

The *total net flow* at a vertex is the total positive flow leaving the vertex minus the total positive flow entering the vertex.

The *interpretation* of the flow-conservation property:

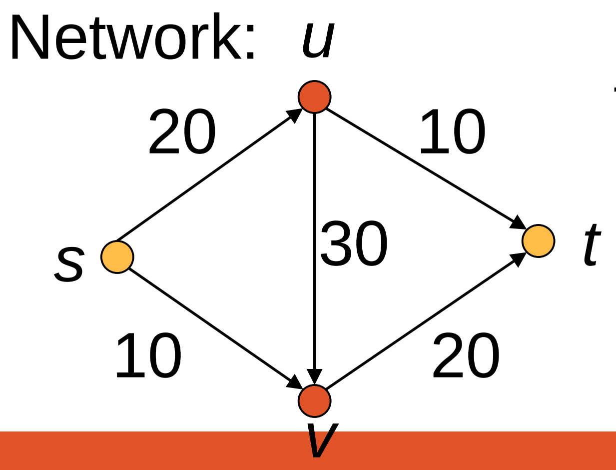
- The total positive flowing entering a vertex other than the source or sink must equal the total positive flow leaving that vertex.
- For all $u \in V - \{s, t\}$, $\sum_{v \in V} f(u,v) = 0$. That is, the total flow out of u is 0.
  For all $v \in V - \{s, t\}$, $\sum_{u \in V} f(u,v) = 0$. That is, the total flow into v is 0.

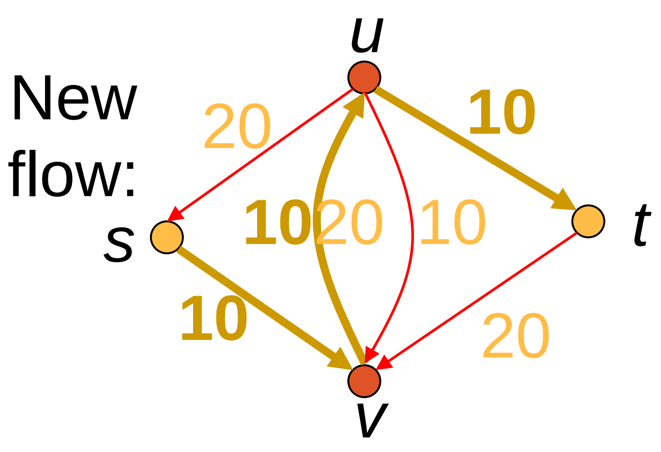
Augmenting path & augmentation

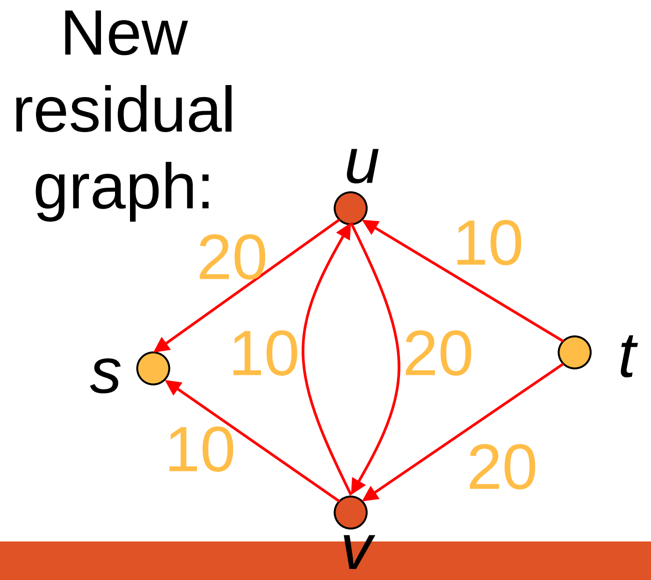
Assume that we are given a flow f in graph G, and the corresponding residual graph G_f

1. Find a new flow in residual graph - through a path with no repeating nodes, and value equal to the minimum capacity on the path (augmenting path)

2. Update residual graph along the path







- The Ford-Fulkerson method
  - *The maximum-flow problem*: given a flow network G with source s and sink t, we wish to find a flow f of maximum value. ($\sum_{u \in V, f(u,v)>0} f(u,v)$)
  - important concepts:

    residual networks

    augmenting paths

    cuts

Ford-Fulkerson-Method(G, s, t)

1. Initialize flow f to 0
2. **while** there exists an augmenting path p
3. **do** augment flow f along p
4. **return** f

- Residual networks
  - Given a flow network and a flow, the residual network consists of edges that can admit more flow.
  - Let f be a flow in $G = (V, E)$ with source s and sink t. Consider a pair of vertices $u, v \in V$. The amount of *additional* flow we can push from u to v before exceeding the capacity $c(u, v)$ is the ***residual capacity*** of (u, v), given by

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

  - Example

    If $c(u, v) = 16$ and $f(u, v) = 11$, then $c_f(u, v) = 16 - 11 = 5$.

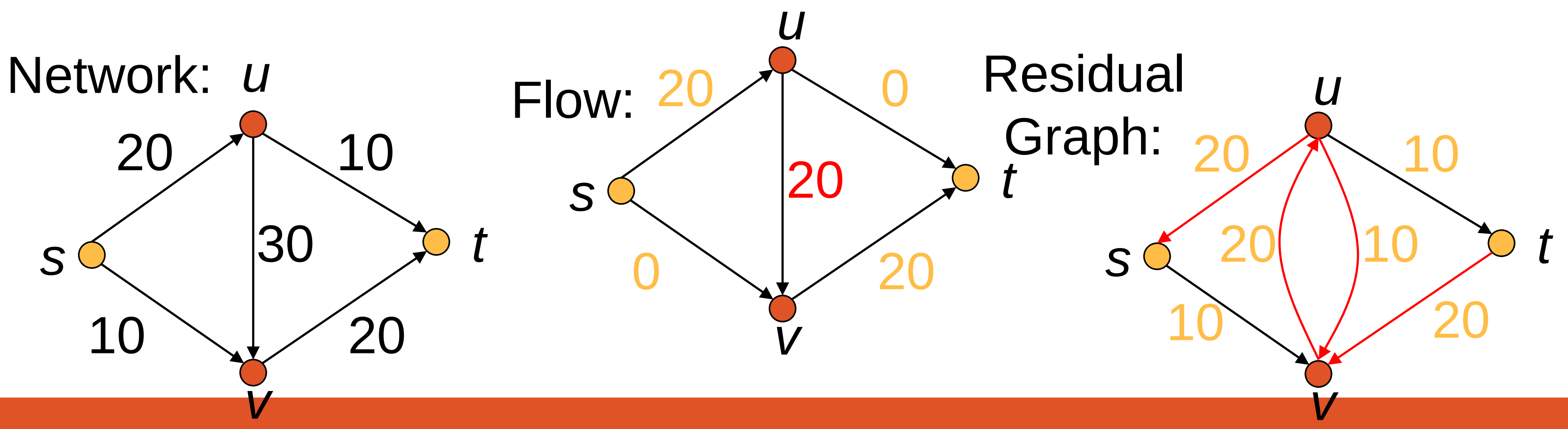
    If $c(u, v) = 16$ and $f(u, v) = -4$, then $c_f(u, v) = 16 - (-4) = 20$.

Residual graph

Assume that we are given a flow f in graph G.

