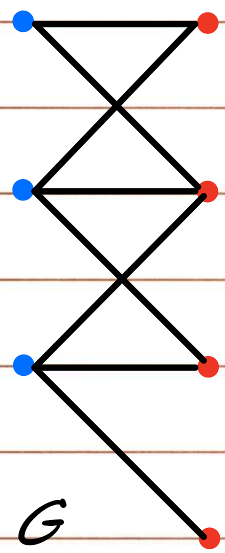
Network Flow

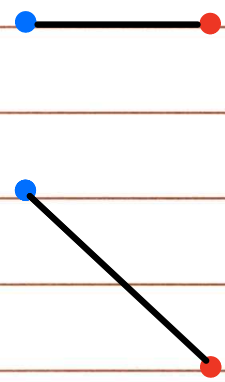
Bipartite Matching Problem

Recall that a bipartite graph $G=(V,E)$ is an undirected graph whose node set can be partitioned as $V = X \cup Y$ with the property that every edge $e \in E$ has one end in X and the other end in Y.

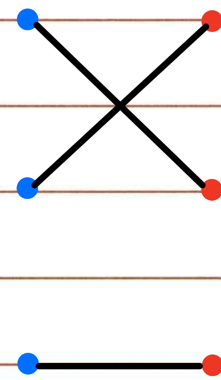
Def. A matching M in G is a subset of the edges $M \subseteq E$ such that each node appears in at most one edge in M.



G



A matching in G



A max. size matching in G

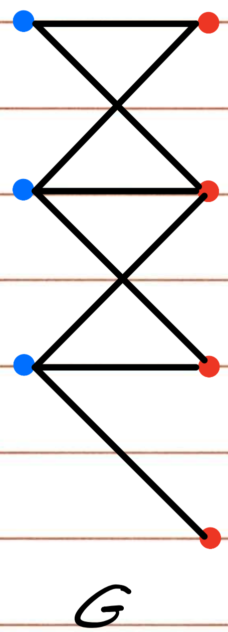
Problem Statement:

Given a bipartite graph G, find a matching of largest possible size in G.

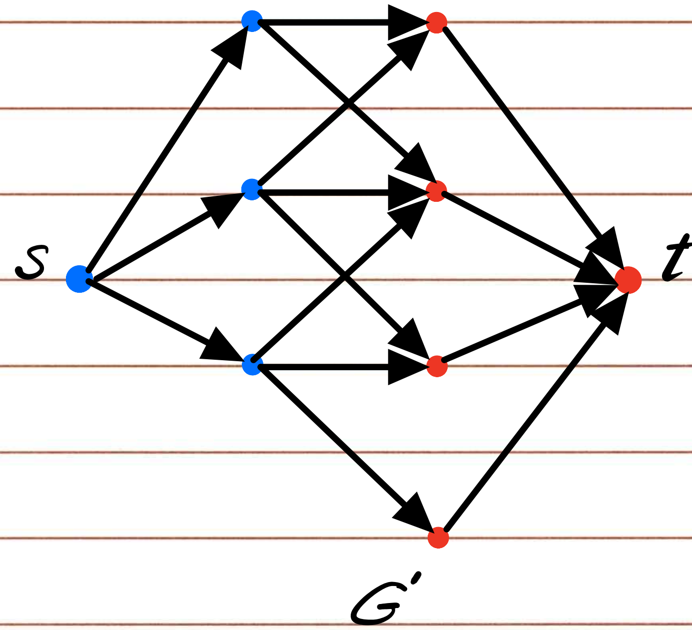
General Plan:

Design a flow network G' that will have a flow value $v(f) = k$ iff there exists a matching of size k in G. Moreover, flow f in G' should identify the matching M in G.

Construction of G′



Set C=1 for all edges



__Solution__

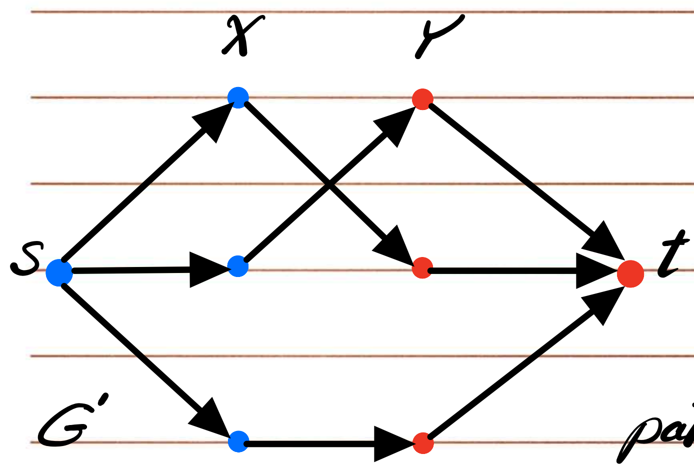
Find max. flow in G′. Say max. flow is f. Edges carrying flow between sets X and Y will correspond to our max. size matching in G.

__Proof Strategy__

To prove this, we will show that G′ will have a flow of value k iff G has a matching of size k.

