Learn DAA : From B K Sharma

TCS-503: Design and Analysis of Algorithms

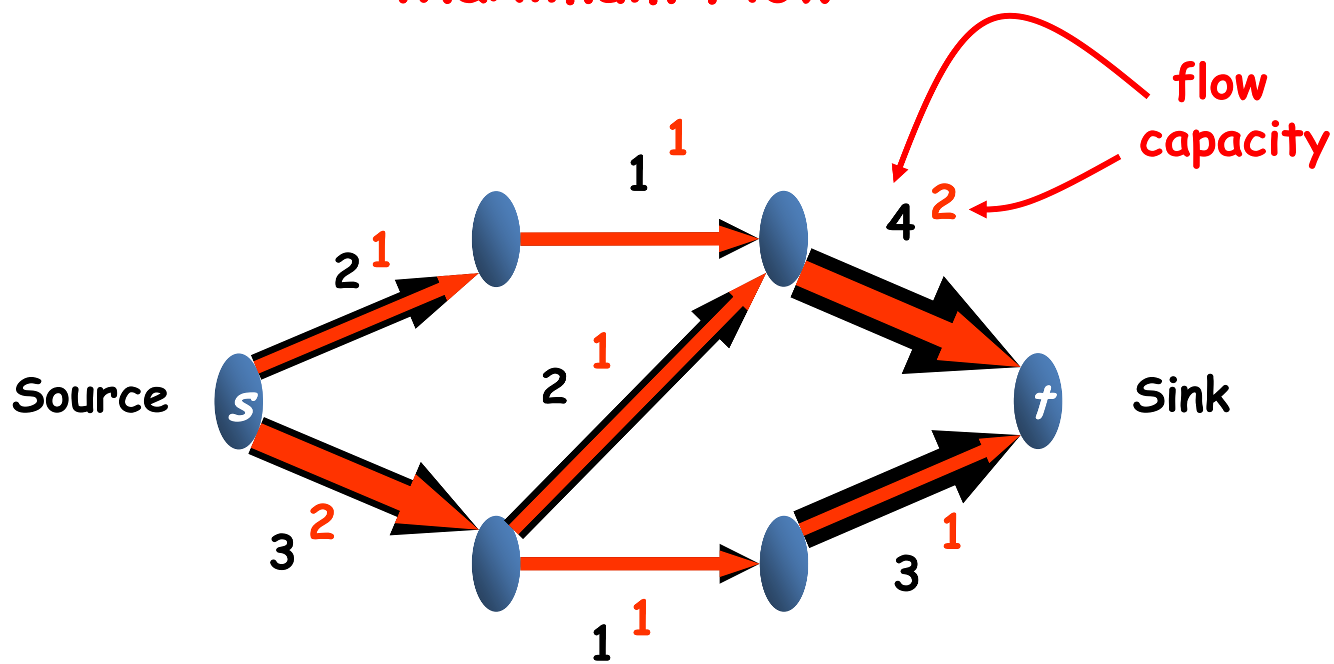
Graph Algorithms
Maximum Flow

Unit IV

- Graph Algorithms:
  - Elementary Graphs algorithms: BFS and DFS
  - Minimum Spanning Trees
  - Single-Source Shortest Paths
  - All-Pairs Shortest Paths
  - Maximum Flow and
  - Traveling Salesman Problem

Learn DAA : From B K Sharma

Maximum Flow



Maximum Flow Problem

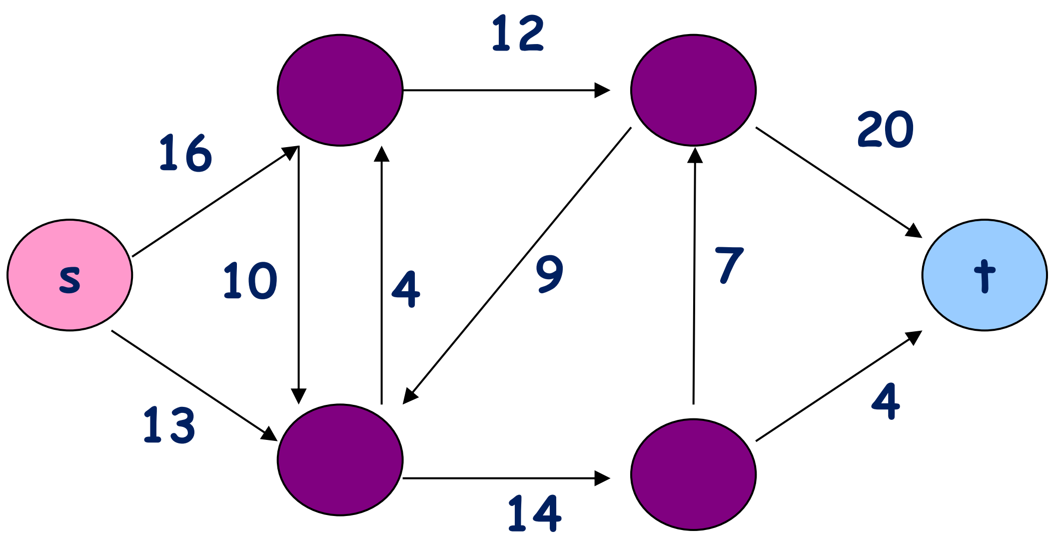
We are given a flow network G with source s and sink t, and we wish to find a flow of maximum value.

Learn DAA : From B K Sharma

Maximum Flow

Flow Network G

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



Learn DAA : From B K Sharma

Maximum Flow

Flow f(u,v)

For each edge (u,v), the **flow** f(u,v) is a real-valued function $f: V*V \rightarrow R$ that must satisfy 3 conditions:

Capacity constraint:

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

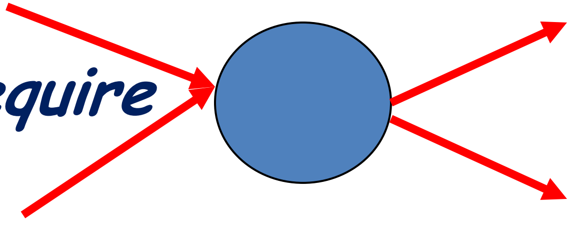
Skew symmetry:

For all $u, v \in V$, we require $f(u, v) = -f(v, u)$.