Residual graph G_f

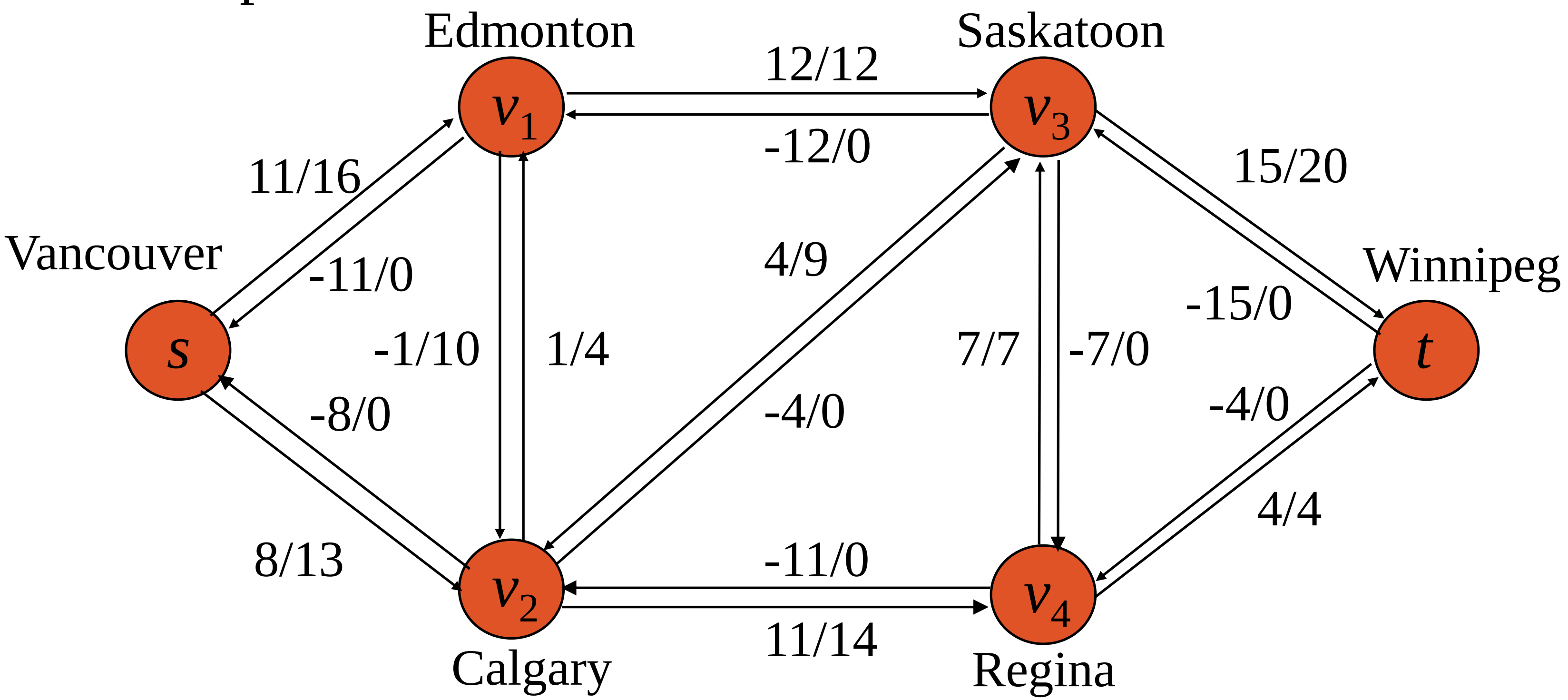
- The same nodes, internal and s,t
- For each edge e in E with $c_e > f(e)$ we put weight $c_e - f(e)$ (**residual capacity**)
- For each edge $e = (u,v)$ in E we put weight $f(e)$ to the backward edge (v,u) (**residual capacity**)



- Residual networks
  - Given a flow network $G = (V, E)$ and a flow f, the ***residual network*** of G induced by f is $G_f = (V, E_f)$, where

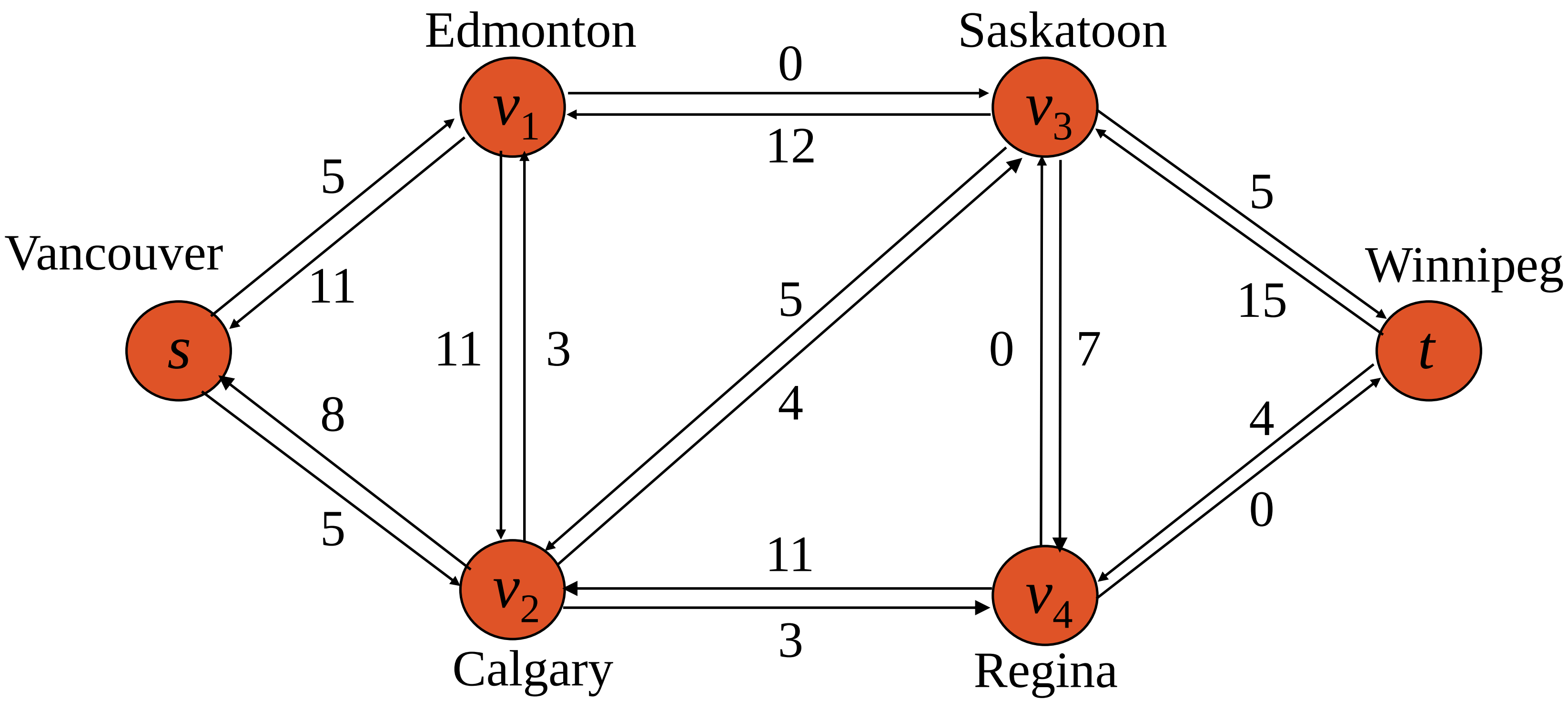
    $$E_f = \{(u, v) \in V \times V: c_f(u, v) > 0\}.$$