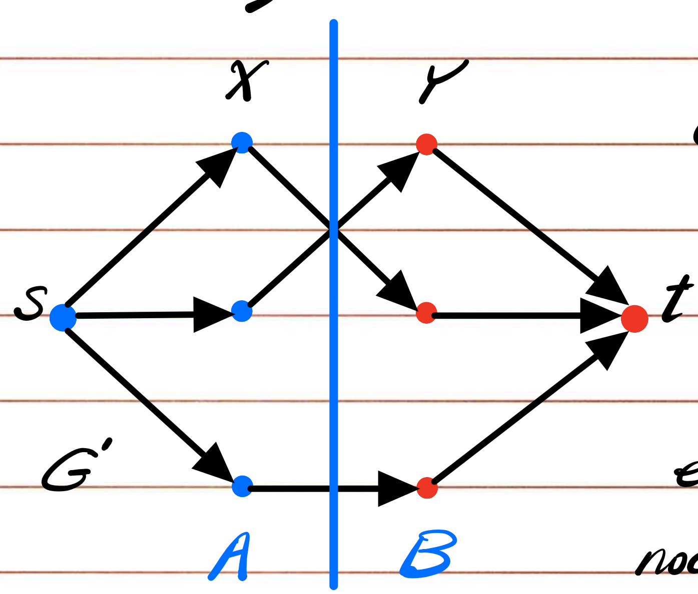
Proof:

A- If we have a matching of size k in G, we can find an s-t flow f of value k in G'.



Set C=1 for all edges. Using the matching of size k, we will find k independent s-t paths each with capacity 1.

B- If we have an s-t flow f of value k in G', we can find a matching of size k in G.



Given flow f of value k, there must be k edges on cut (A,B) that carry flow. These k edges cannot share any node, i.e. a matching of size k.

What is the complexity of the above solution if the basic Ford-Fulkerson alg. were used to find max. flow in G'?

Complexity of Ford-Fulkerson : $O(Cm)$

Since all edge capacities are set to 1, $C=O(n)$
So the complexity will be $O(mn)$
(Strongly Polynomial time)

Edge-Disjoint Paths Problem

Def. A set of paths is edge-disjoint if their edge sets are disjoint.

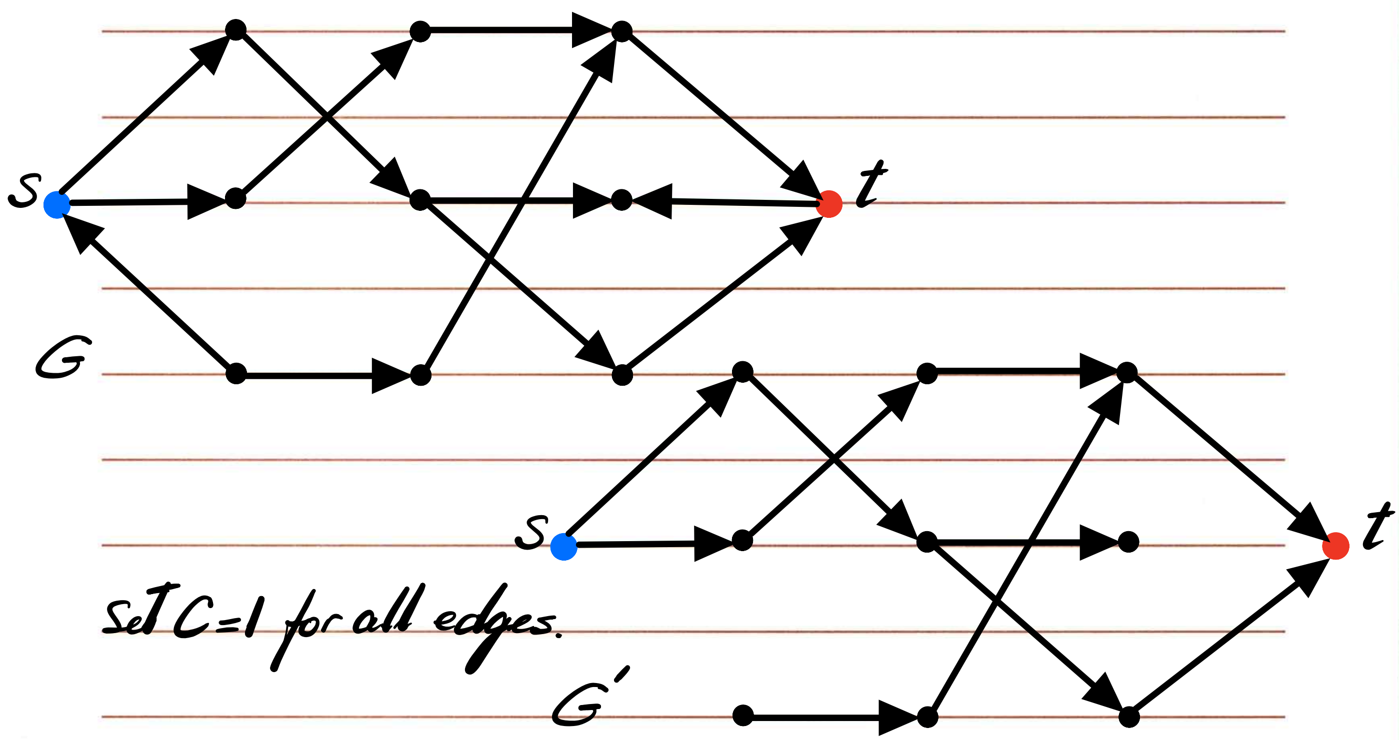
Problem Statement

Given a directed graph $G=(V,E)$ and two nodes s & $t \in V$, find the max. number of edge-disjoint s-t paths in G.

