Flow Networks

Topics

Flow Networks
Residual networks
Ford-Fulkerson's algorithm
Ford-Fulkerson's Max-flow Min-cut Algorithm

Flow Networks

A directed graph can be interpreted as a flow network to analyse material flows through networks.

Material courses through a system from a source (where it is produced) to a sink (where it is consumed).
Examples :

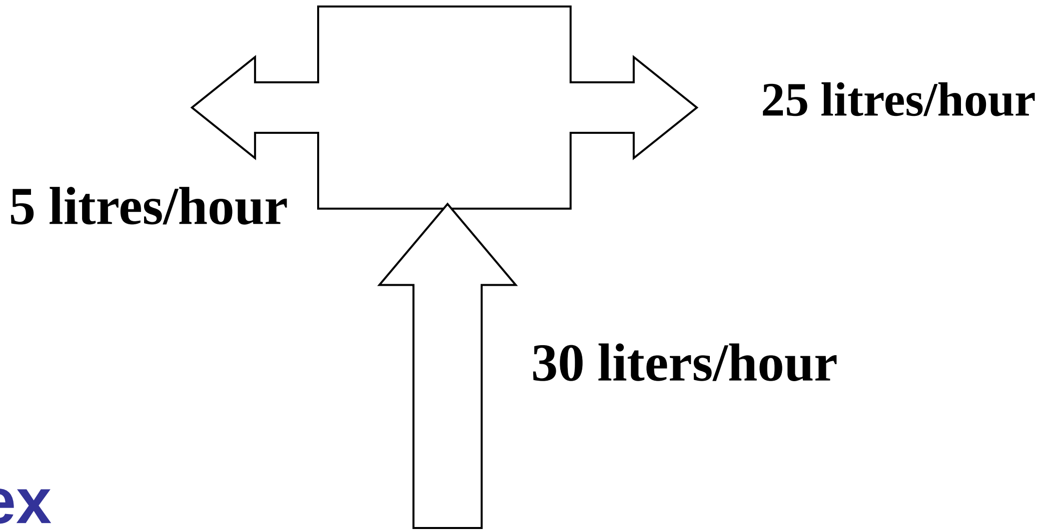
- **Water through pipelines**
- **Newspapers through distribution system**
- **Electricity through cables**
- **Cars on a production line**
  - **on roads**

The source produces the material at a steady rate .
The sink consumes the material at a steady rate

Flow: the rate at which the material moves from one point to another

100 litres of water per hour in a pipe

30 Amperes of electric current in a circuit

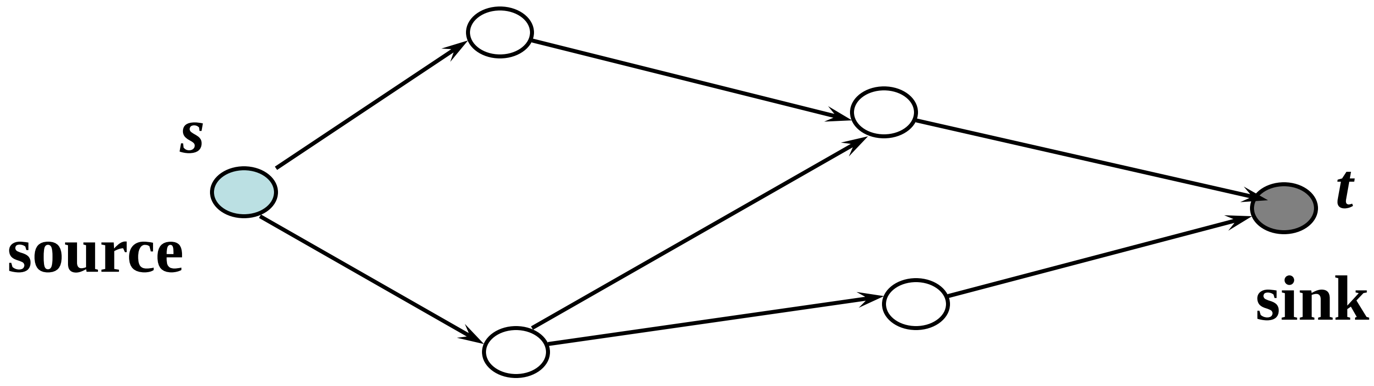


The rate at which a material enters a vertex = the rate at which the material leaves the vertex

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

A flow network has a source vertex s, and a sink vertex t.

For every vertex $v \in V$ there is a path from s to v and v to t in a connected graph.



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

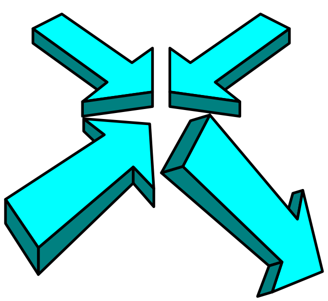
1. **Capacity constraint** : For all $u,v \in V$, we require $f(u,v) \le c(u,v)$.
The net flow from one vertex to another must not exceed the given capacity.

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

The net flow from a vertex u to a vertex v is the negative of the net flow in the reverse direction.
The net flow from a vertex to itself is zero for all $u \in V$, that is $f(u,u) = 0$.

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

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



The total net flow out of a vertex other than the source or sink is zero.

The quantity *f(u,v)* can be negative or positive, it is called the net flow from vertex *u* to *v*.

The value of a flow is defined as

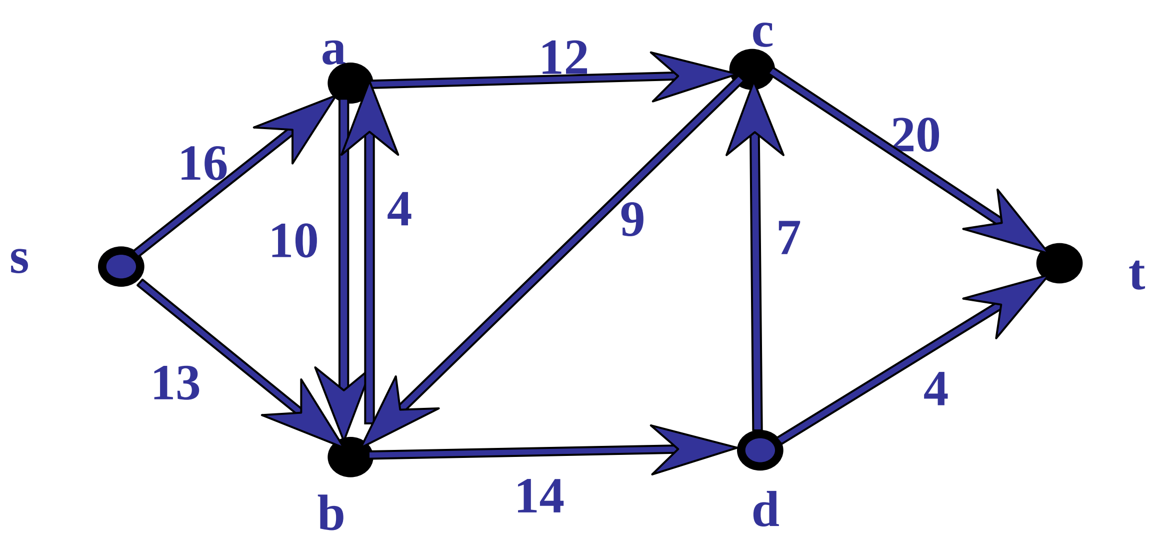
$$|f| = \sum_{v \in V} f(s,v)$$

In the maximum-flow problem, we are given a flow network G with source *s* and sink *t*, and we wish to find a flow of maximum value from *s* to *t*.