  - Example

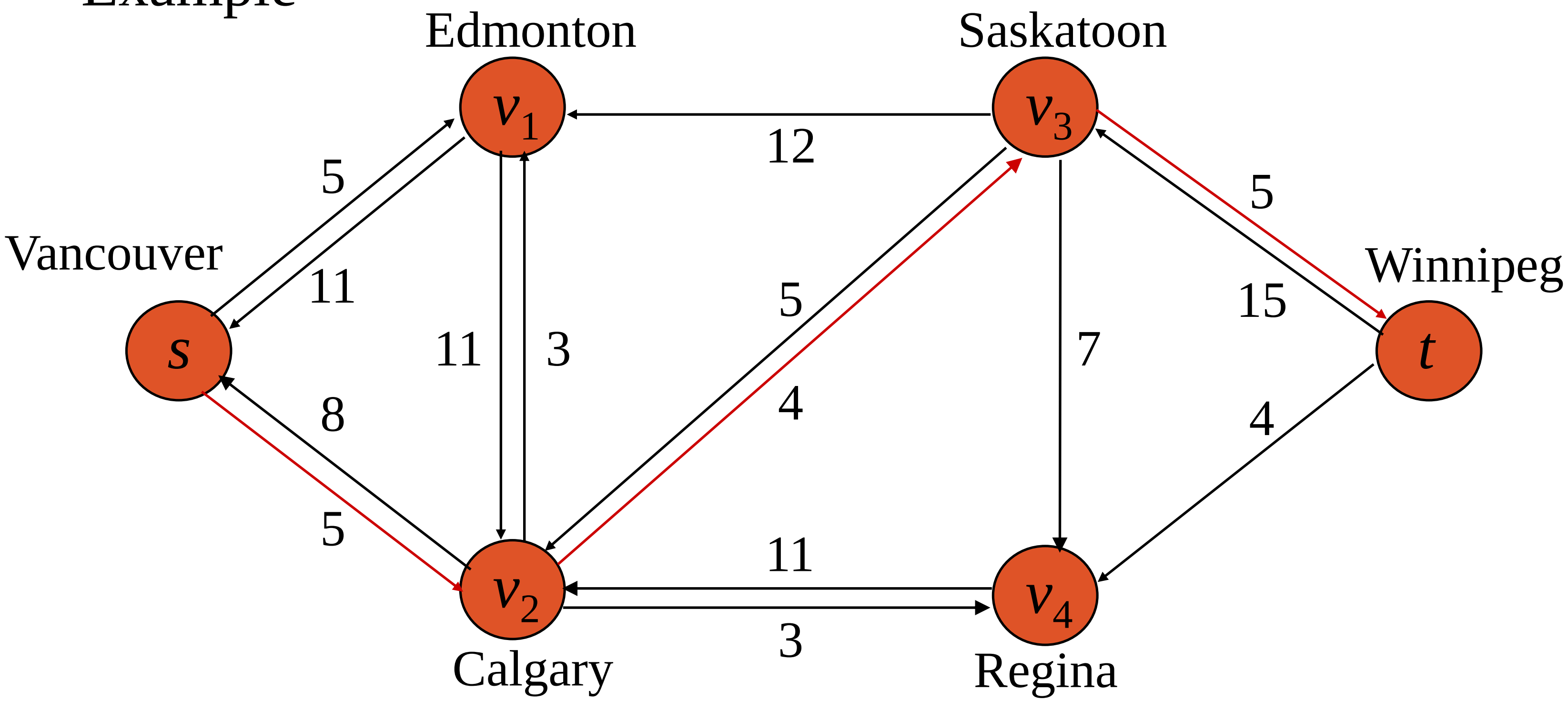


- Residual networks

residual network:



- 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.
  - Example



- Augmenting paths
  - In the above residual network, path $s \to v_2 \to v_3 \to t$ is an augmenting path.
  - We can increase the flow through each edge of this path by up to 4 units without violating a capacity constraint since the smallest residual capacity on this path is $c_f(v_2, v_3) = 4$.
  - *residual capacity of an augmenting path*

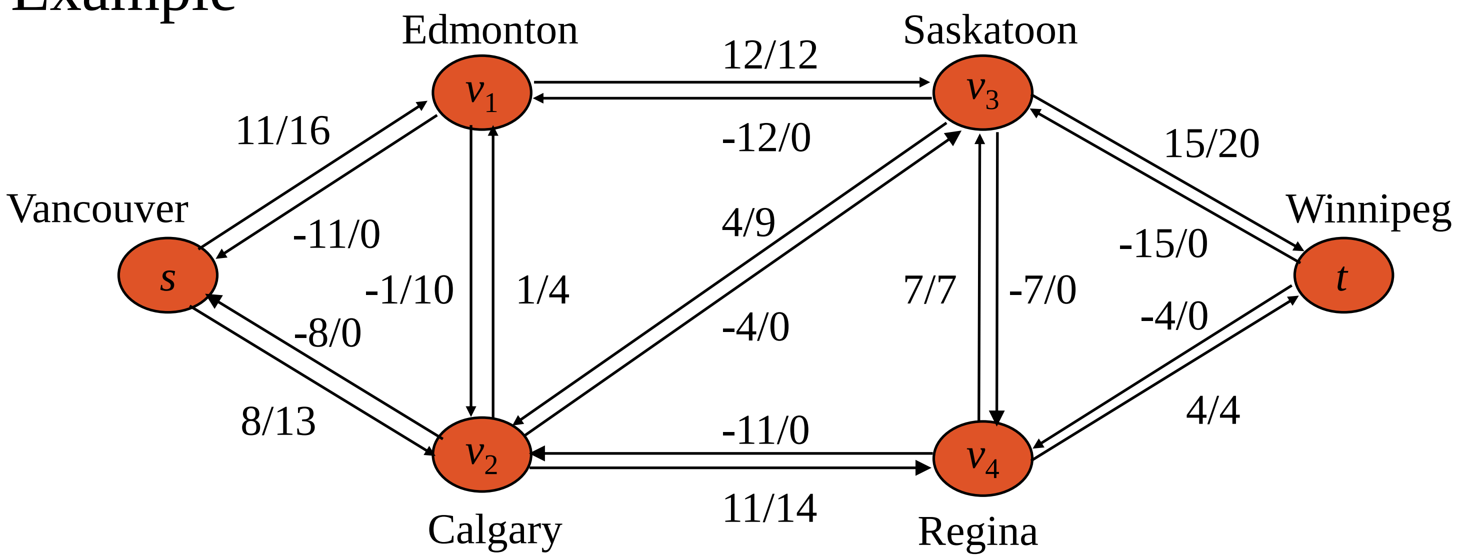
    $$c_f(p) = \min\{c_f(u, v) : (u, v) \text{ is on } p\}.$$

  - **Lemma 26.3** Let $G = (V, E)$ be a network, let f be a flow in G, and let p be an augmenting path in G_f. Define a function $f_p: V \times V \to \boldsymbol{R}$ by

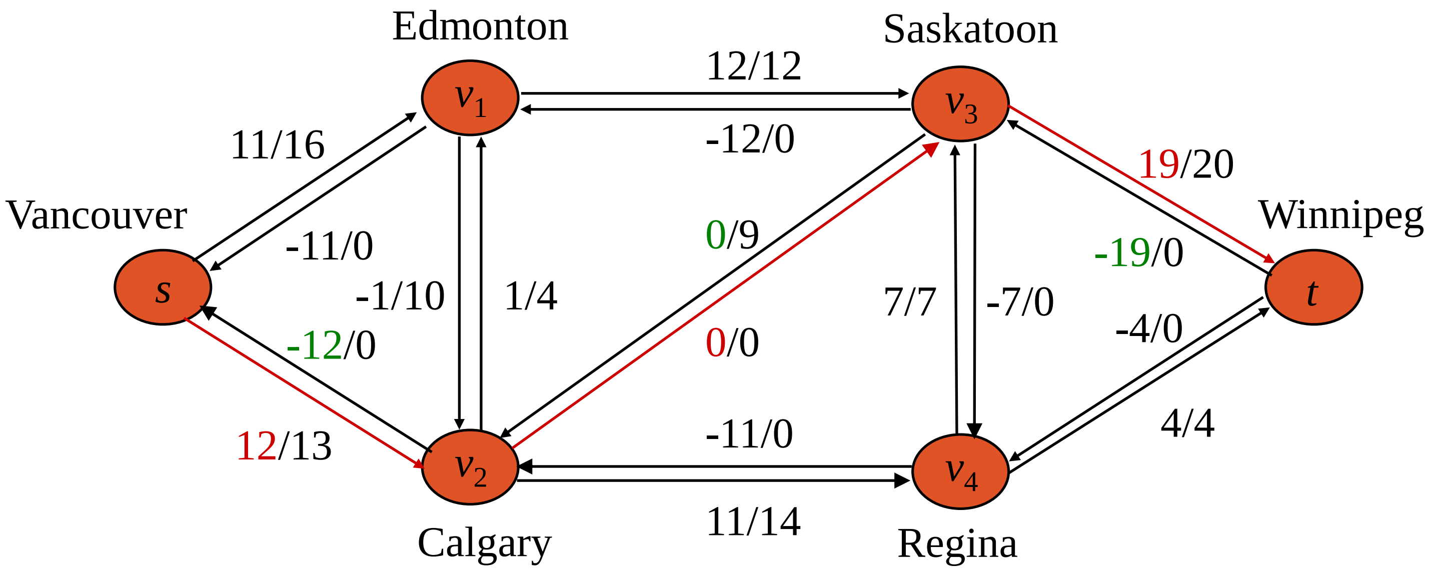
    $$f_p(u, v) = \begin{cases} c_f(p) & \text{if } (u, v) \text{ is on } p, \\ -c_f(p) & \text{if } (v, u) \text{ is on } p, \\ 0 & \text{otherwise.} \end{cases}$$

    Then, f_p is a flow in G_f with value $|f_p| = c_f(p)$.