Flow conservation:

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

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



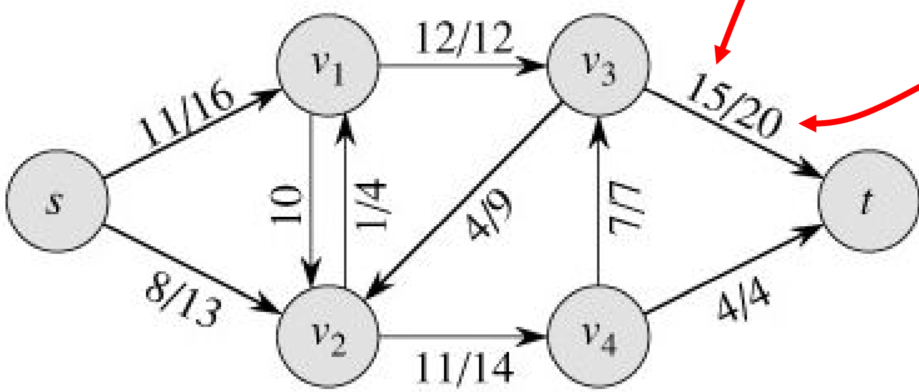
Learn DAA : From B K Sharma

Maximum Flow

Flow f(u,v)

flow

capacity



$f(v_2, v_1) = 1,\ c(v_2, v_1) = 4.$

$f(v_1, v_2) = -1,\ c(v_1, v_2) = 10.$

$f(v_3, s) + f(v_3, v_1) + f(v_3, v_2) + f(v_3, v_4) + f(v_3, t)$

$= 0 \quad + \quad (-12) \quad + \quad 4 \quad + \quad (-7) \quad + \quad 15$

$=0$

Maximum Flow

The Value of a Flow |f|

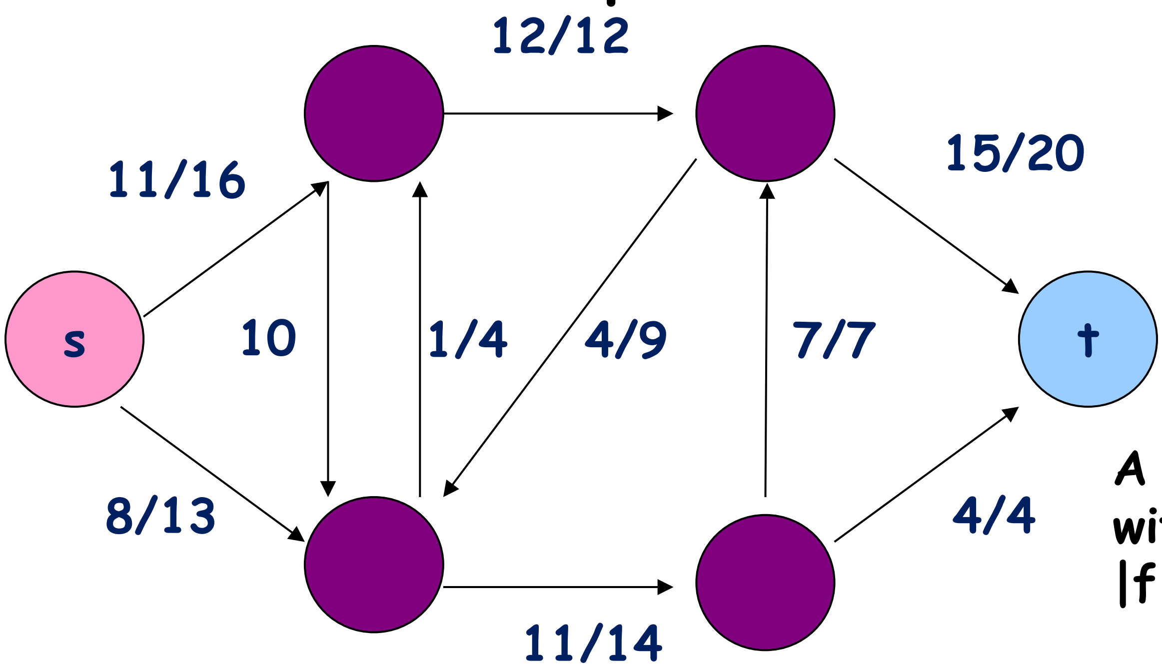
$$|f| = \sum_{v \in V} f(s,v) = \sum_{v \in V} f(v,t)$$

That is the total flow leaving s = the total flow arriving in t.

Maximum Flow

The Value of a Flow |f|

Example



A flow f in G with value |f|=19

$|f| = f(s, v_1) + f(s, v_2) + f(s, v_3) + f(s, v_4) + f(s, t)$

$= 11 + 8 + 0 + 0 + 0 = 19$

$|f| = f(s, t) + f(v_1, t) + f(v_2, t) + f(v_3, t) + f(v_4, t)$

$= 0 + 0 + 0 + 15 + 4 = 19$

Learn DAA : From B K Sharma

Maximum Flow

The Residual Capacity $c_f(u,v)$

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*

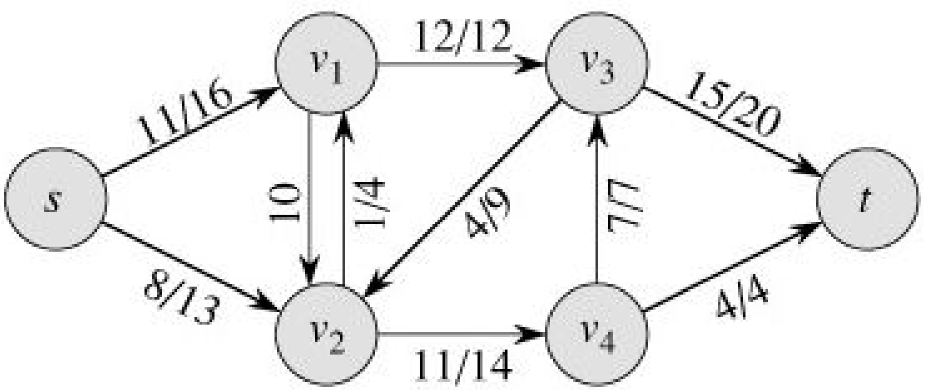
$$c_f(u,v) = c(u,v) - f(u,v)$$

Learn DAA : From B K Sharma

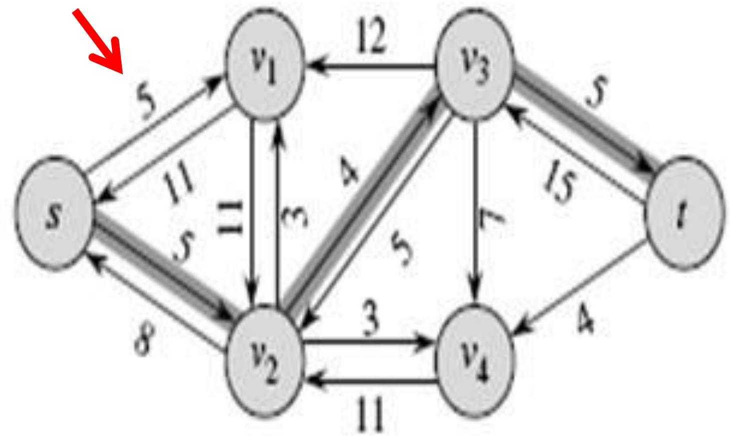
Maximum Flow

The Residual Capacity $c_f(u,v)$

Example



Residual Capacity



For example, if *c(u, v) = 16 and f (u, v) = 11, then we can increase f (u, v)* by *c f (u, v) = 5 units before we exceed the capacity constraint on edge (u, v).*