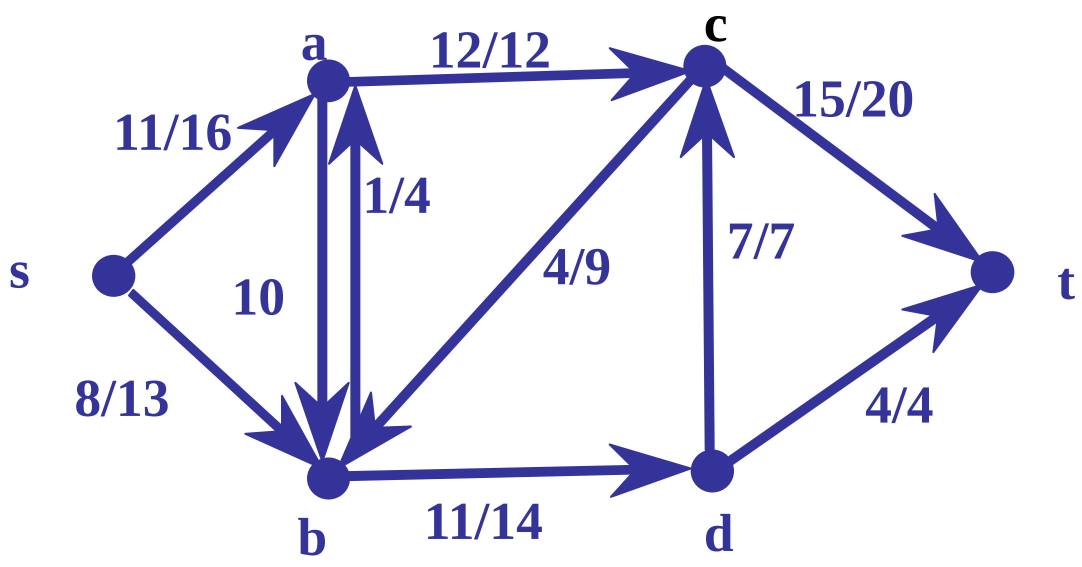
There is no net flow between *u* and *v* if there is no edge between them.
If $(u,v) \notin E$ and $(v,u) \notin E$, then $c(u,v) = c(v,u) = 0$.
Hence, the capacity constraint, $f(u,v) \leq 0$ and $f(v,u) \leq 0$.
By skew symmetry, $f(u,v) = -f(v,u)$,
therefore, $f(u,v) + f(v,u) = 0$.

Nonzero net flow from vertex *u* to vertex *v* implies that $(u,v) \in E$ or $(v,u) \in E$ (or both).

Consider the network *G=(V,E)* shown in the figure below. The network is for a transport system that transports crates of an item from source vertex *s* to sink vertex *t* through a number of intermediate points. Each edge *(u,v)* $\in$ *E* in the network is labeled with its capacity *c(u,v).*



Let us consider a flow in *G*, $|f|$=19
If *f(u,v)* >0, edge (*u,v*) is labeled *f(u,v)/c(u,v)*
If *f(u,v)* ≤ 0, the edge is labeled by its capacity only.



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

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

Initially, c (a ,b) = 8, and c (b, a) = 3 -- Fig. a.
f (a, b) = 5 and f (b, a) = 2, -- Fig. b
the net flow is shown as 3/8 in direction a to b – Fig. c

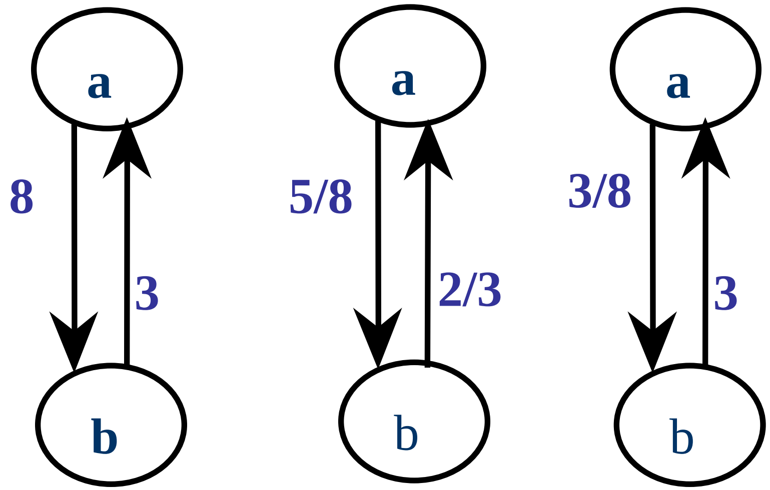
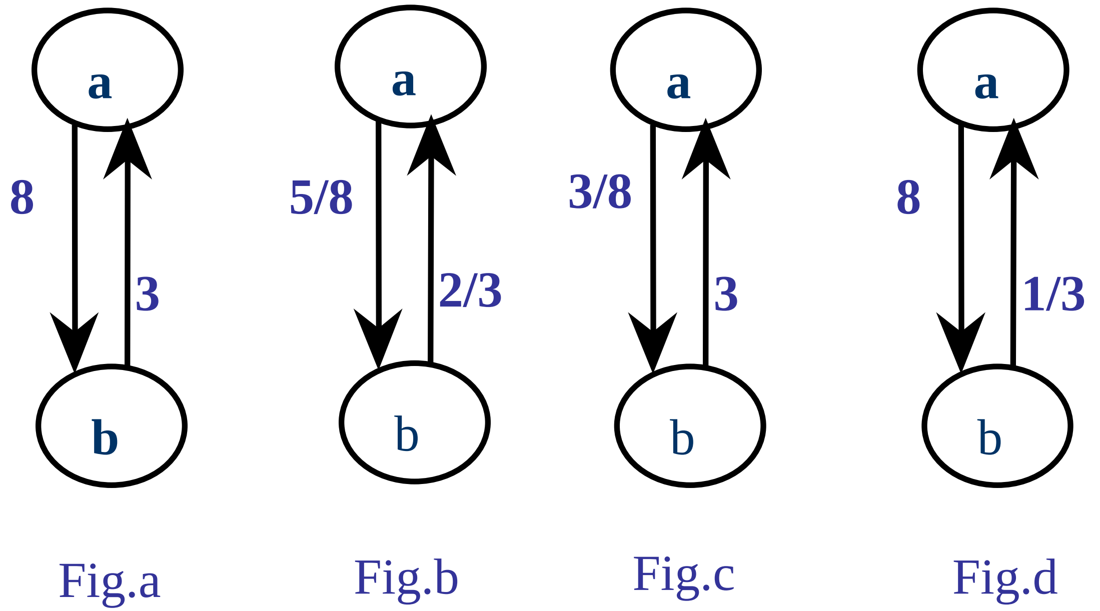