- Example

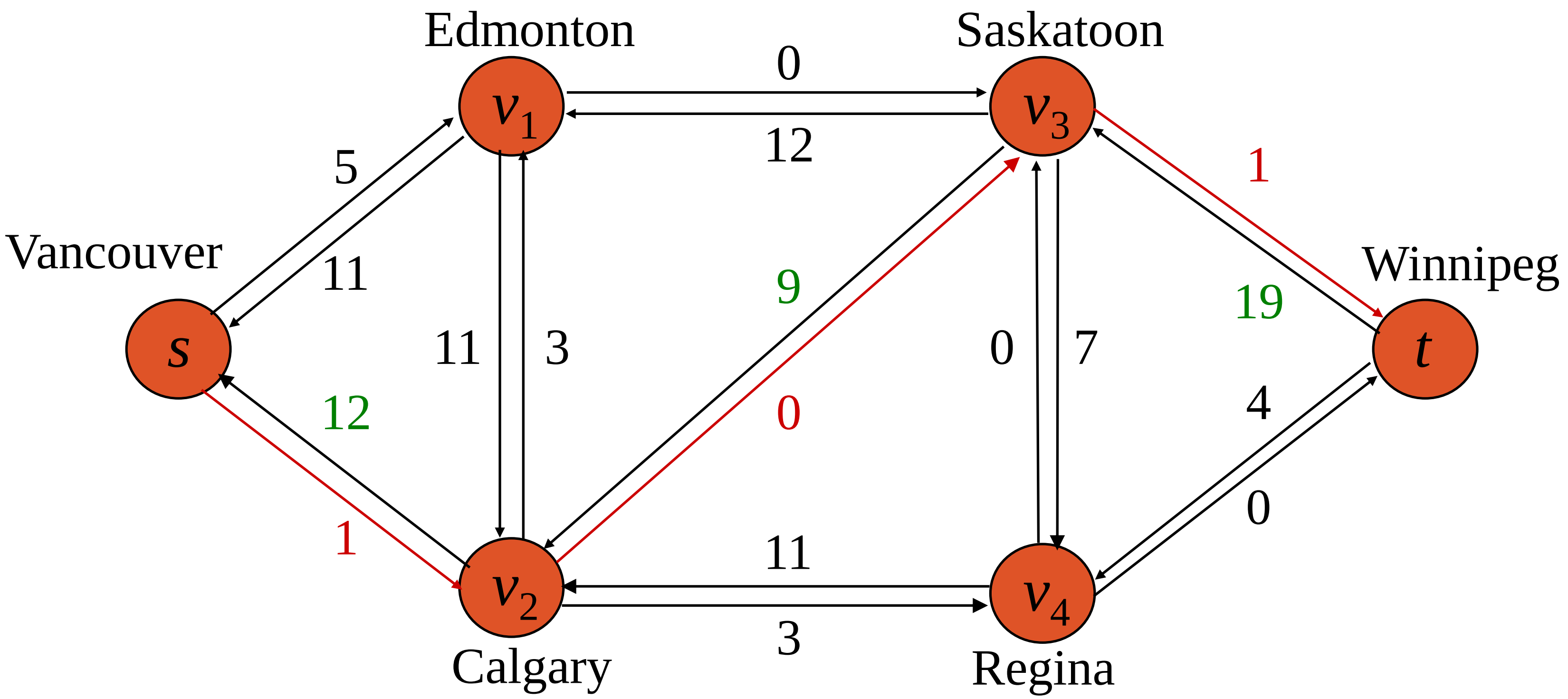


Edmonton
Saskatoon
12/12
-12/0
11/16
15/20
Vancouver
Winnipeg
-11/0
4/9
-15/0
-1/10
1/4
7/7
-7/0
-8/0
-4/0
-4/0
8/13
-11/0
4/4
11/14
Calgary
Regina



Edmonton
Saskatoon
12/12
-12/0
11/16
19/20
Vancouver
Winnipeg
-11/0
0/9
-19/0
-1/10
1/4
7/7
-7/0
-12/0
0/0
-4/0
12/13
-11/0
4/4
11/14
Calgary
Regina

- Residual network induced by the new flow



- Augmenting paths
  - **Corollary 26.4** Let $G = (V, E)$ be a network, let f be a flow in G, and let p be an augmenting path in G_f. Let f_p be defined as in Lemma 26.3. Define a function $f': V \times V \to \boldsymbol{R}$ by
    $$f' = f + f_p.$$
    Then, f' is a flow in G with value $|f'| = |f| + |f_p| > |f|$.

    Proof. Immediately from Lemma 26.2 and 26.3.
- Ford-Fulkerson Algorithm
  - The Ford-Fulkerson method repeatedly augments the flow along augmenting paths until a maximum flow has been found.
  - A flow is maximum if and only if its residual network contains no augmenting path.

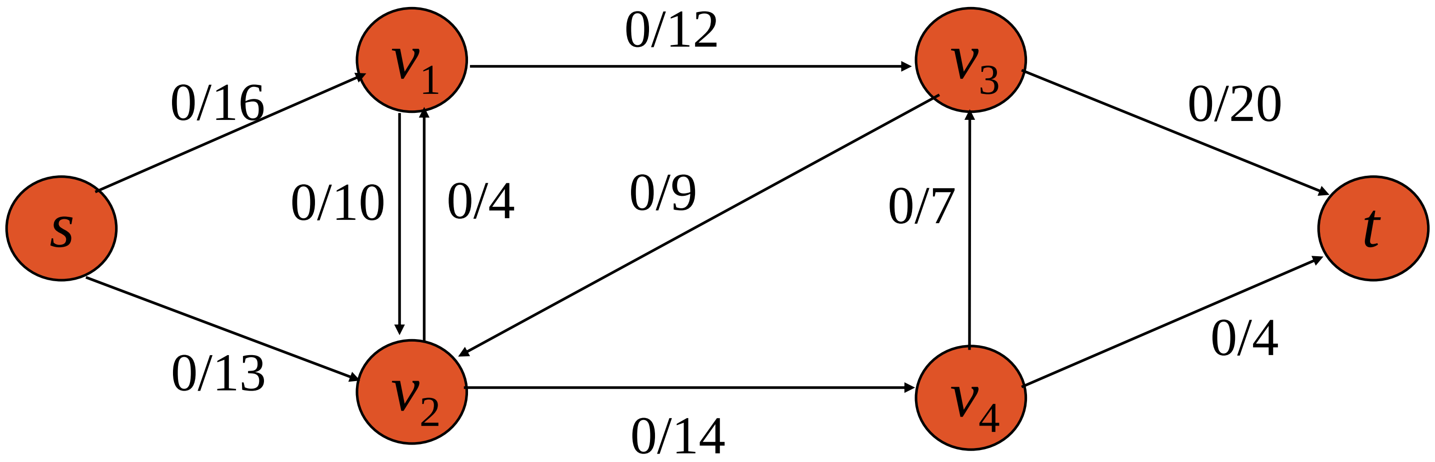
- Ford-Fulkerson algorithm

Ford_Fulkerson(G, s, t)