Maximum Flow

The Residual Network

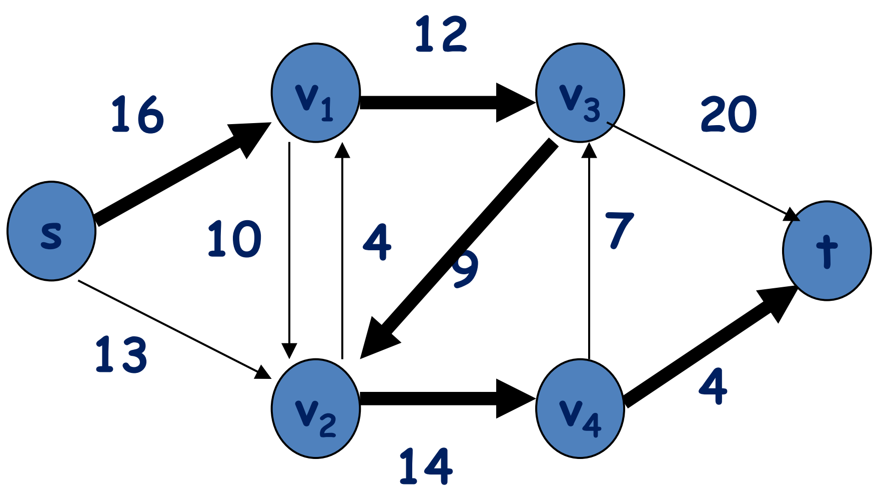
Given a flow network and a flow, the residual network consists of edges that can admit more net flow.

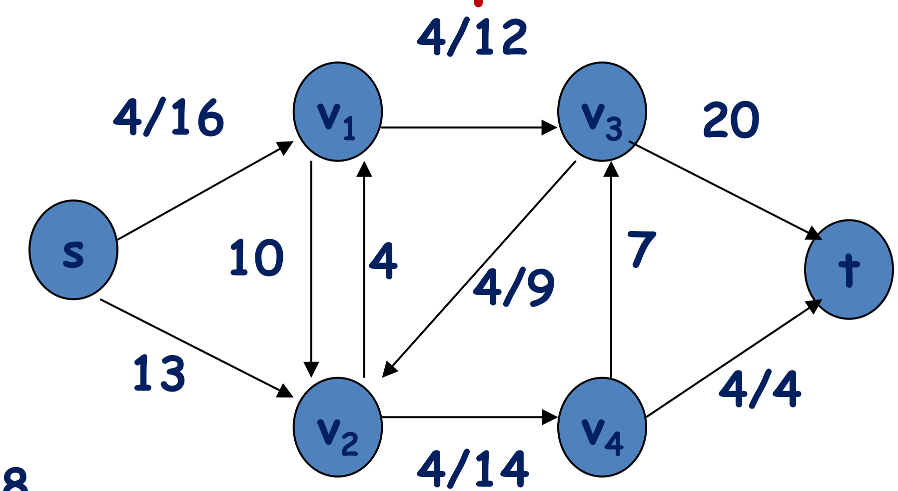
The edges of the residual network are the edges on which the residual capacity is positive.

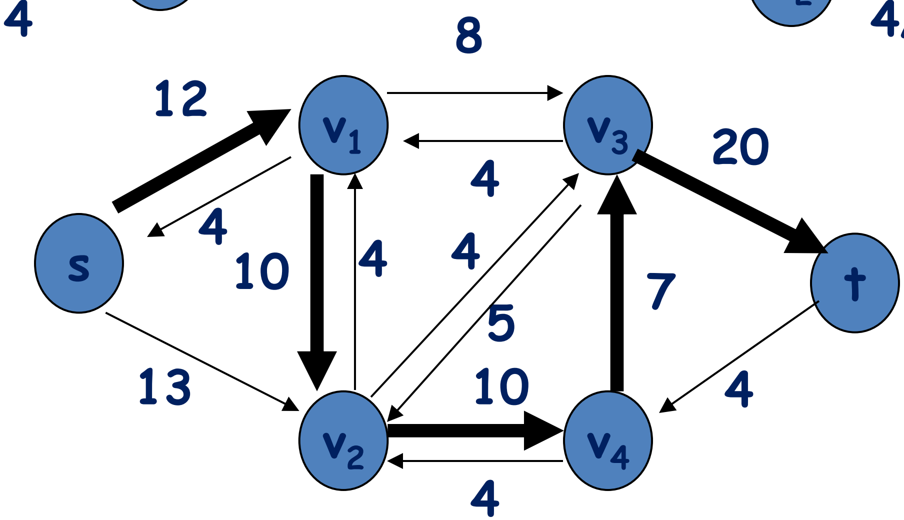
Learn DAA : From B K Sharma

Maximum Flow

The Residual Network: Example



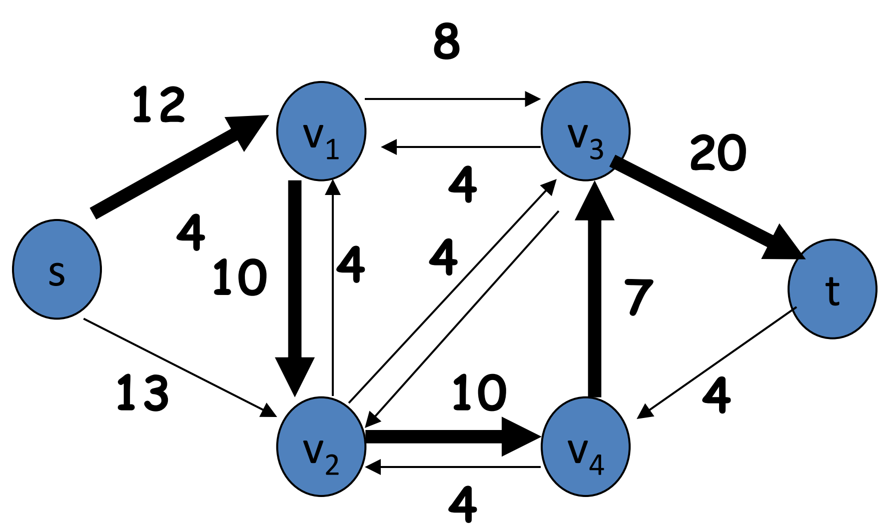




Maximum Flow

Augmenting Path p

Given a flow network $G=(V,E)$ and a flow f, an augmenting path is a simple path from s to t in the residual network G_f.