General Plan:

Design a flow network G' that will have a flow value $v(f) = k$ iff there exist k edge-disjoint s-t paths in G. Moreover, flow f in G' should identify the set of k edge-disjoint s-t paths in G.

Construction of G'



Set $C=1$ for all edges.

Solution:

- Find max flow f in G'.
- $v(f)$ will equal the max. number of edge-disjoint s-t paths in G.
- f will identify the edges on these paths.

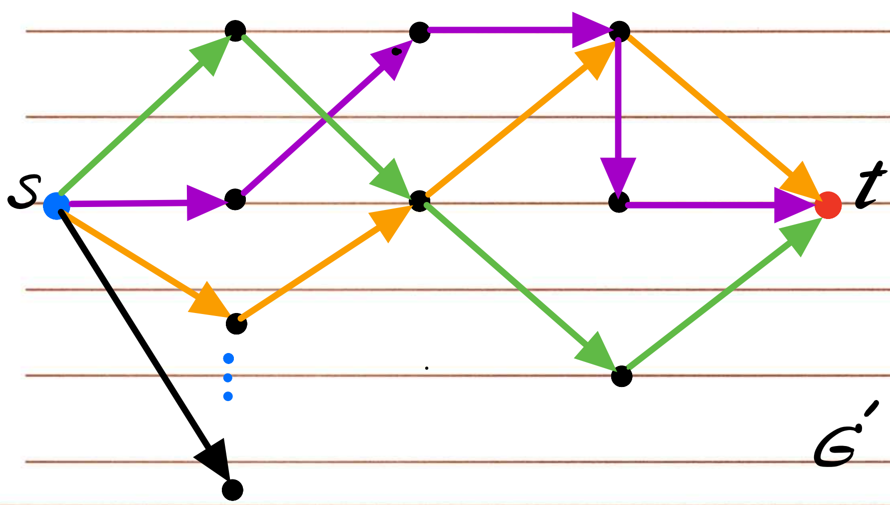
To prove this, we will show that there are k edge-disjoint paths in G iff there exists a flow of value k in G'.

<u>Proof</u>:

A- If we have k edge disjoint s-t paths in G, we can find a flow of value k in G'.

Obviously, since each path can carry 1 unit of flow (independently) from s to t in G'.

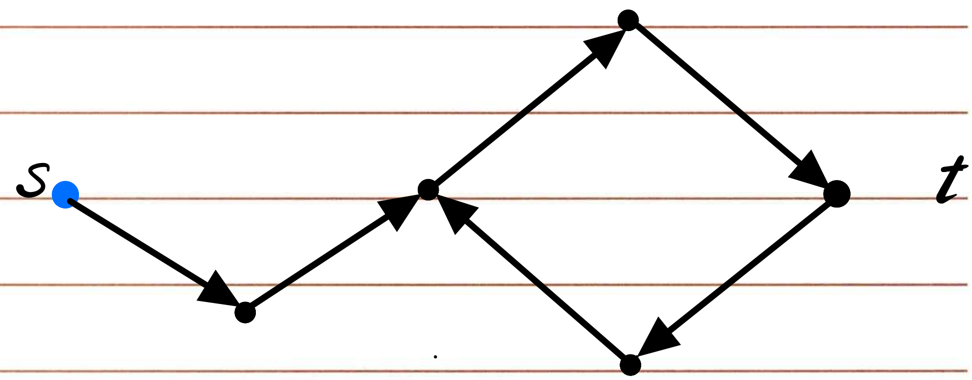
B- If we have a flow of value k in G', we can find k edge-disjoint s-t paths in G.