Fig.a Fig.b Fig.c



If we increase the flow from b to a from 2 to 6 then the net flow is 1/3 in the direction b to a as shown in Fig. d.

The Ford_Fulkerson method

The method is iterative,
Starts with $f(u,v)$ for $(u,v) \in V$, initial flow of value 0.
The method is based on the augmenting path which is defined as a path from s to t along which we can push more flow and then augment flow along this path.

Procedure Ford_Fulkerson_method(G,s,t)

1. $f \leftarrow 0$;
2. **while** there exists an augmenting path p
3. **do** augment flow along path p
4. **return f**

Residual Networks

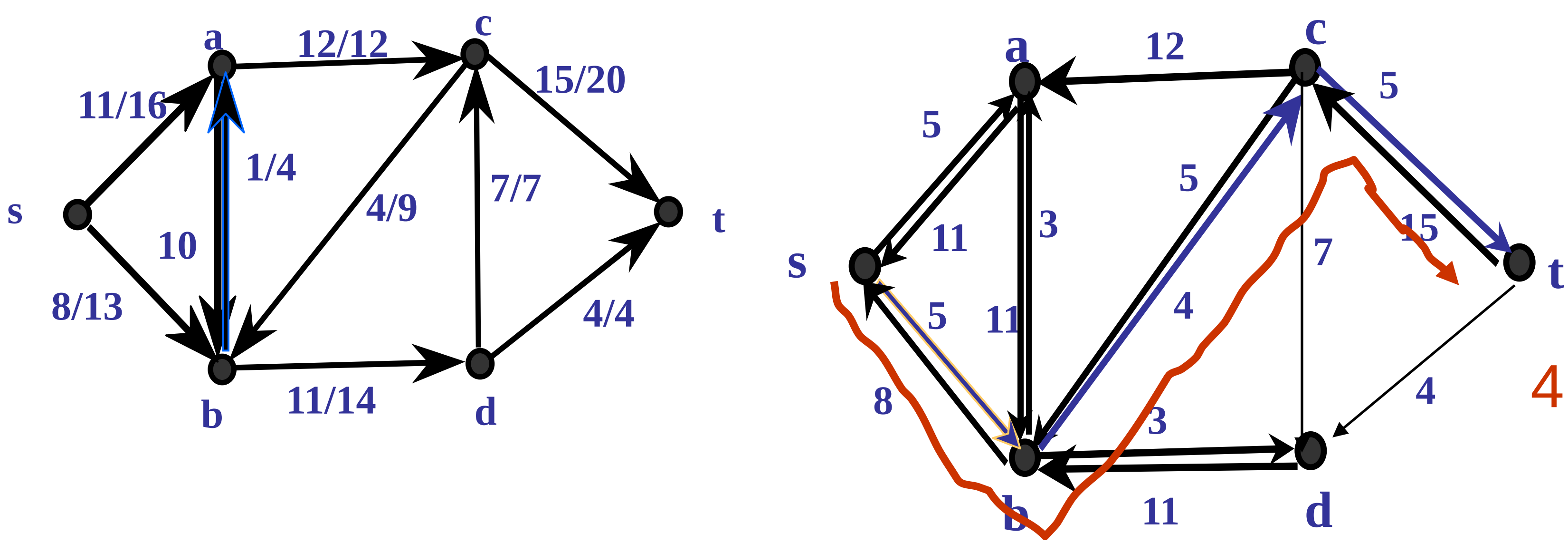
Consider a flow network *G(V,E)* with source *s* and sink *t* and let *f* be a flow in *G*.
Consider a pair of vertices $u,v \in V$.
Residual capacity between u and v is given by

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

- **the additional net flow we can push from *u* to *v* before exceeding the capacity.**
  For example, if $c(u,v) = 25$ and $f(u,v) = 19$, then $r(u,v) = 6$.

If $f(u,v) < 0$ then $r(u,v) > c(u,v)$

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



Each edge in the residual network can admit positive net flow only.

The residual network may include several edges that are not in the original network, $(u,v) \in E_f$ and $(u,v) \notin E$ is possible (E_f is not a subset of E). However, (u,v) appears in G_f only if $(v,u) \in E$ and there is a positive flow from v to u. Because the net flow $f(u,v)$ is negative,

$$r(u,v) = c(u,v)-f(u,v) > 0 \text{ and } (u,v) \in E_f$$

An edge (u,v) can appear in a residual network only if at least one of (u,v) and (v,u) appears in the original network.

$$|E_f| \le 2|E|$$

Augmenting Paths

It is a simple path from s to t in G_f. Each edge (u,v) on an augmenting path admits some additional positive net flow from u to v without violating the capacity constraint on the edge. The residual capacity of a path p is given by,

$$r(p) = \min \{ r(u,v) : (u,v) \text{ is in } p \}$$

Let's define a flow function f_p,

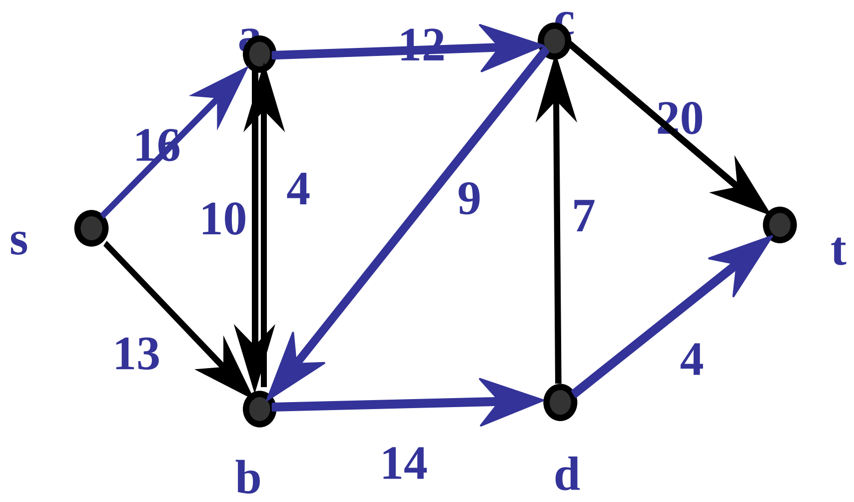
$$f_p = \begin{cases} r(p) \ \ if \ \ (u,v) \ \ is \ \ on \ \ p, \\ -r(p) \ \ if \ \ (v,u) \ \ is \ \ on \ \ p \\ 0 \ \ otherwise \end{cases}$$