Bold Edges (s-v_1-v_2-v_4-v_3-t)

Maximum Flow Algorithms

- Ford-Fulkerson Algorithm
- Edmonds-Karp Algorithm

Maximum Flow Algorithms

Ford-Fulkerson Algorithm

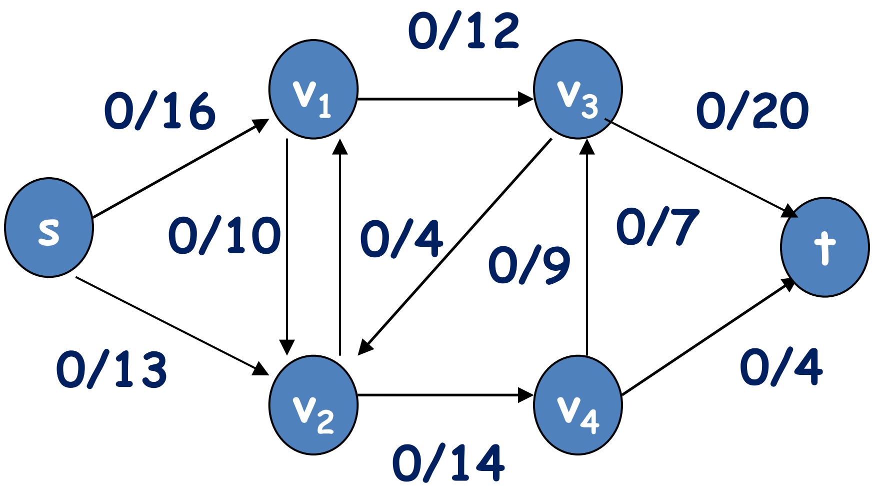
Alg.: FORD-FULKERSON(G,s,t)

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

Learn DAA : From B K Sharma

Maximum Flow Algorithms

Ford-Fulkerson Algorithm: Example

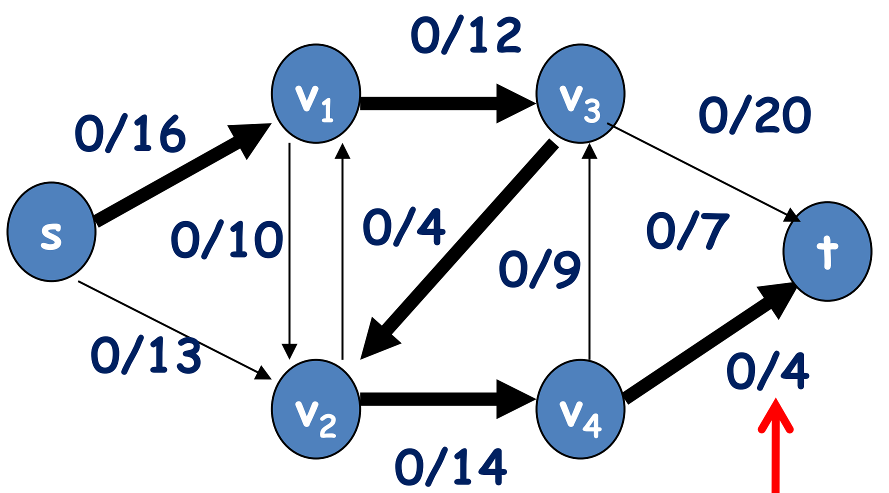


(a)

Learn DAA : From B K Sharma

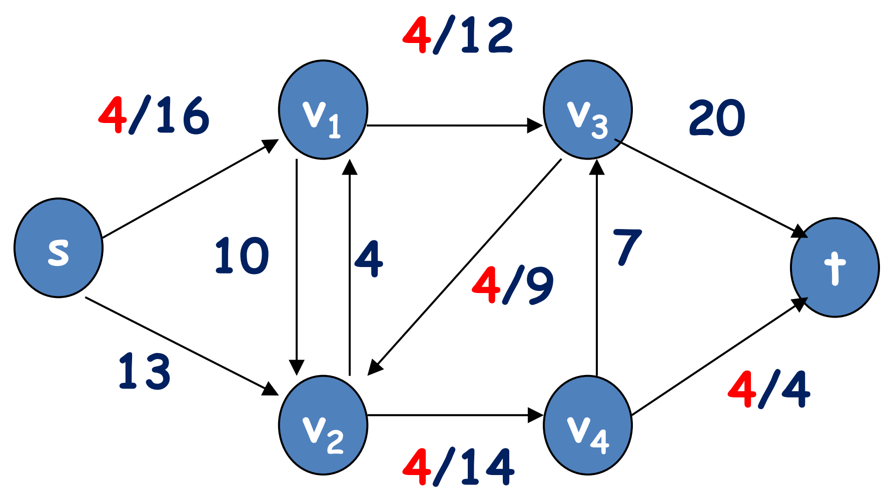
Maximum Flow Algorithms

Ford-Fulkerson Algorithm: Example



G_f with A.P p

$c_f(p)$



New flow f that results from adding f_p to f.

p: s-v_1-v_3-v_2-v_4-t (b)

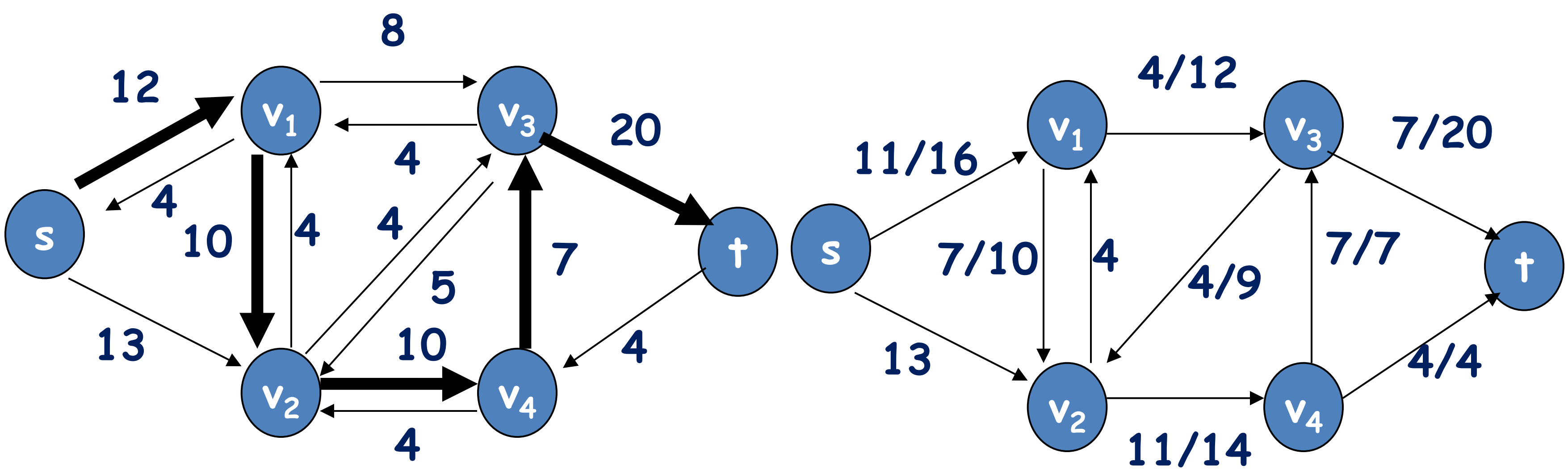
$c_f(p) \leftarrow \min\{c_f(u,v): (u,v) \text{ is in } p\}$

$c_f(p)=4$

Learn DAA : From B K Sharma

Maximum Flow Algorithms

Ford-Fulkerson Algorithm: Example



G_f with A.P p

p: $s-v_1-v_2-v_4-v_3-t$ (c)