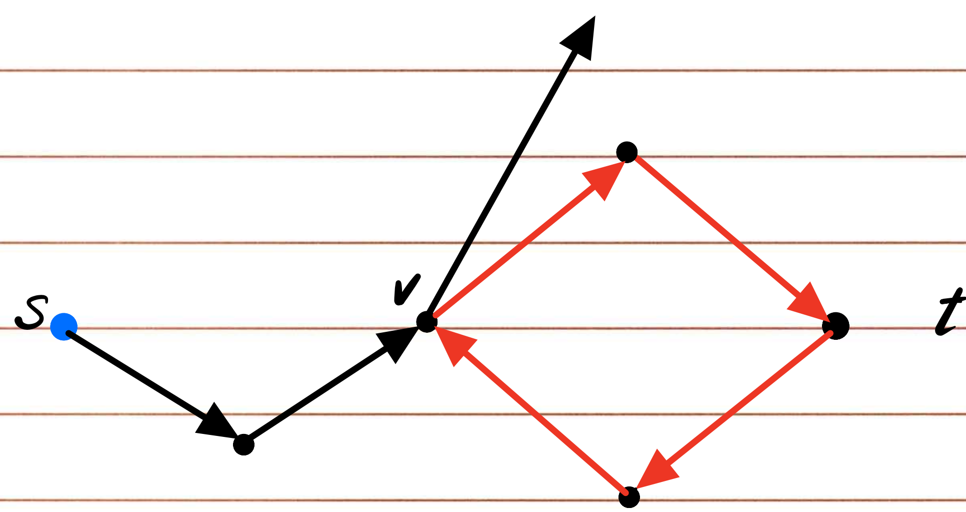


Using conservation of flow, we can follow each unit of flow leaving s and find the s-t path that carries that flow into t. We remove all edges on that path from G' and repeat the process.
Since there are k units of flow leaving s, we can find k edge-disjoint s-t paths.

Run time complexity if Ford-Fulkerson were used: $O(mn)$ as in previous solution

What if we run into a flow cycle on our path to t?

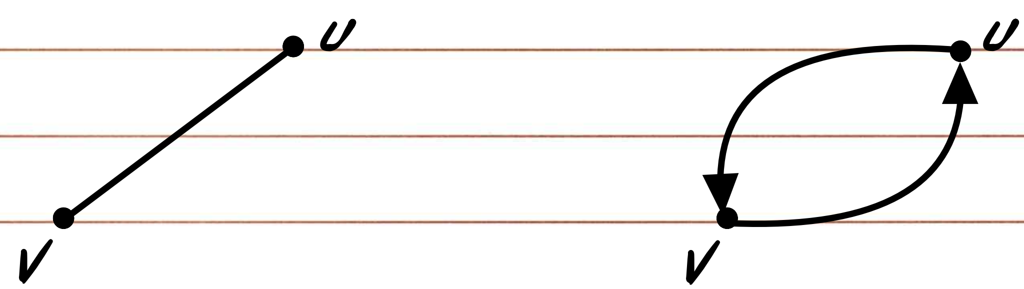




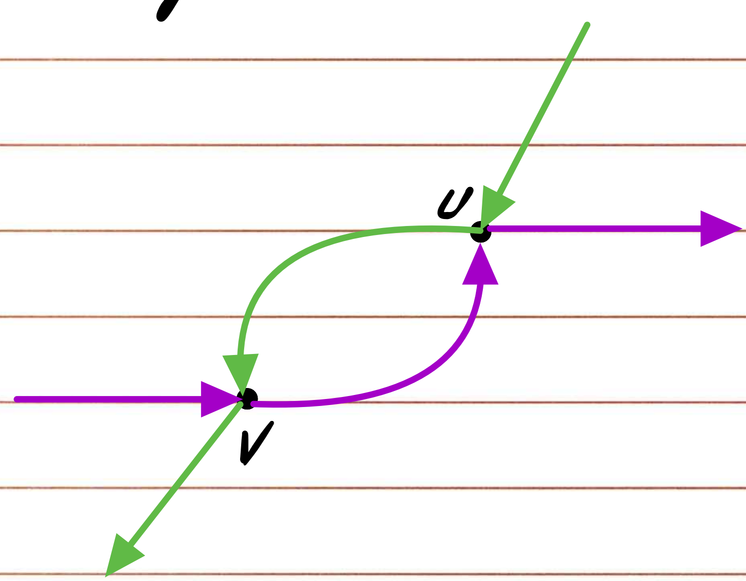
Since there are $\geq$ units of flow entering v, there must be $\geq$ units of flow leaving v. Remove the edges on the cycle and use the second edge out of v to continue toward t.

How can we apply this solution to undirected graphs?

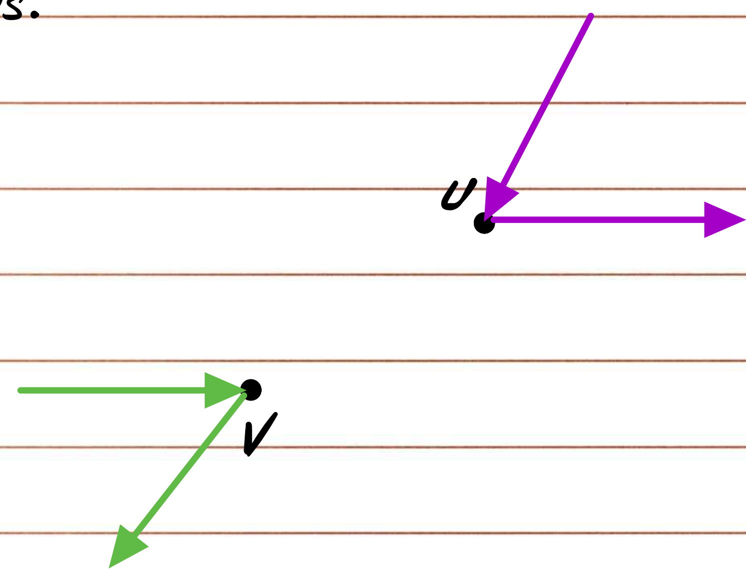
Represent each undirected edge with 2 directed edges in opposite directions.



But now we can end up with paths that use the edge vu twice!