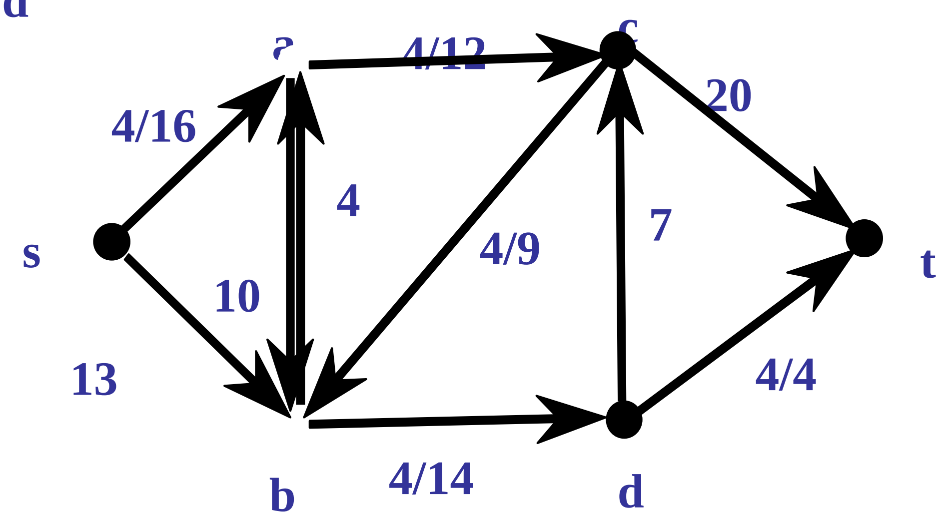
f_p is a flow in G_f with value $|f_p| = r(p) > 0$.
If we add f_p to f, we get another flow in G whose value is closer to the maximum.

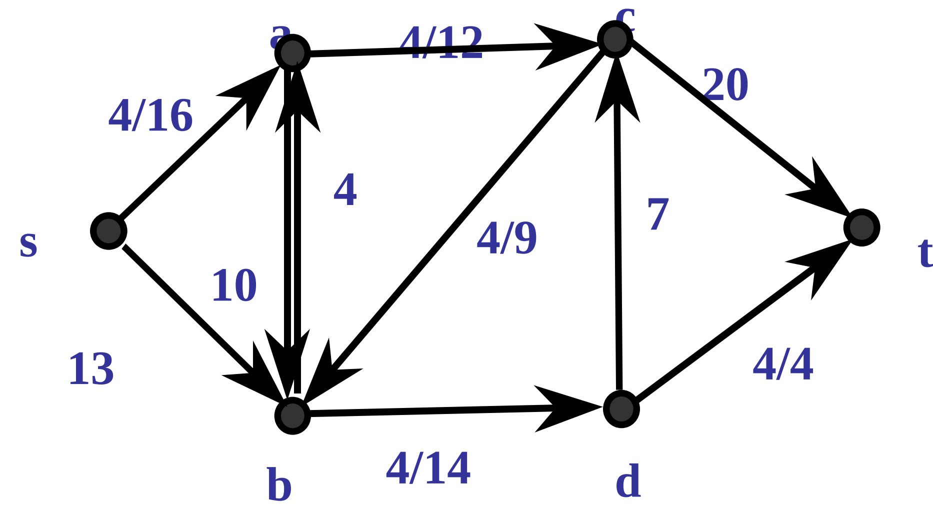
Algorithm

Procedure Ford-Fulkerson(G,s,t)
Input : Flow Network $G(V,E)$
Output : Maximum flow for the given network

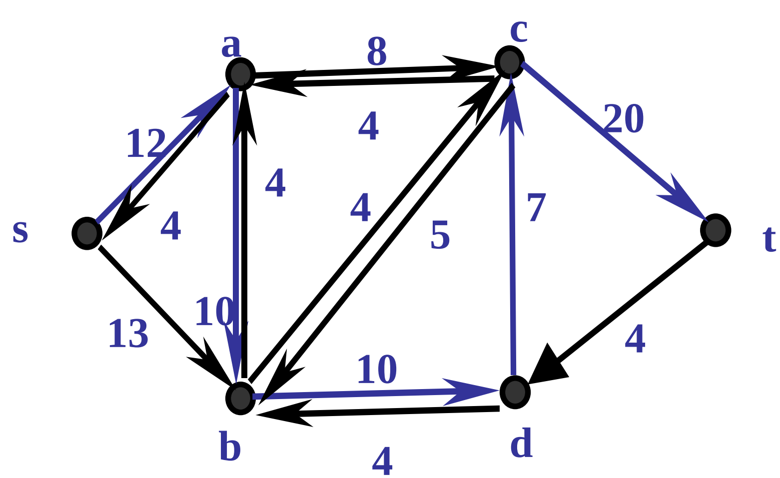
1. for each edge $(u,v) \in E$
2. do $f[u,v] \leftarrow 0$;
3. $f[v,u] \leftarrow 0$;
4. while there exists a path p from s to t in the residual network G_f
5. do $r(p) \leftarrow \min \{r(u,v) : (u,v) \text{ is in } p\}$;
6. for each edge (u,v) in p
7. do $f[v,u] \leftarrow - f[u,v]$;
8. $f[u,v] \leftarrow f[u,v] + r(p)$;
9. return