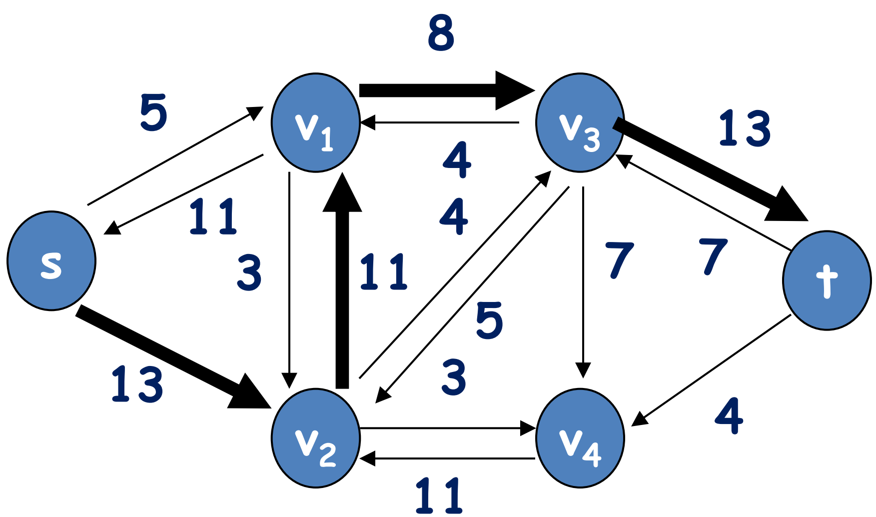
New flow f that results from adding f_p to f.

$c_f(p) \leftarrow \min\{c_f(u,v): (u,v) \text{ is in } p\}$

$c_f(p)=7$

Maximum Flow Algorithms

Ford-Fulkerson Algorithm: Example



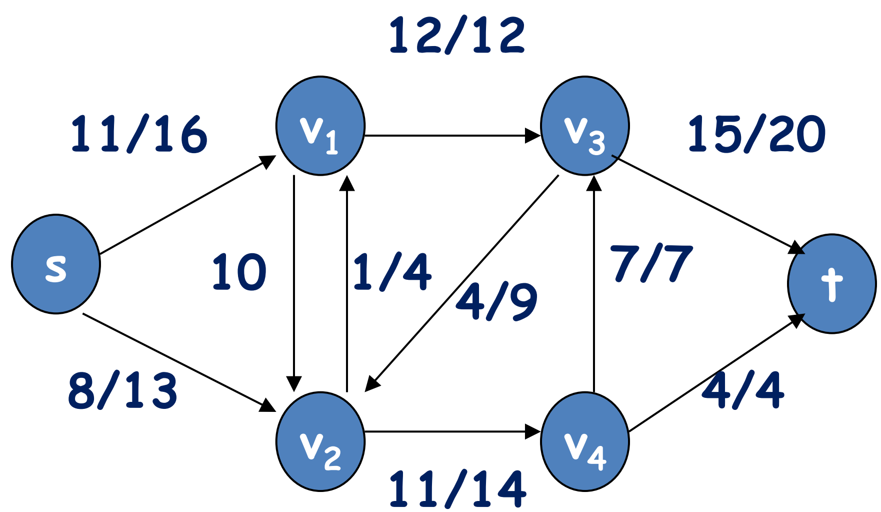
G_f with A.P p

p: $s-v_2-v_1-v_3-t$

(d)

$c_f(p) \leftarrow \min\{c_f(u,v): (u,v) \text{ is in } p\}$

$c_f(p)=7$

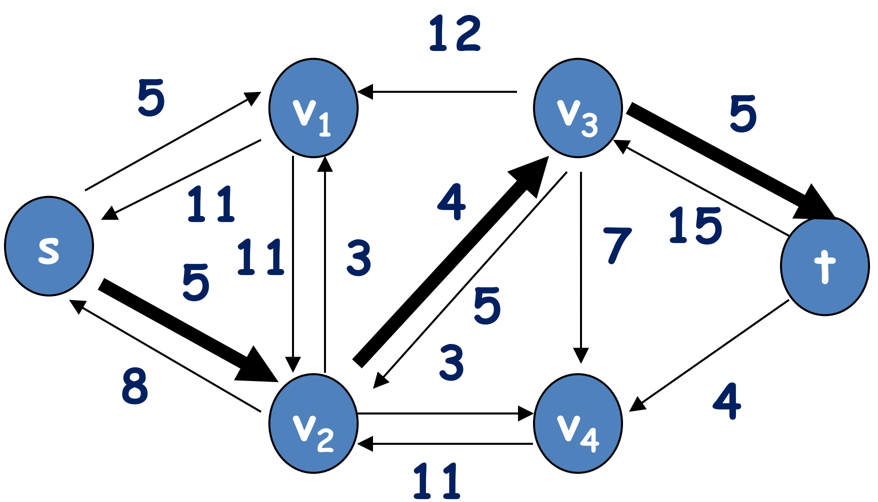


New flow f that results from adding f_p to f.

Learn DAA : From B K Sharma

Maximum Flow Algorithms

Ford-Fulkerson Algorithm: Example



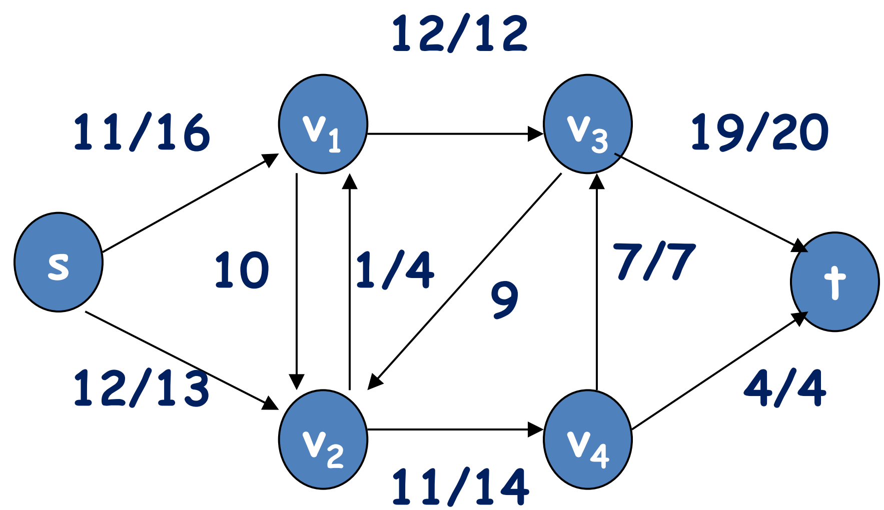
G_f with A.P p

p: $s\text{-}v_2\text{-}v_3\text{-}t$

(e)