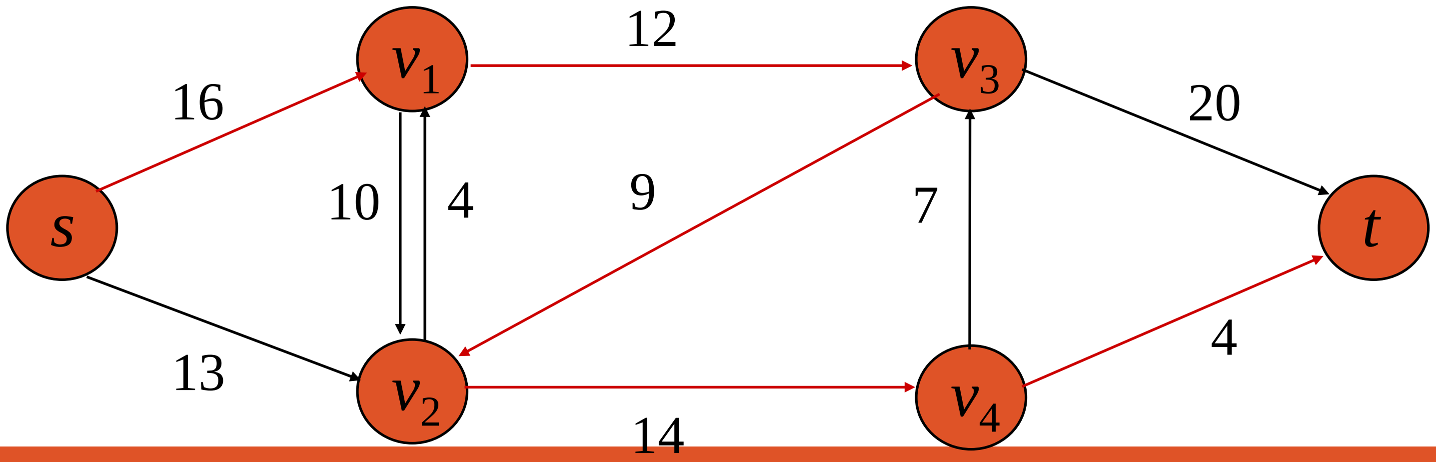
for each edge $(u, v) \in E(G)$
  do $f(u, v) \leftarrow 0$
     $f(v, u) \leftarrow 0$
while there exists a path p from s to t in G_f
  do $c_f(p) \leftarrow \min\{c_f(u, v) : (u, v)$ is in $p\}$
     for each edge (u, v) on p
       do $f(u, v) \leftarrow f(u, v) + c_f(p)$
          $f(v, u) \leftarrow - f(u, v)$

- Sample trace

Initially, the flow on edge is 0.

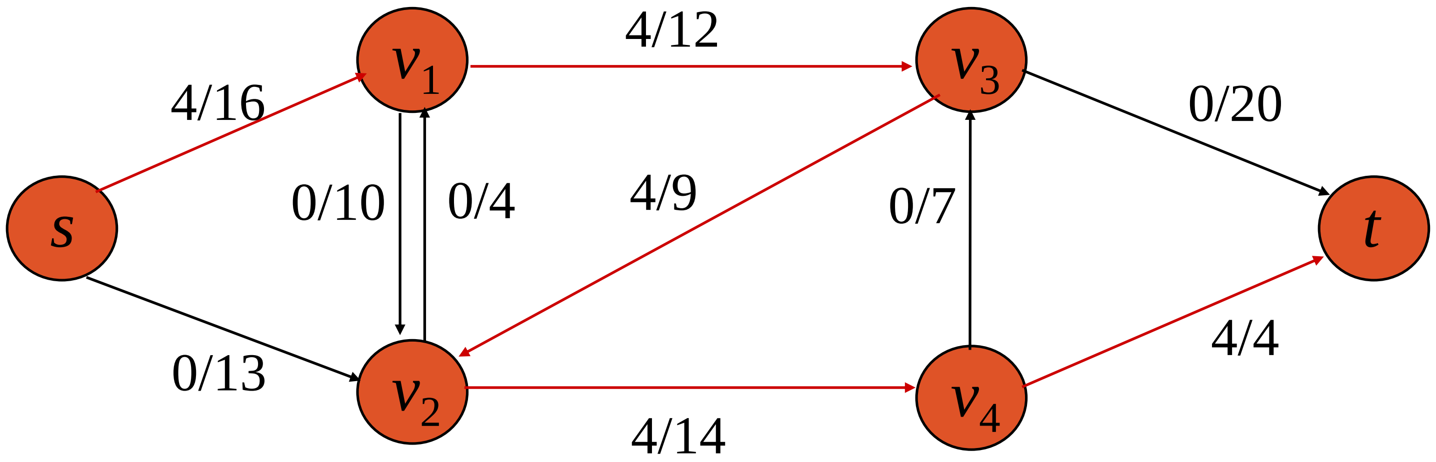


The corresponding residual network:

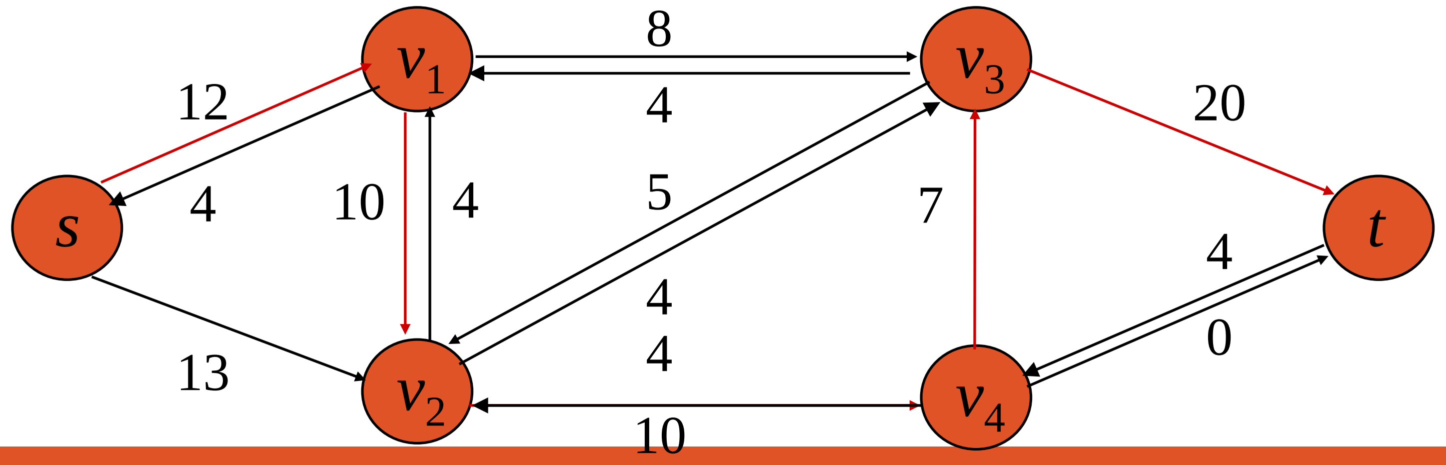


- Sample trace

Pushing a flow 4 on $p1$ (an augmenting path)

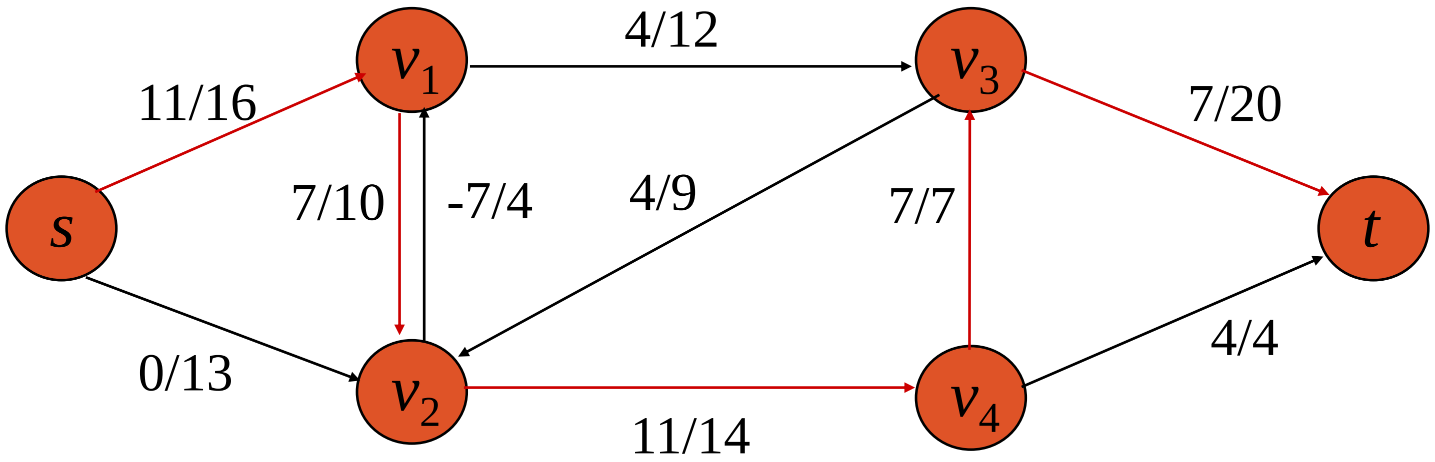


The corresponding residual network:

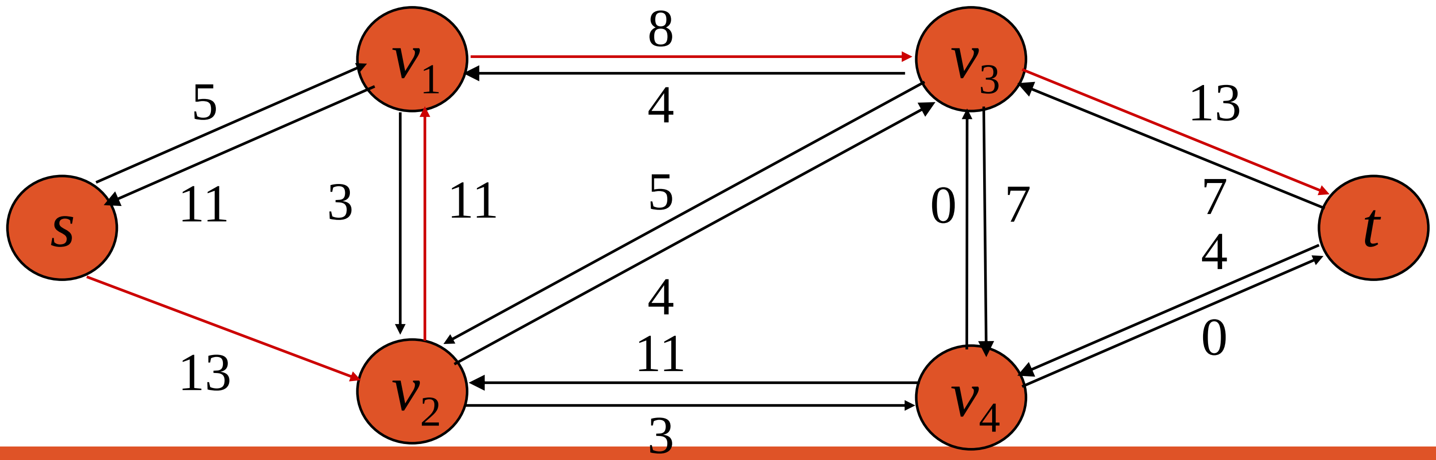


- Sample trace

Pushing a flow 7 on *p*2 (an augmenting path)

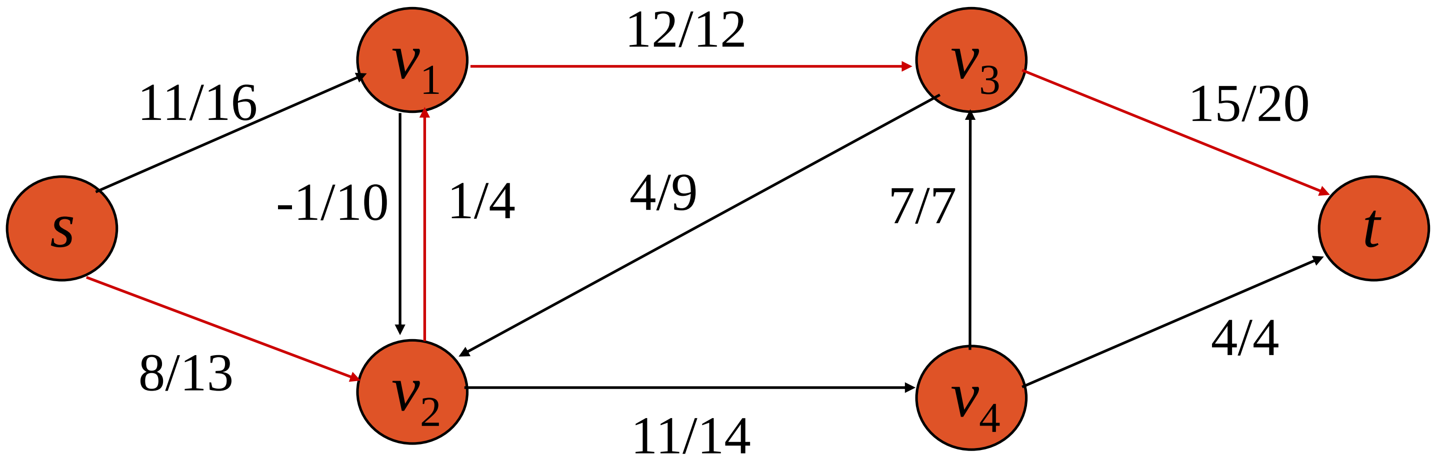


The corresponding residual network:

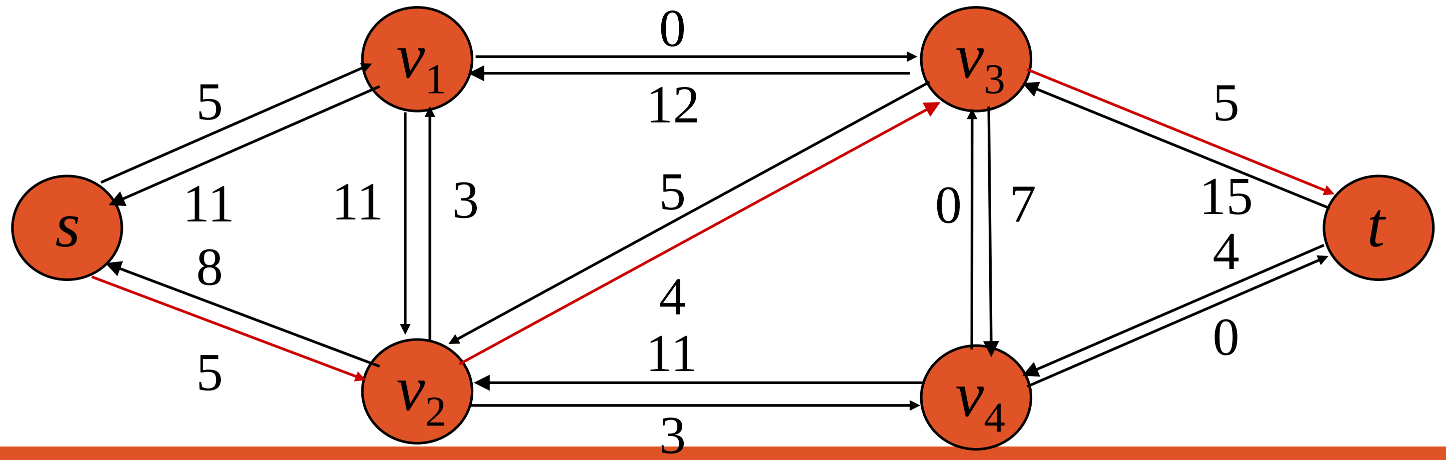


- Sample trace

Pushing a flow 8 on *p*3 (an augmenting path)

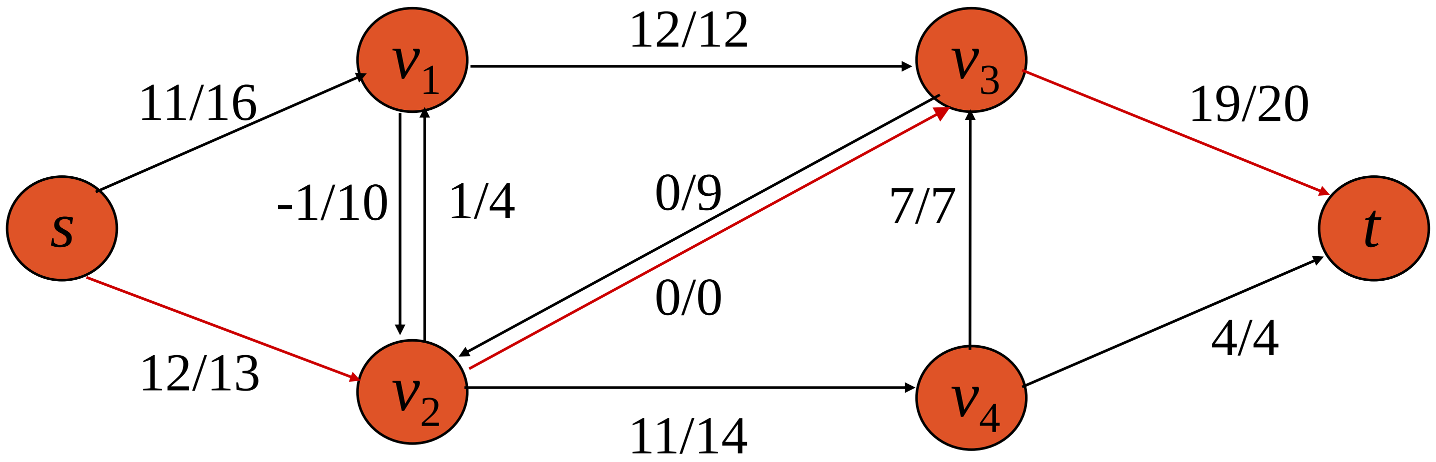


The corresponding residual network:

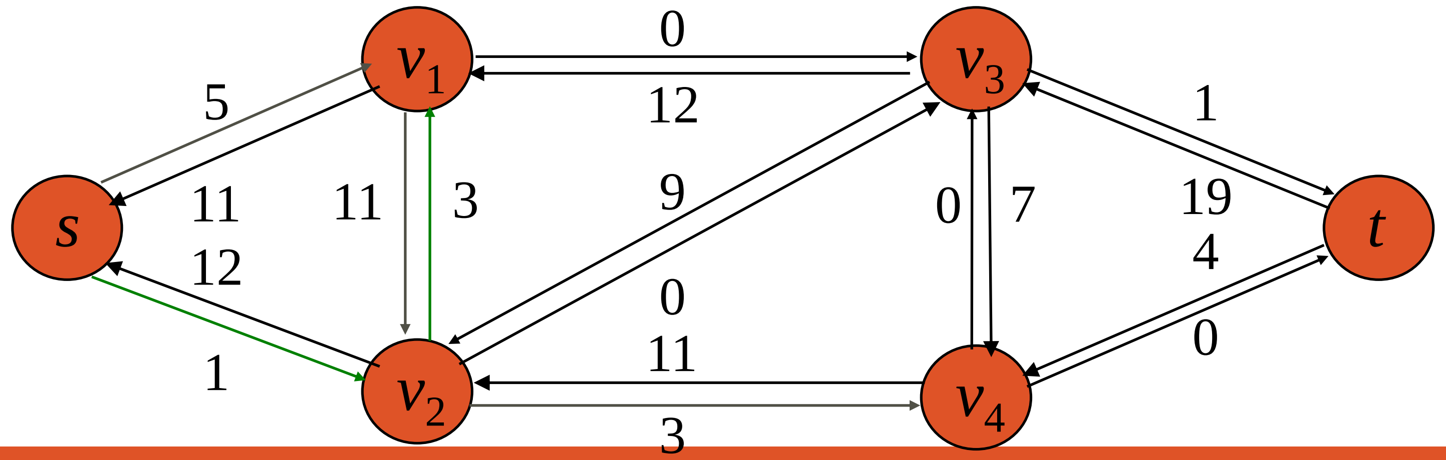


- Sample trace

Pushing a flow 4 on *p*4 (an augmenting path)



The corresponding residual network: no augmenting paths!



MAX-FLOW MIN-CUT THEOREM

- Max-flow min-cut theorem

Theorem 26.7 If f is a flow network $G = (V, E)$ with source s and sink t, then the following conditions are equivalent:

1. f is a maximum flow in G.

2. The residual network G_f contains no augmenting paths.

3. $|f| = c(S, T)$ for some cut (S, T) of G.

Proof. (1) $\Rightarrow$ (2): Suppose for the sake of contradiction that f is a maximum flow in G but that G_f has an augmenting path p. Then, by Corollary 26.4, the flow sum $f + f_p$, where f_p is given by Lemma 26.3, is a flow in G with value strictly greater than $|f|$, contradicting the assumption that f is a maximum flow.

- Max-flow min-cut theorem

Theorem 26.7 If f is a flow network $G = (V, E)$ with source s and sink t, then the following conditions are equivalent:

1. f is a maximum flow in G.

2. The residual network G_f contains no augmenting paths.

3. $|f| = c(S, T)$ for some cut (S, T) of G.

Proof. (2) $\Rightarrow$ (3): Suppose that G_f has no augmenting path. Define $S = \{v \in V$: there exists a path from s to v in $G_f\}$ and $T = V - S$. The partition (S, T) is a cut: we have $s \in S$ trivially and $t \notin S$ because there is no path from s to t in G_f. For each pair of vertices u and v such that $u \in S$ and $v \in T$, we have $f(u, v) = c(u, v)$, since otherwise $(u, v) \in E_f$, which would place v in set S. By Lemma 26.5, therefore, $|f| = f(S, T) = c(S, T)$.

- Max-flow min-cut theorem

Theorem 26.7 If f is a flow network $G = (V, E)$ with source s and sink t, then the following conditions are equivalent:

1. f is a maximum flow in G.

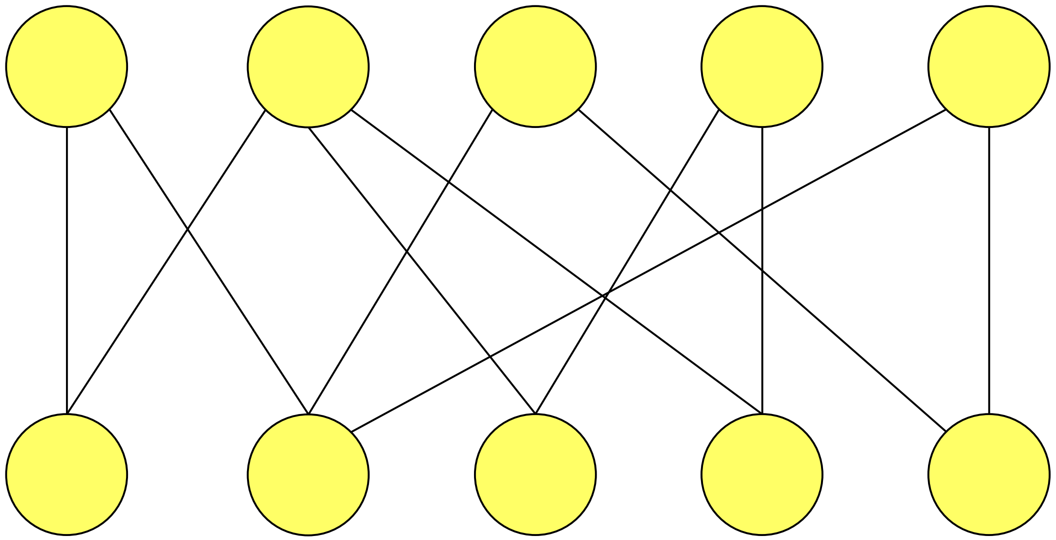
2. The residual network G_f contains no augmenting paths.

3. $|f| = c(S, T)$ for some cut (S, T) of G.