a
c
12
16
20
4
10
9
7
s
t
13
4
b
14
d

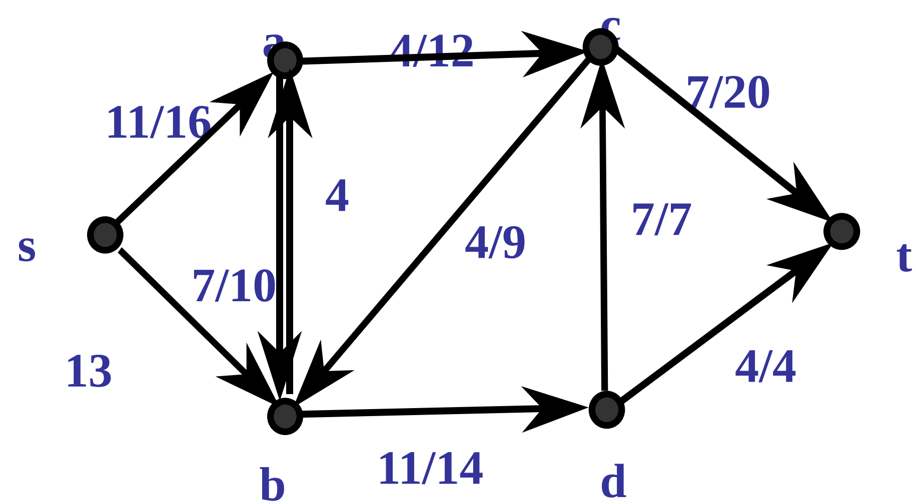

a
4/12
c
4/16
20
4
7
s
4/9
t
10
13
4/4
b
4/14
d



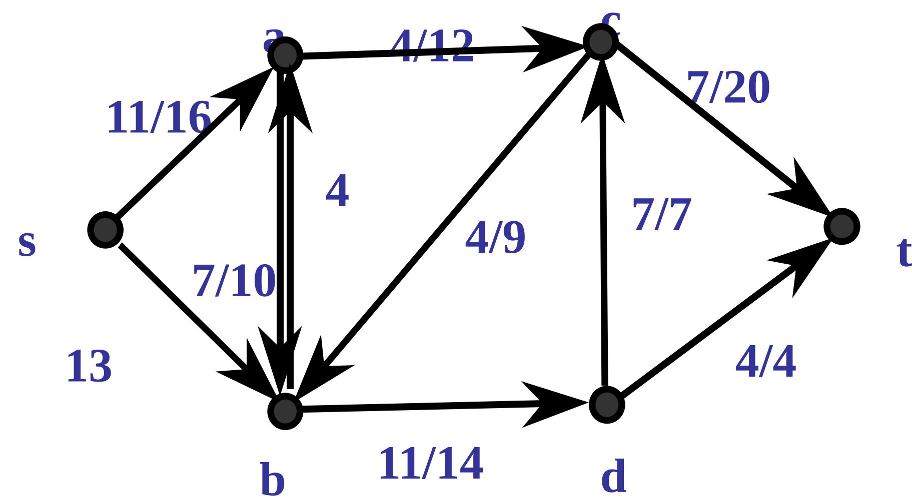
a
4/12
c
4/16
20
4
7
s
4/9
t
10
13
4/4
4/14
b
d

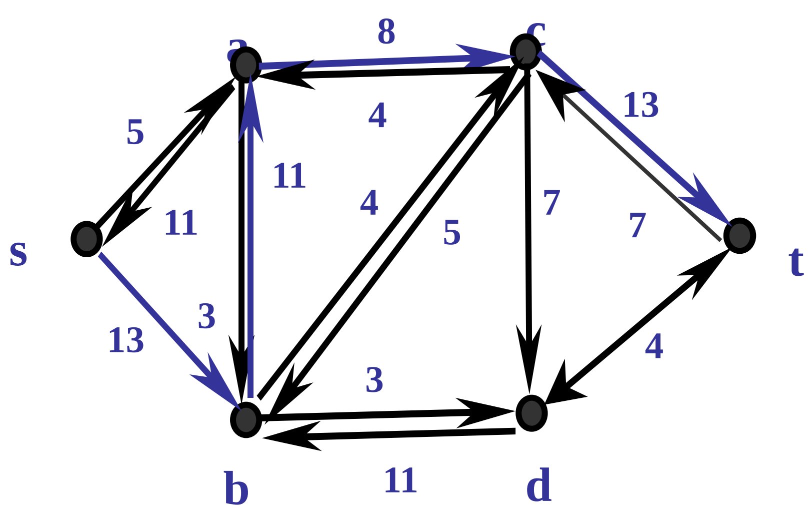


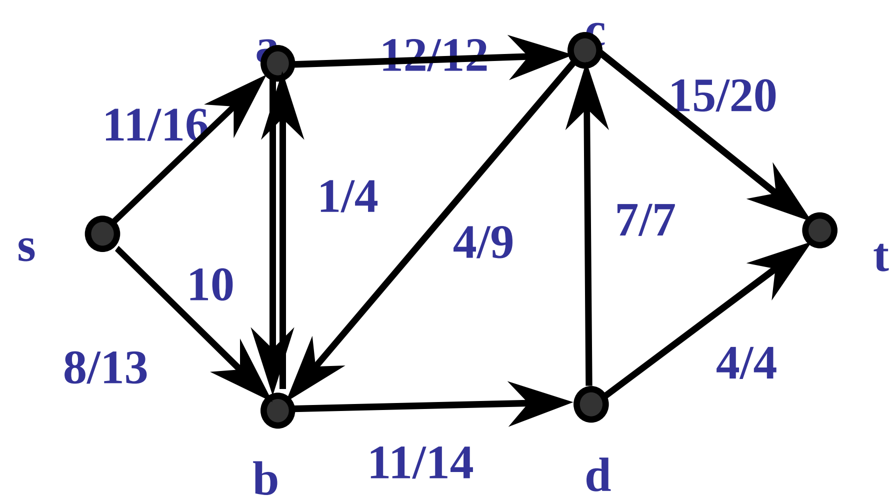
a
8
c
4
12
20
4
4
7
s
4
5
t
10
13
4
10
b
4
d