$c_f(p) \leftarrow \min\{c_f(u,v): (u,v) \text{ is in } p\}$

$c_f(p)=4$

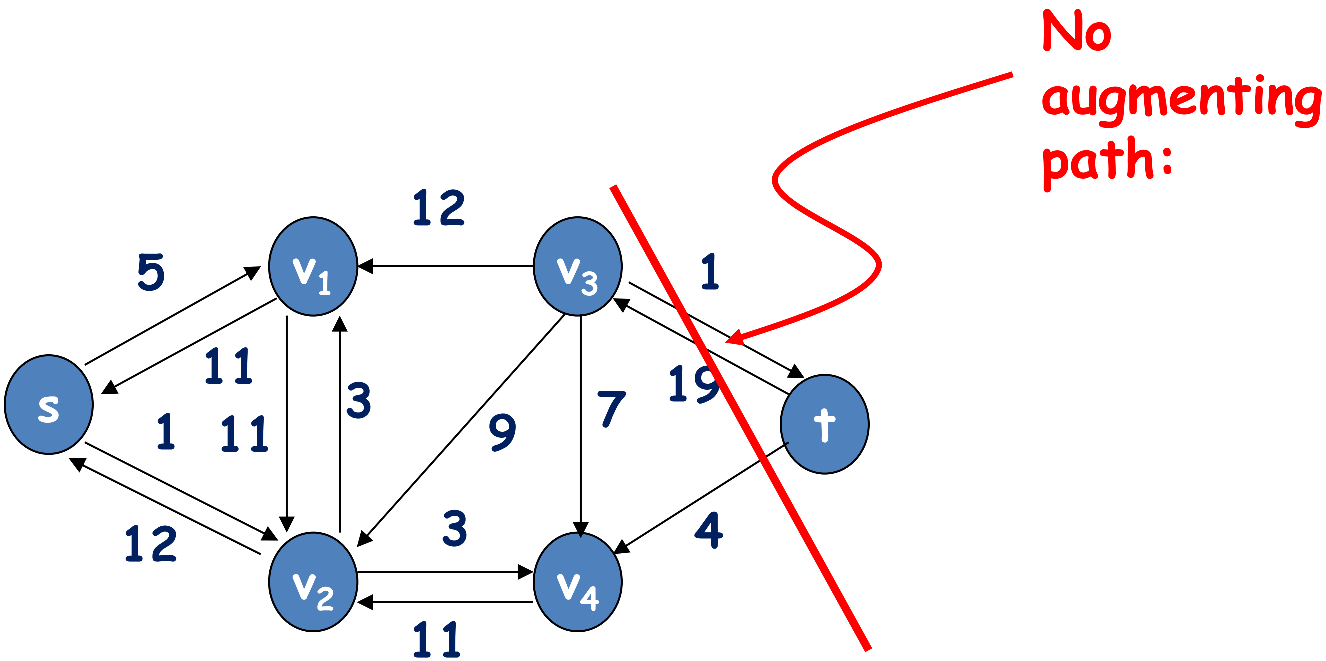


New flow f that results from adding f_p to f.

Learn DAA : From B K Sharma

Maximum Flow Algorithms

Ford-Fulkerson Algorithm: Example



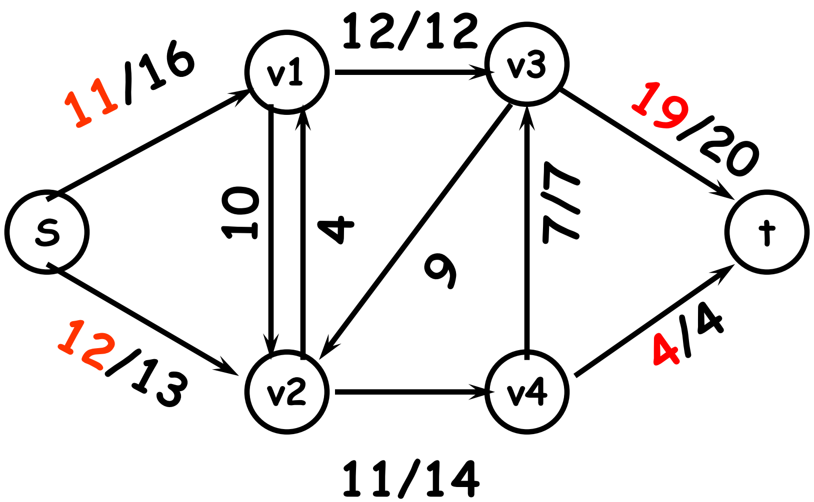
The G_f at the last while loop test.

(f)

Learn DAA : From B K Sharma

Maximum Flow Algorithms

Ford-Fulkerson Algorithm: Example



Finally we have:

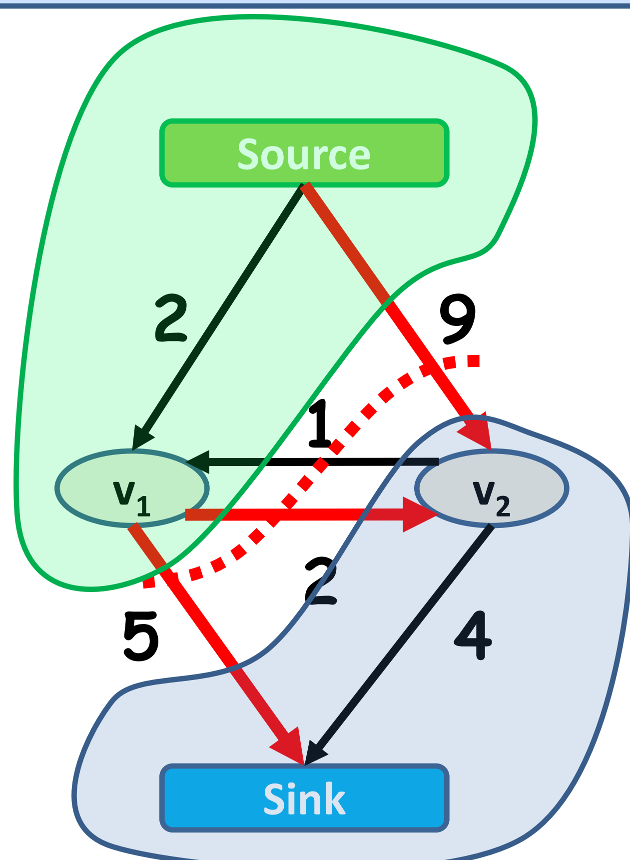
$| f | = f(s, V) = 23$

Learn DAA : From B K Sharma

The ST Min-Cut Problem

What is a ST-cut?

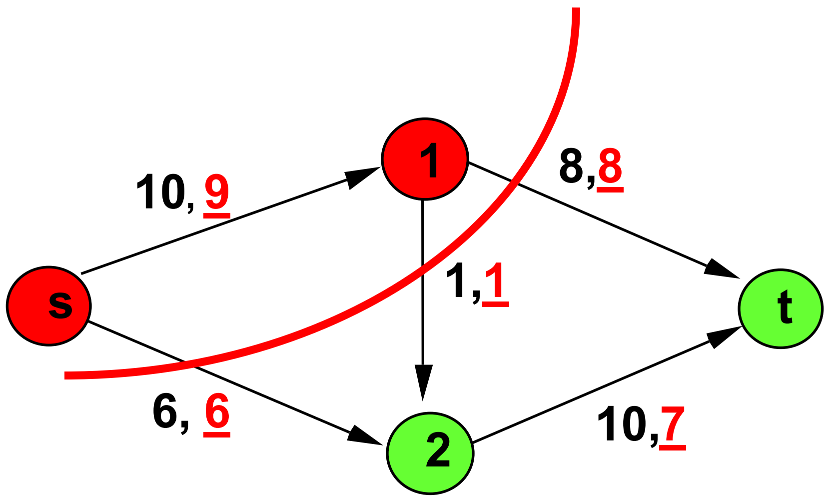
An st-cut (S,T) divides the nodes between source and sink.