To solve this problem, first remove all flow cycles on single edges before finding s-t paths.



Node-Disjoint Paths Problem

Def. A set of paths is node-disjoint if their node sets (except for starting and ending nodes) are disjoint.

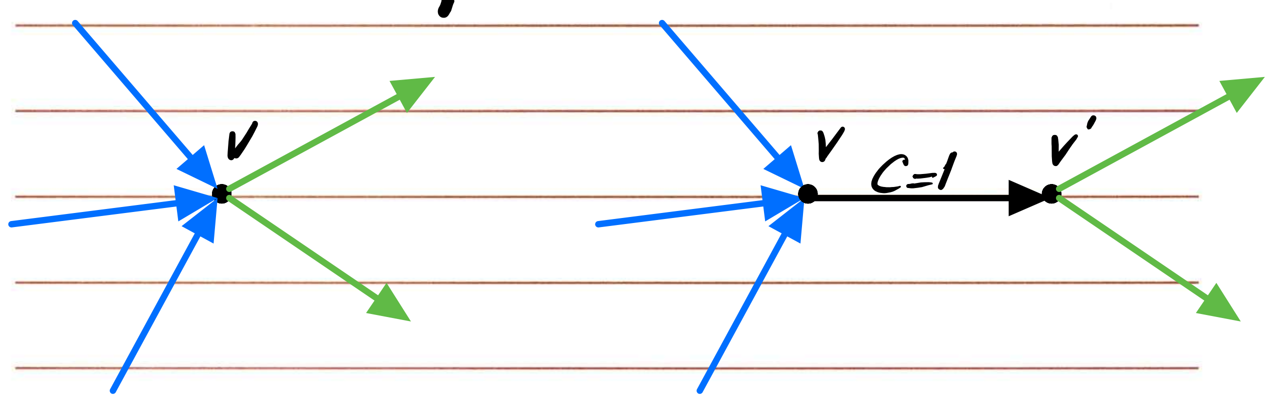
Problem Statement

Given a directed graph $G = (V, E)$ and two nodes s & $t \in V$, find the max. number of node-disjoint s-t paths in G.

General Plan: (As in previous solutions)

The only new challenge here is to design G'
such that our paths do not share any nodes.

Construction of G':



node v in G   node v in G'

Circulation &
Circulation with lower bounds

Def. A Circulation network is a directed graph $G=(V,E)$ with following features:

- Each edge e has a non-negative capacity c_e
- Associated with each node $v \in V$ is a demand d_v
- if $d_v > 0$, node v has demand of d_v for flow (Sink)
- if $d_v < 0$, node v has a supply of $|d_v|$ for flow (Source)
- if $d_v = 0$, v is neither a sink nor a source

Notation

We will call $f(e)$ Circulation through edge $\underline{e}$.
$f(e)$ has the following properties:

1- Capacity Constraint:
For each edge $e \in E$, $0 \leqslant f(e) \leqslant c_e$

2- Demand Condition:
For each node $v \in V$, $\sum_{e \text{ into } v} f(e) - \sum_{e \text{ out of } v} f(e) = d_v$
or $f^{in}(v) - f^{out}(v) = d_v$

FACT: If there is a feasible circulation with demands $\{d_v\}$, then $\sum_v d_v = 0$

Proof:
$$f^{in}(v) - f^{out}(v) = d_v$$
$$\sum_v d_v = \sum_v \left(f^{in}(v) - f^{out}(v) \right)$$

Consider edge $\underline{e}$ $\quad v \xrightarrow[f(e)]{e} u$

How does $f(e)$ contribute to
$\sum_v \left(f^{in}(v) - f^{out}(v) \right)$?

For each edge $\underline{e}$, $f(e)$ contributes to

$$\sum_{v} \left(f^{in}(v) - f^{out}(v) \right)$$

- once as a positive value (flow into v)
- once as a negative value (flow out of v)

So, $f(e)$'s cancel each other out and therefore, $\sum_{v} \left(f^{in}(v) - f^{out}(v) \right) = 0$