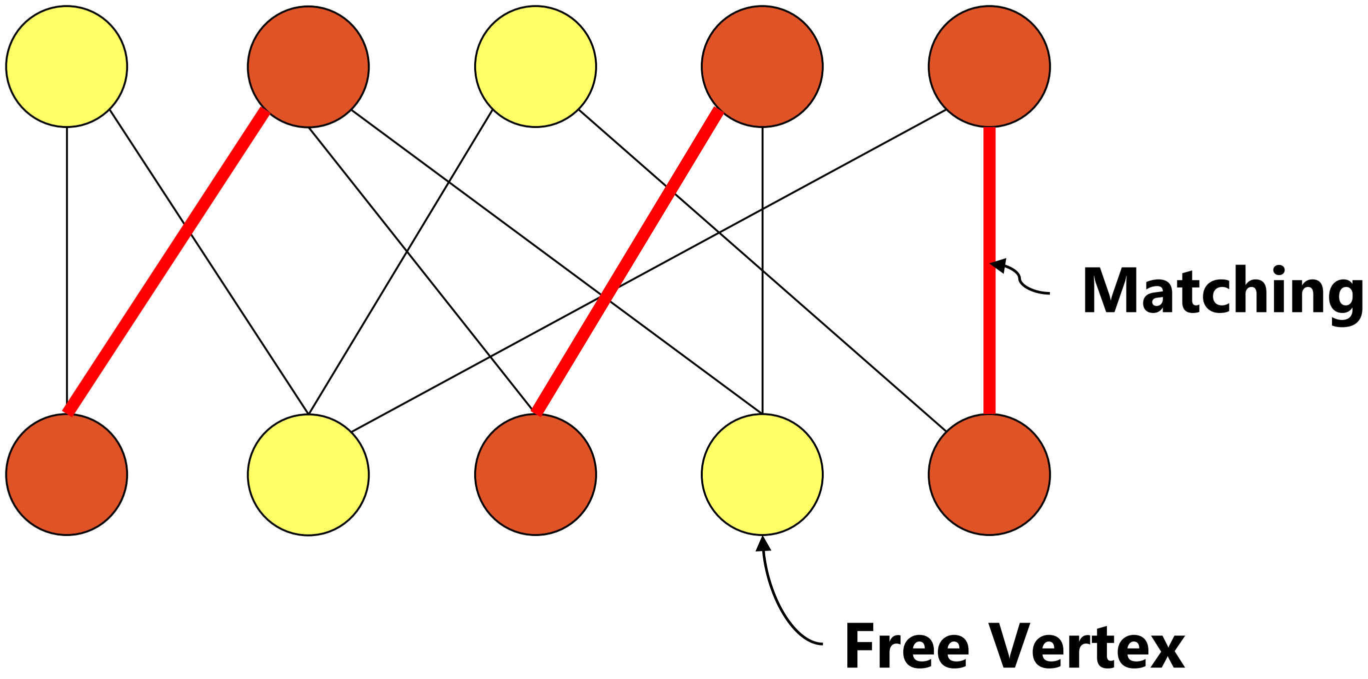
Proof. (3) $\Rightarrow$ (1): By Corollary 26.6, $|f| \le c(S, T)$ for all cuts (S, T). The condition $|f| = c(S, T)$ thus implies that f is a maximum flow.

BIPARTITE MATCHING

Unweighted Bipartite Matching

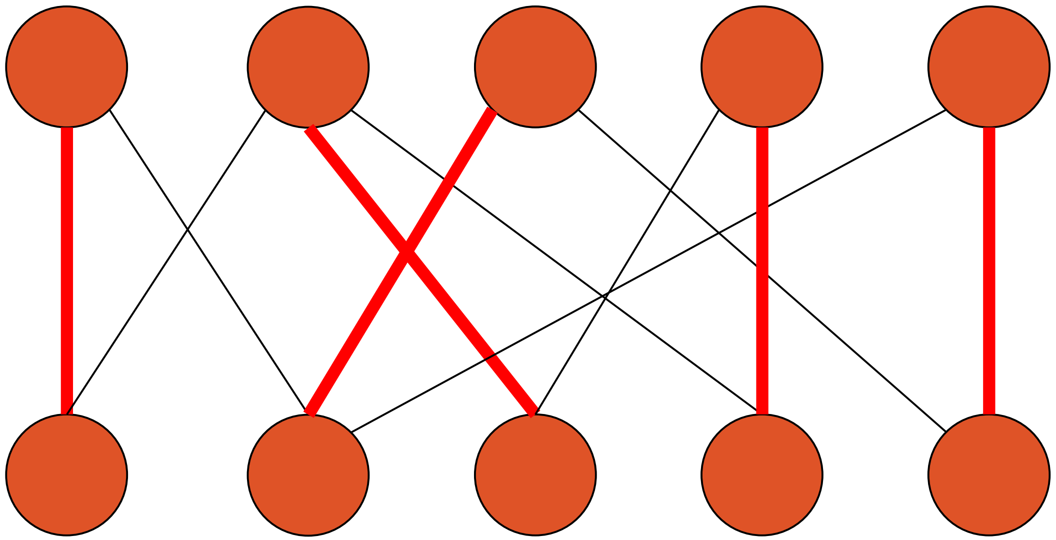


Definitions



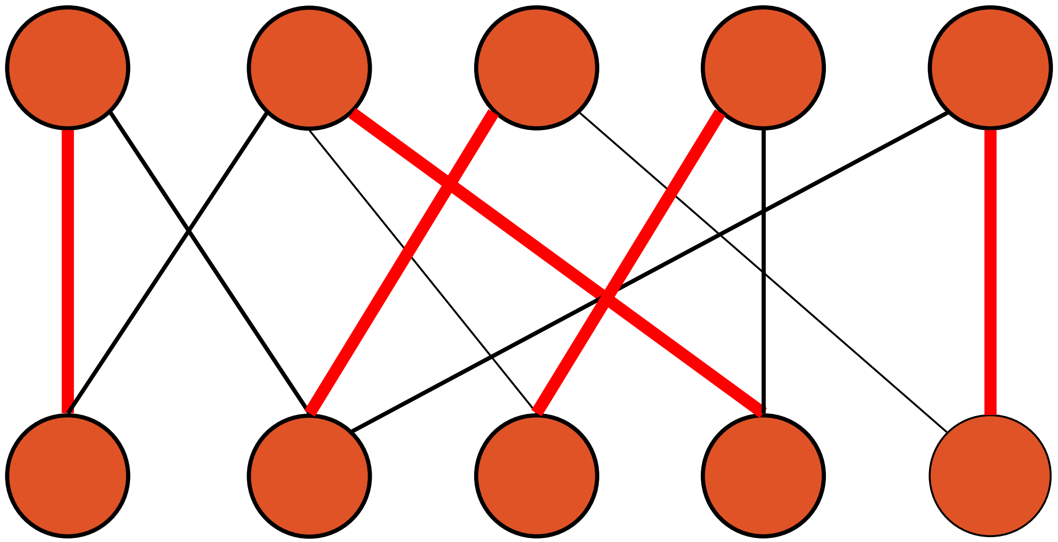
Definitions

- Maximum Matching: matching with the largest number of edges



Definition

- Note that maximum matching is not unique.



Intuition

- Let the top set of vertices be men
- Let the bottom set of vertices be women
- Suppose each edge represents a pair of man and woman who like each other

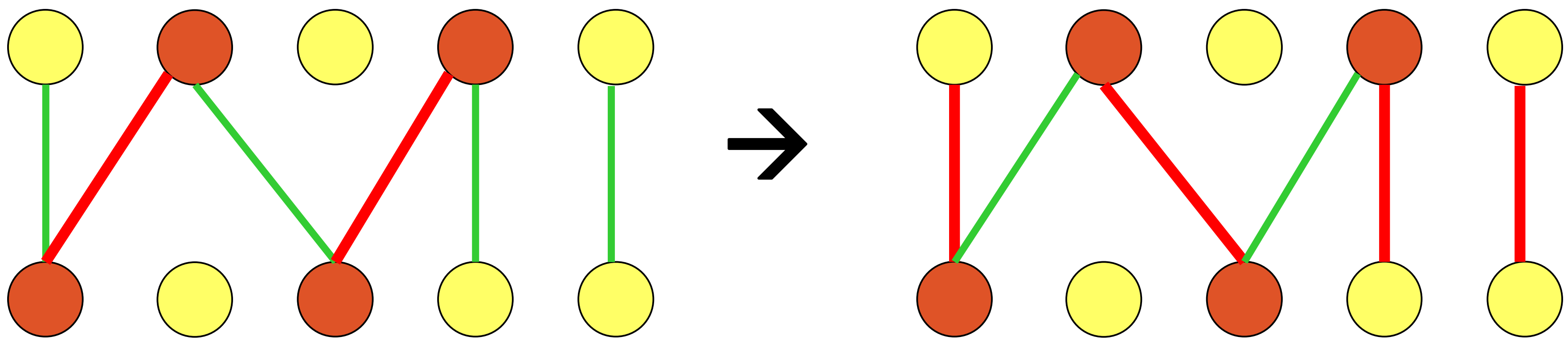
- Maximum matching tries to maximize the number of couples!

Applications

- Matching has many applications. For examples,
  - Comparing Evolutionary Trees
  - Finding RNA structure
  - …
- .

Idea

- “Flip” augmenting path to get better matching



- Note: After flipping, the number of matched edges will increase by 1!

Idea of Algorithm

- Start with an arbitrary matching
- While we still can find an augmenting path
  - Find the augmenting path P
  - Flip the edges in P

Thank You