Max-Flow Min-Cut Theorem

Cut

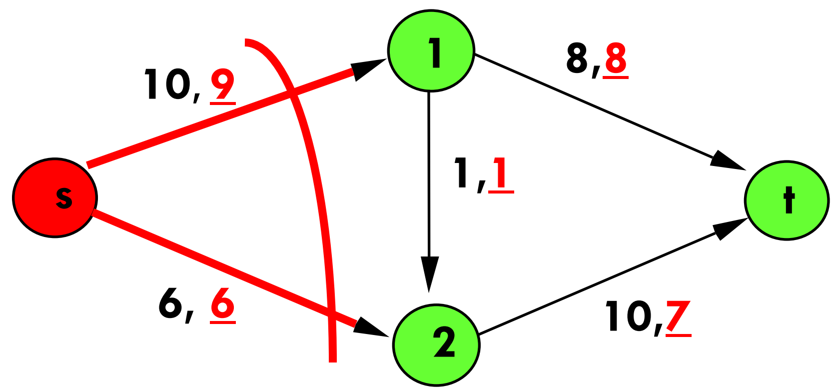
A cut *(S, T)* of flow network G=(V,E) is a partition of V into two disjoint subsets S and T=V-S such that $s \in S$ and $t \in T$.



S = { s, 1 } and T = { 2, t }

Max-Flow Min-Cut Theorem

Cut

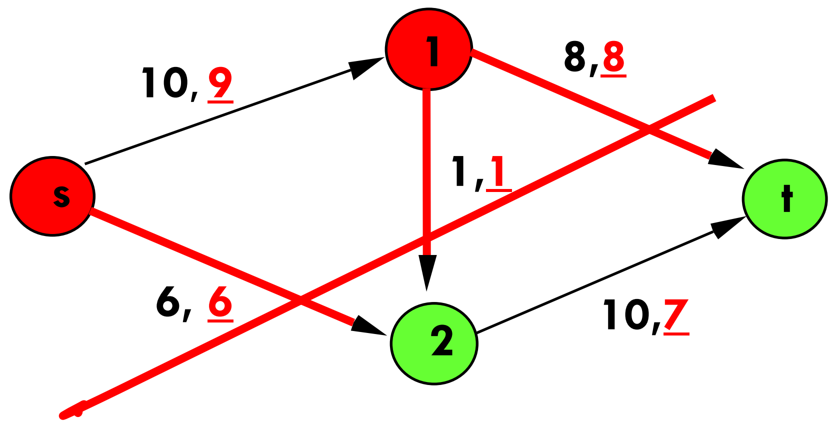


S={s} and T={1,2,t}

Learn DAA : From B K Sharma

Max-Flow Min-Cut Theorem

Cut

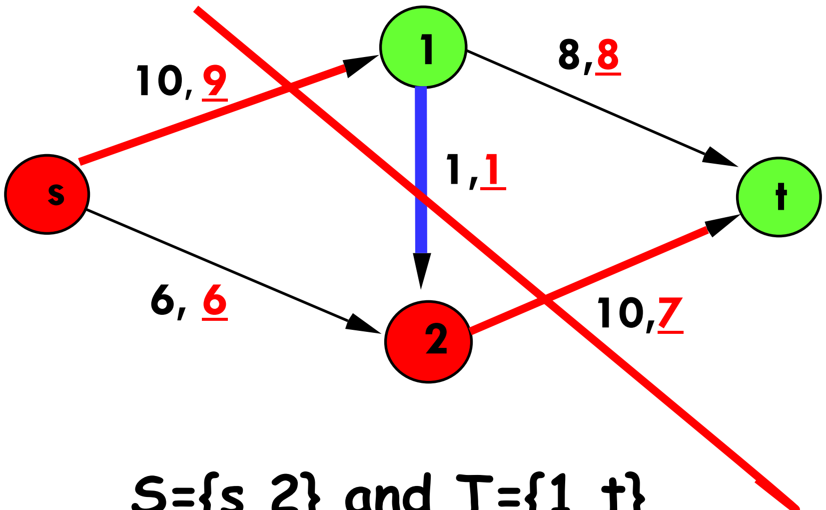


S={s, 1} and T={2, t}

Learn DAA : From B K Sharma

Max-Flow Min-Cut Theorem

Cut

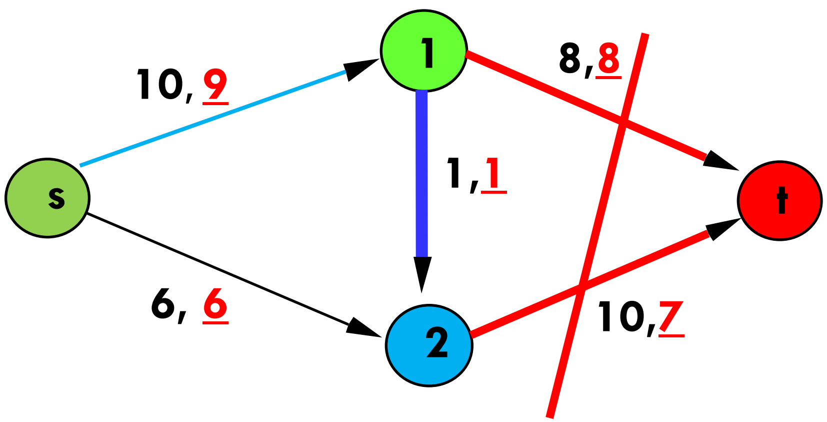


S={s,2} and T={1,t}

Learn DAA : From B K Sharma

Max-Flow Min-Cut Theorem

Cut



S={t} and T={s,1,2}

s $\notin$ S & t $\notin$ T

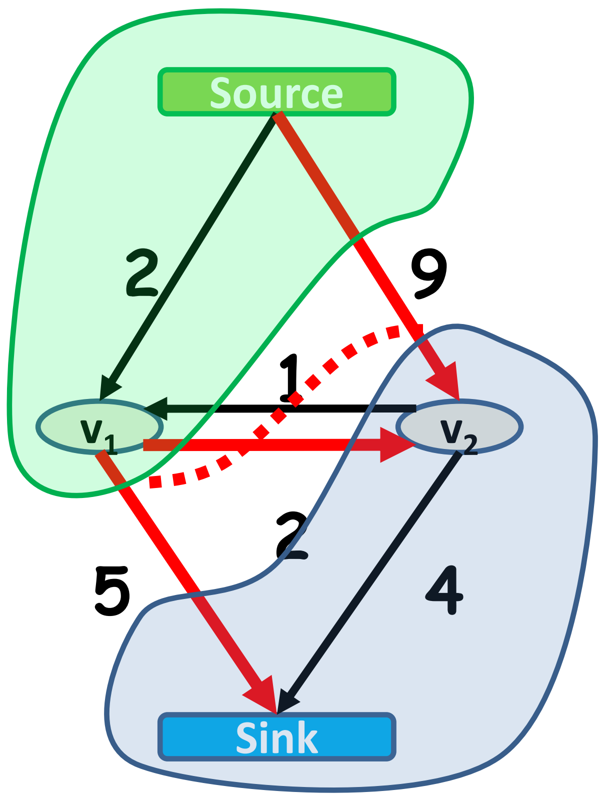
Learn DAA : From B K Sharma

The ST Min-Cut Problem

What is the cost/capacity of a ST-Cut?