$$\sum_{v} d_v = \sum_{v} \left(f^{in}(v) - f^{out}(v) \right)$$

$$\sum_{v} d_v = 0$$

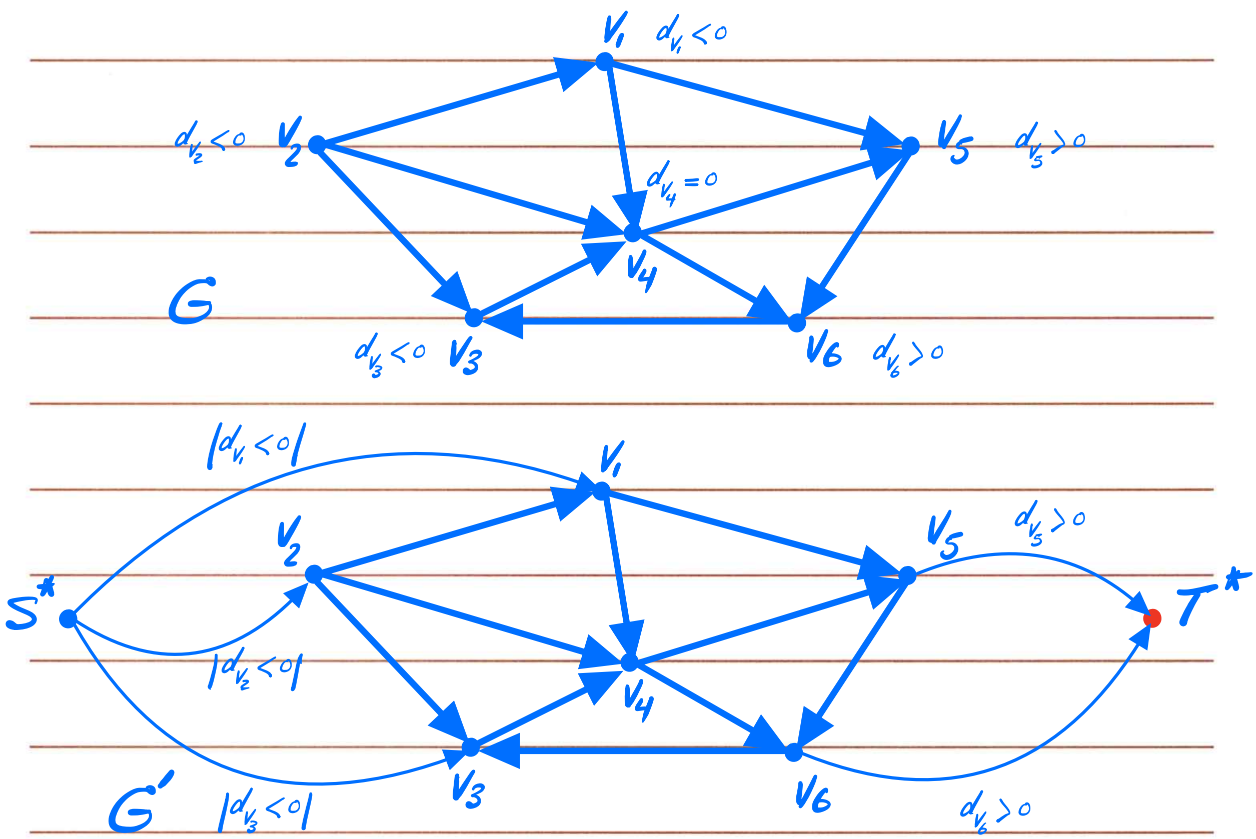
$$\sum_{v:d_v>0} d_v = \sum_{v:d_v<0} -d_v = D$$

Total demand value

Problem Statement

Given a circulation network G with demands $\{d_v\}$, is there a feasible circulation in G?

Solution: We will reduce the feasible circulation problem to Max Flow.



Find max flow f in G'

Can we have $v(f) < D$?

Yes. This will indicate that no feasible circulation with demands $\{d_v\}$ exists in G.

Can we have $v(f) > D$?

No! The s-t cuts closest to S^* and T^* each have a capacity of D.

Can we have $v(f) = D$?

Yes. And this will indicate that there is a feasible circulation in G.

Proof Template

A) If there is a feasible circulation f with demand values $\{d_v\}$ in G, we can find a max flow of value D in G'.

explain how ...

B) If there is a max flow of value D in G', we can find a feasible circulation f with demand values $\{d_v\}$ in G.

explain how ...

Def. A Circulation network with lower bounds is a directed graph $G = (V, E)$ with following features:

- Each edge $\underline{e}$ has a non-negative capacity c_e and a lower bound constraint on flow l_e.
- Associated with each node $v \in V$ is a demand d_v
- if $d_v > 0$, node v has demand of d_v for flow (Sink)
- if $d_v < 0$, node v has a supply of $|d_v|$ for flow (Source)
- if $d_v = 0$, v is neither a sink nor a source

Notation

We will call $f(e)$ Circulation through edge $\underline{e}$.
$f(e)$ has the following properties:

1- Capacity Constraint:
For each edge $e \in E$, $l_e \leqslant f(e) \leqslant c_e$

2- Demand Condition:
For each node $v \in V$, $f^{in}(v) - f^{out}(v) = d_v$

Problem Statement

Given a circulation network G with demands $\{d_v\}$ and lower bounds $\{l_e\}$, is there a feasible circulation in G?

Solution: We will reduce the feasible circulation with lower bounds problem to the feasible circulation problem.

Solution:

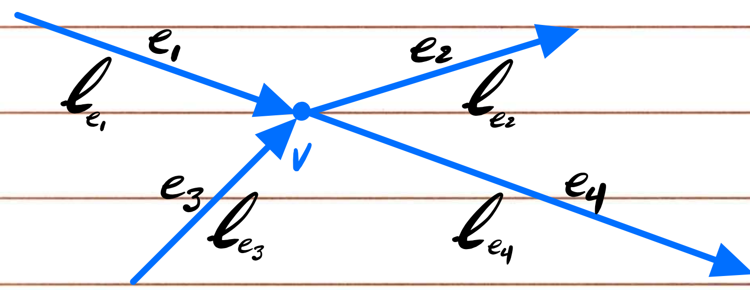
We will find the feasible circulation (if it exists) in two passes.