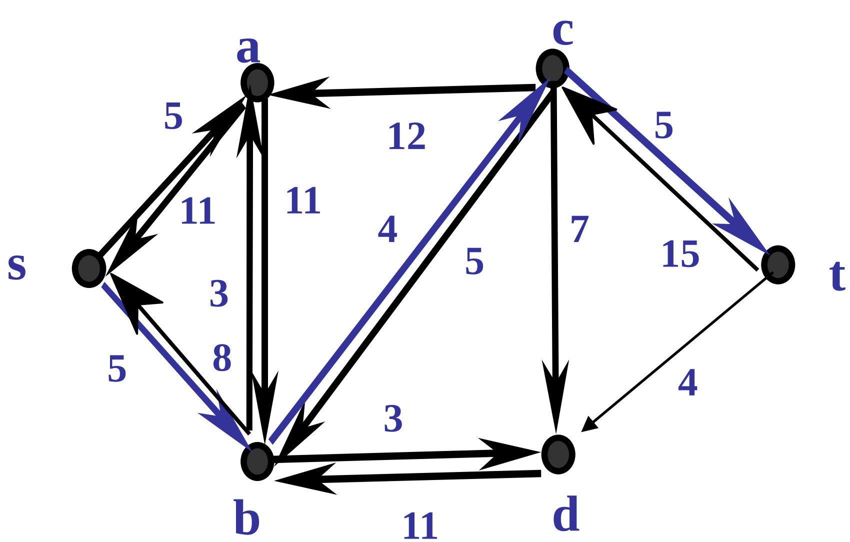


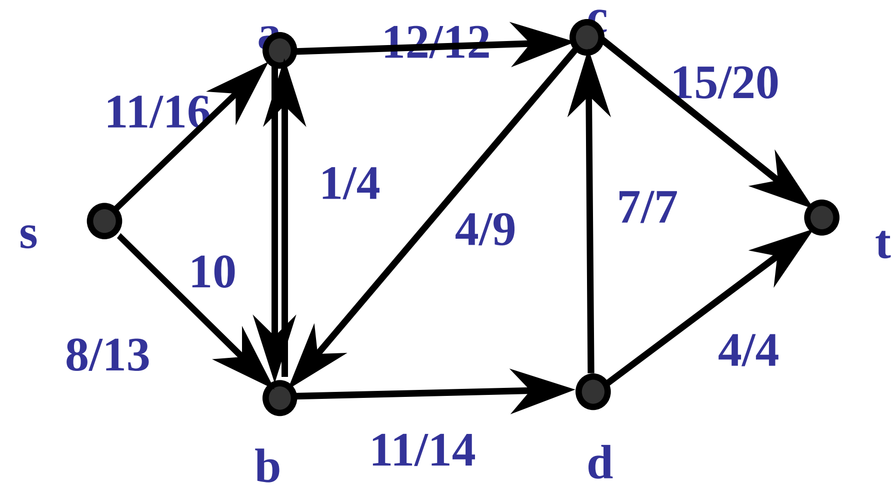
a
4/12
c
11/16
7/20
4
7/7
4/9
s
t
7/10
13
4/4
11/14
b
d