- Pass #1. Find f_0 to satisfy all l_e's.

- Pass #2. Use the remaining capacity of the network to find a feasible circulation f_1 (if it exists)

- Combine the two flows: $f = f_0 + f_1$

Consider node v in G and its incident edges:



Since $f_0(e_i) = \ell_{e_i}$, we will have a flow imbalance at node v due to f_0.

$$f_0^{in}(v) - f_0^{out}(v) = \sum_{e \text{ into } v} \ell_e - \sum_{e \text{ out of } v} \ell_e = L_v$$

L_v is called the flow imbalance at node v.

Solution outline

1- Push flow f_0 through G, where
$f_0(e) = l_e$

2- Construct G', where $C'_e = C_e - l_e$
and $d'_v = d_v - L_v$

3- Find a feasible circulation in G'.
(if it exists, call this f_1)

4- If in step 3 there is no feasible circulation in G' $\Rightarrow$
There is no feasible circulation in G.

Otherwise,
feasible circulation in $G = f_0 + f_1$

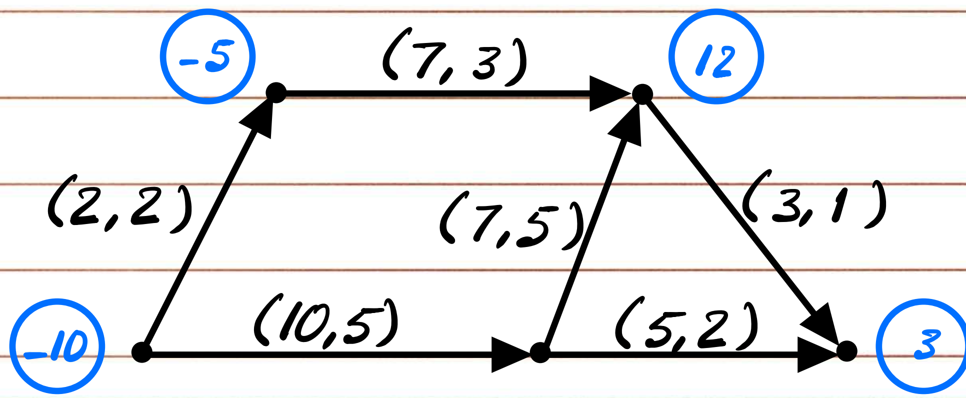
Numerical Example on next page

(x, y) indicates

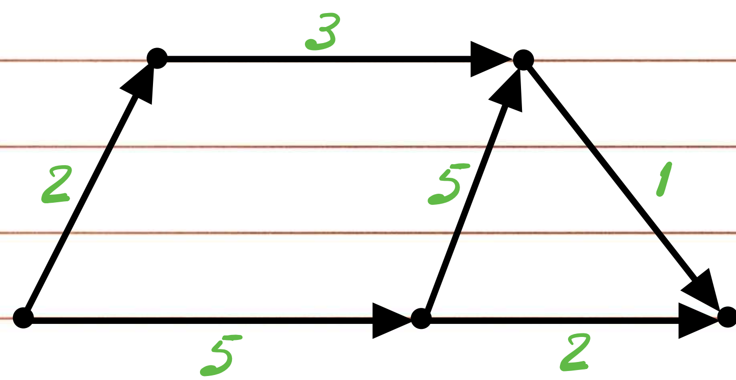
- edge capacity of x
- lower bound flow of y

z indicates demand value of z

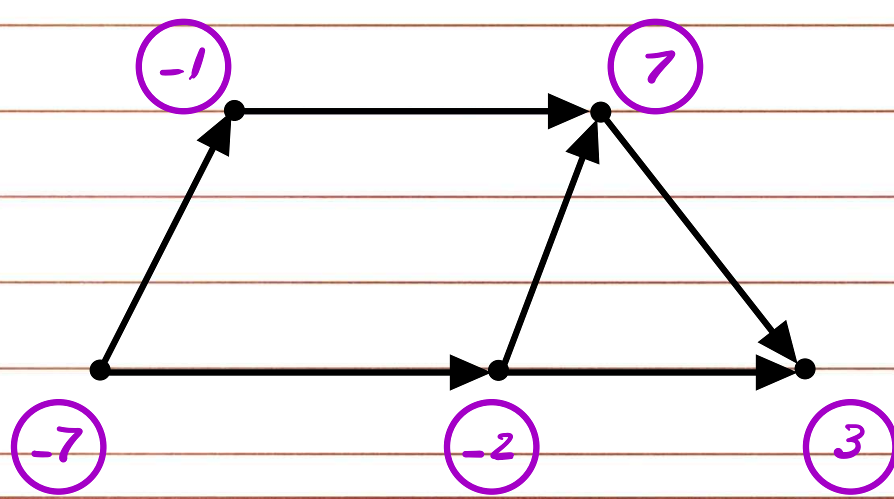
G

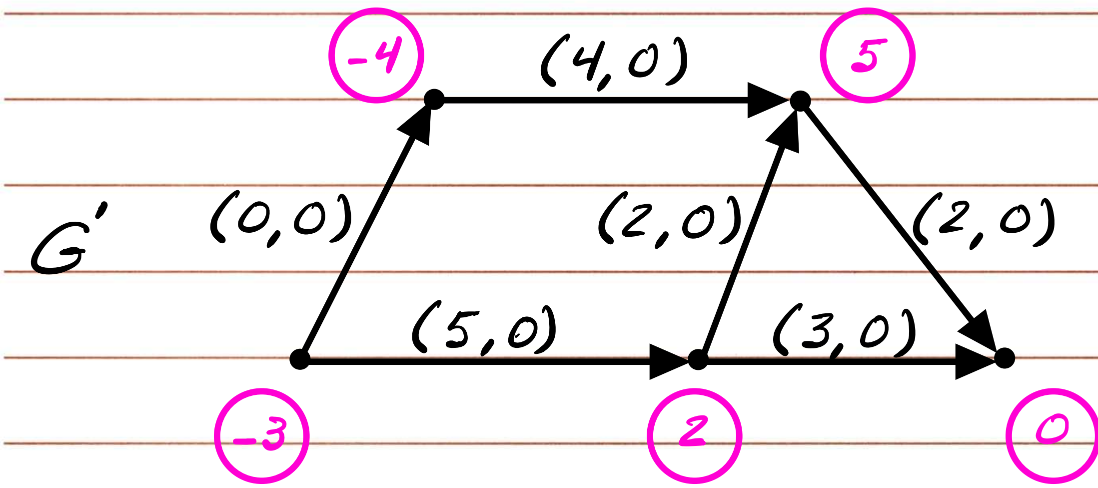


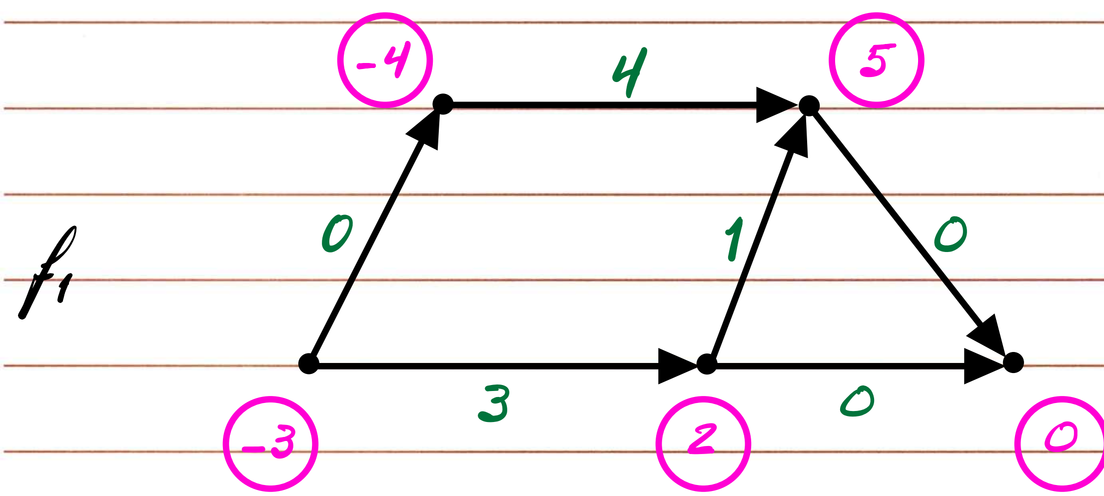
f_0



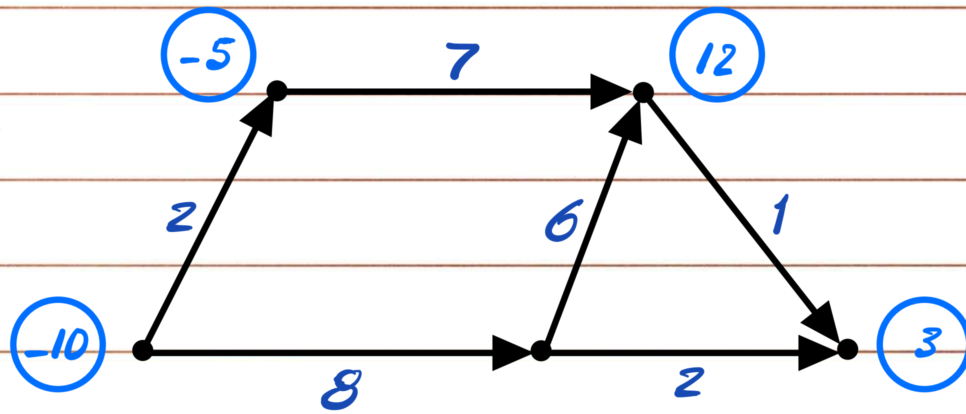
L_v's




G'
-4
5
(4,0)
(0,0)
(2,0)
(2,0)
(5,0)
(3,0)
-3
2
0


f1
-4
5
4
0
1
0
3
0
-3
2
0

$f_0 + f_1$



Survey Design Problem

Input: - Information on which customer purchased which products
- Maximum and minimum number of questions to send to customer i. (m_i, l_i)
- Maximum and minimum number of questions to ask about a product j. (m_j, l_j)