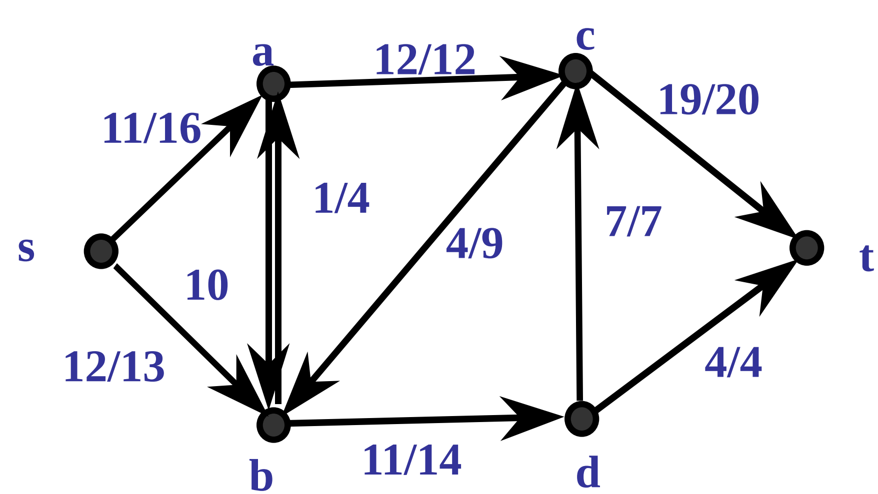

a
c
4/12
11/16
7/20
4
4/9
7/7
s
t
7/10
13
4/4
b
11/14
d

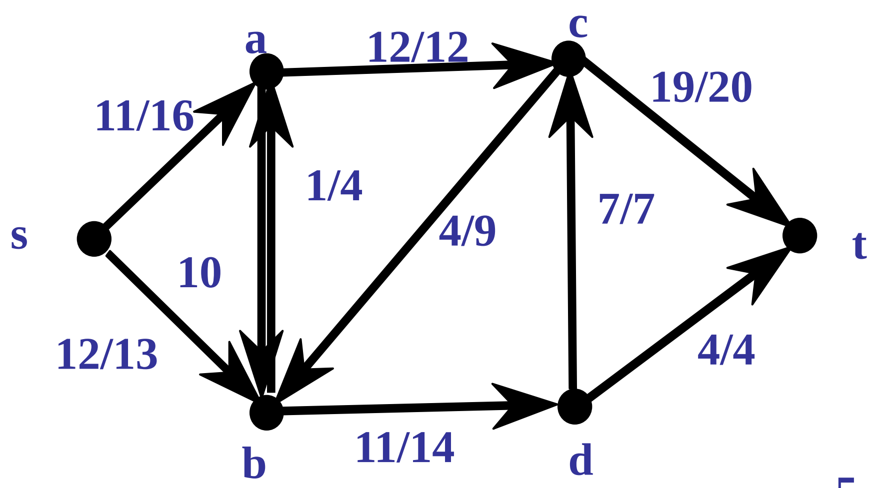

a
8
c
4
13
5
11
11
4
7
7
5
s
t
3
13
4
3
b
11
d

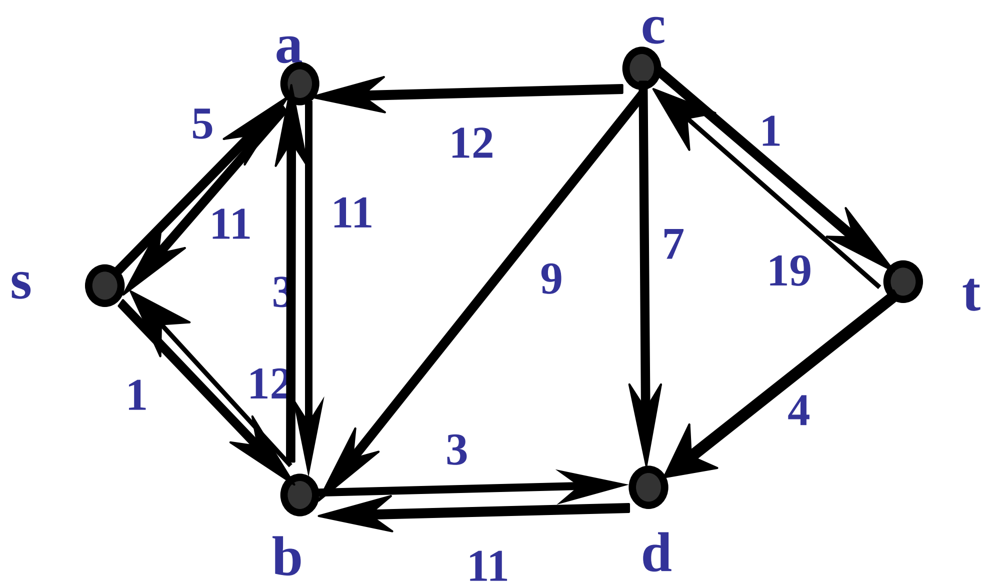

a
12/12
c
15/20
11/16
1/4
7/7
4/9
s
t
10
8/13
4/4
b
11/14
d


a
c
5
12
5
11
11
4
5
7
15
s
t
3
8
5
3
4
b
d
11


a
12/12
c
11/16
15/20
1/4
4/9
7/7
s
t
10
8/13
4/4
b
11/14
d


a
12/12
c
11/16
19/20
1/4
4/9
7/7
s
t
10
12/13
4/4
b
11/14
d

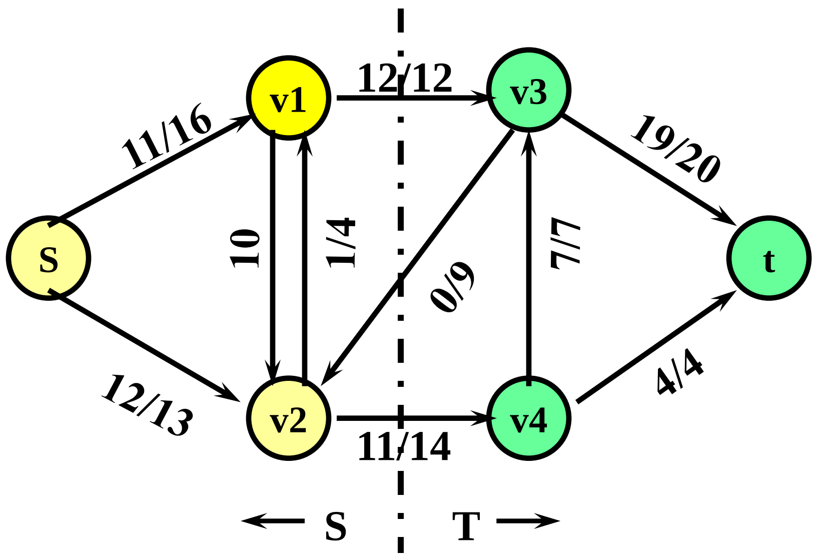

a
12/12
c
11/16
19/20
1/4
7/7
s
4/9
t
10
12/13
4/4
b
11/14
d


a
c
5
12
1
11
11
7
s
3
9
19
t
12
1
4
3
b
11
d

Ford Fulkerson – cuts of flow networks

New notion: cut (S,T) of a flow network

A cut (S,T) of a flow network $G=(V,E)$ is a partition of V in to S and $T = V \setminus S$ such that $s \in S$ and $t \in T$.



In the example:

$S = \{s,v1,v2\}$, $T = \{v3,v4,t\}$

Net flow $f(S,T) = f(v1,v3) + f(v2,v4) + f(v2,v3)$

$= 12 + 11 + (-0) = 23$

Capacity $c(S,T) = c(v1,v3) + c(v2,v4)$

$= 12 + 14 = 26$

Implicit summation notation: $f(S,T) = \sum_{u \in S} \sum_{v \in T} f(u,v)$

Cuts of Flow slides prepared by Shwetha and Pradeep

Ford Fulkerson – cuts of flow networks

Assumption:

The value of any flow f in a flow network G is bounded from above by the capacity of any cut of G

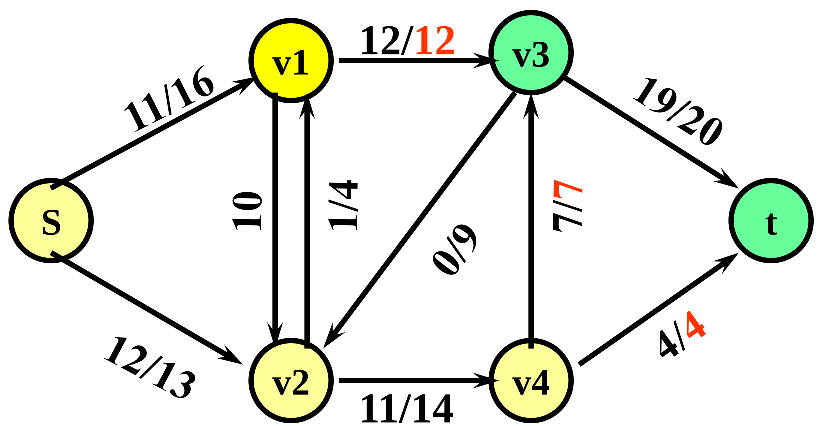
Lemma: $\boxed{|f| \leq c(S, T)}$

$$|f| = f(S, T)$$

$$= \sum_{u \in S} \sum_{v \in T} f(u, v)$$

$$\leq \sum_{u \in S} \sum_{v \in T} c(u, v)$$

$$= c(S, T)$$



Cuts of Flow slides prepared by Shwetha and Pradeep

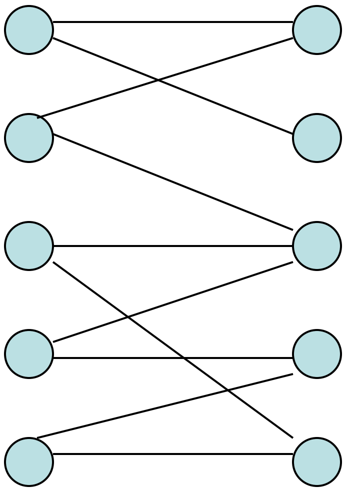
Exercise Set 6

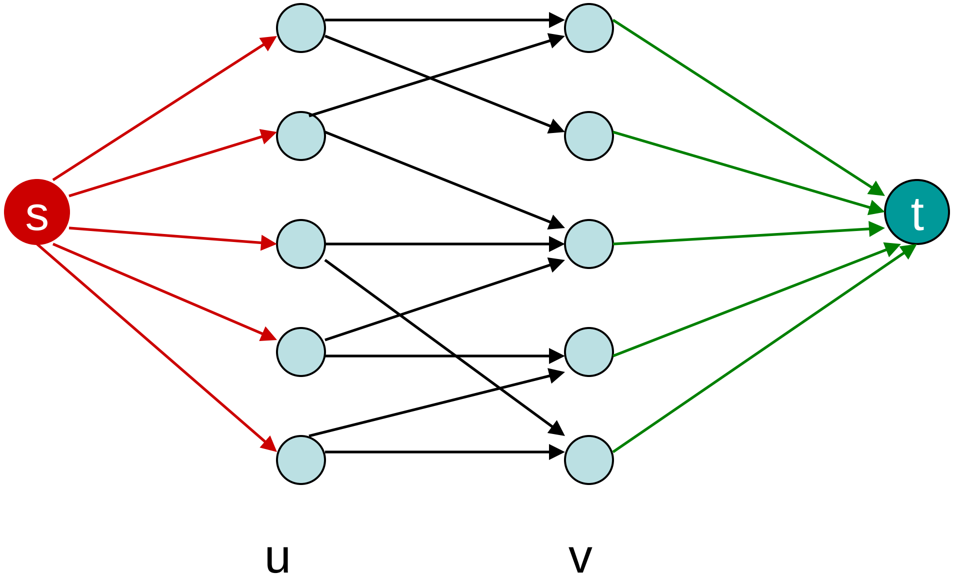
- Suppose that each source s_i in a multisource, multisink problem produces exactly p_i units of flow, so that $f(s_i,V) = p_i$. Suppose that each sink t_j consumes exactly q_j units so that $f(V,t_j) = q_j$, where . Show how to convert the problem of finding a flow f that obeys these additional constraints into the problem of finding a maximum flow in a single-source, single-sink flow network.
- Given a flow network $G = (V, E)$, let $f1$ and $f2$ be functions from $V \times V$ to **R**. The flow sum $f1 + f2$ is the function from $V \times V$ to **R** defined by $(f1 + f2)(u, v) = f1(u, v) + f2(u, v)$ for all $u, v \in V$. If $f1$ and $f2$ are flows in G, which of the three flow properties must the flow $f1 + f2$ satisfy, and which might it violate?
- The edge connectivity of an undirected graph is the minimum number k of edges that muct be removed to disconnect the graph. For example, the edge connectivity of a tree is 1, and the edge connectivity of a cyclic chain of vertices is 2. Show that how the edge connectivity of an undirected graph $G = (V,E)$ can be determined by running a maximum-flow algorithm on at most $|V|$ flow networks, each having $O(V)$ vertices and $O(E)$ edges.

Bipartite Matching

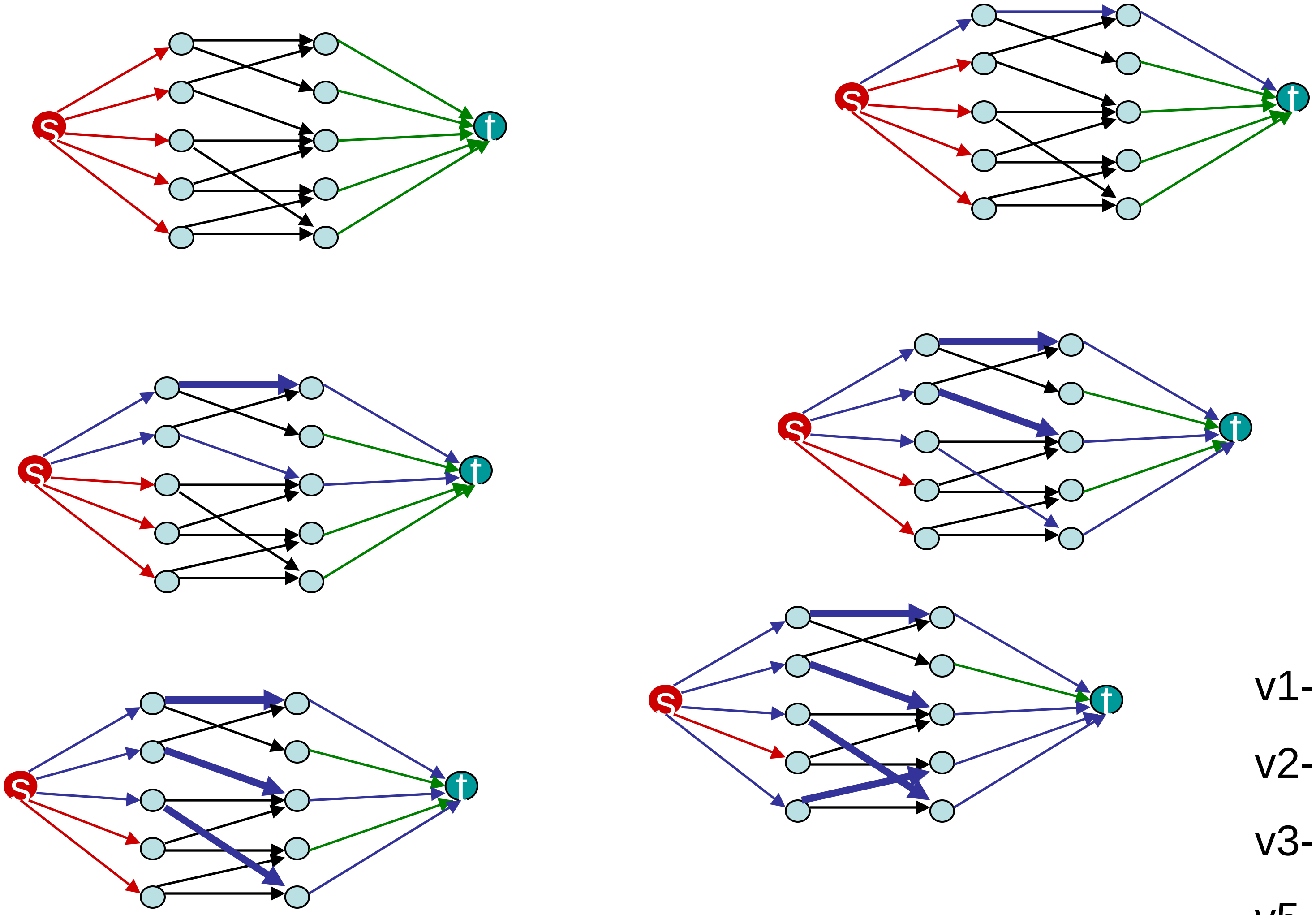
- Finding a matching M in G of largest size
- A bipartite graph $G = (V,E)$ is an undirected graph whose node set is partitioned into two sets X and Y such that $V = X \cup Y$. Every edge $e \in E$ has one end in X and the other end in Y.
- A matching M in G is a subset of the edges $M \subseteq E$ such that each node $v \in V$ appears in at most one edge in M.

Bipartite graph and Flow Network





Each edge has a capacity of ONE



v1-u1

v2-u3

v3-u5

v5-u4