Sum of cost of all edges going from S to T

5 + 2 + 9 = 16



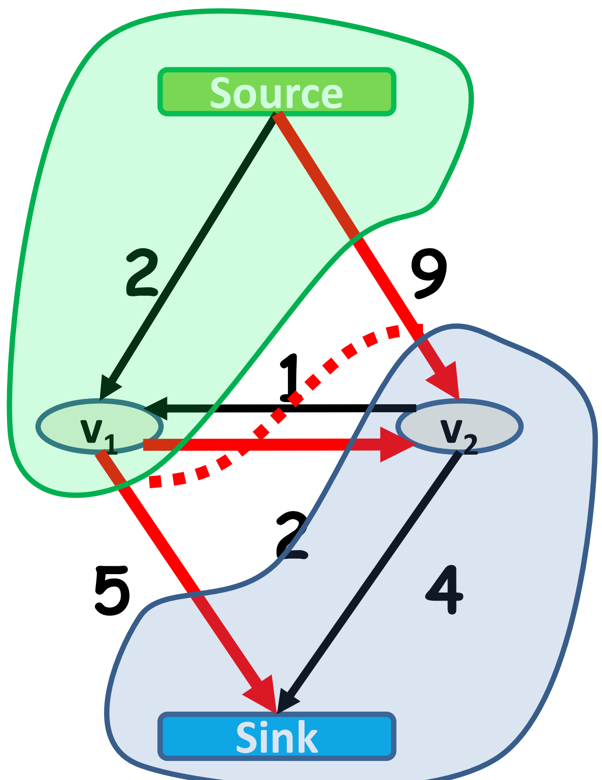
Learn DAA : From B K Sharma

The ST Min-Cut Problem

What is the ST-Min-Cut?

ST-Cut with the minimum cost

2 + 1 + 4 = 7



The ST Min-Cut Problem

Min Cut

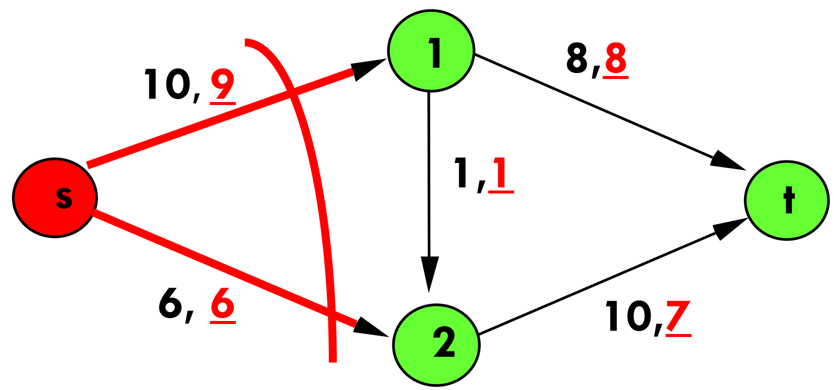
Min-cut: a "cut" on the graph crossing the fewest number of edges separating the source-set S and the sink-set T.

The edges S->T in this set should have a tail in S and a head in T.

c(S,T): The capacity of the minimum cut is the sum of all the outbound edges in the cut.

The ST Min-Cut Problem

Flow across Cut and Capacity of Cut



$S=\{s\}$ and $T=\{1,2,t\}$

$f(S,T)=9 + 6 = 15$

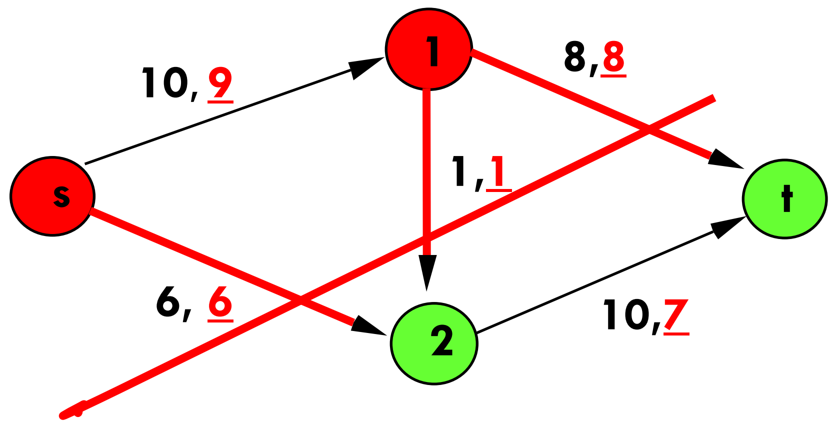
$c(S,T)=10 + 6=16$

$c(S,T)$: Sum of weights of edges on leaving S.

Learn DAA : From B K Sharma

The ST Min-Cut Problem

Flow across Cut and Capacity of Cut



S={s, 1} and T={2, t}

f(S,T)=8+1+6 = 15

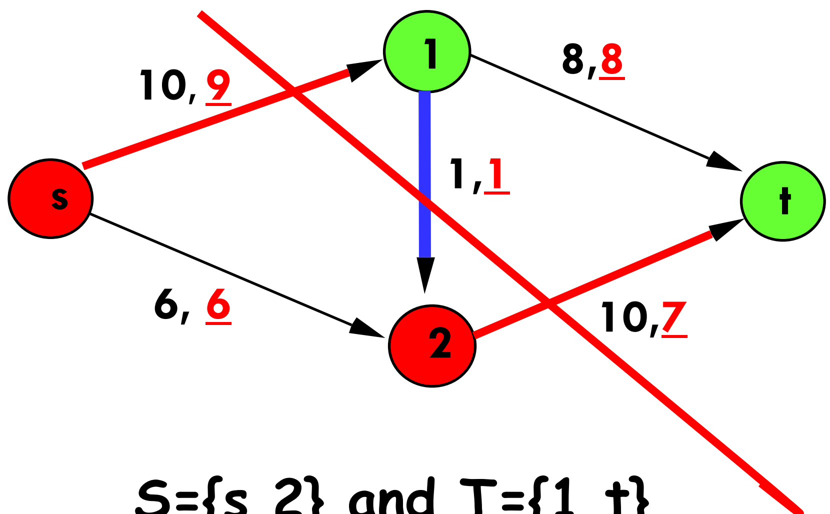
c(S,T)=8+1+6=15

c(S,T): Sum of weights of edges on leaving S.

Learn DAA : From B K Sharma

The ST Min-Cut Problem

Flow across Cut and Capacity of Cut



S={s,2} and T={1,t}

f(S,T)=9 + 7 +(-1) = 15

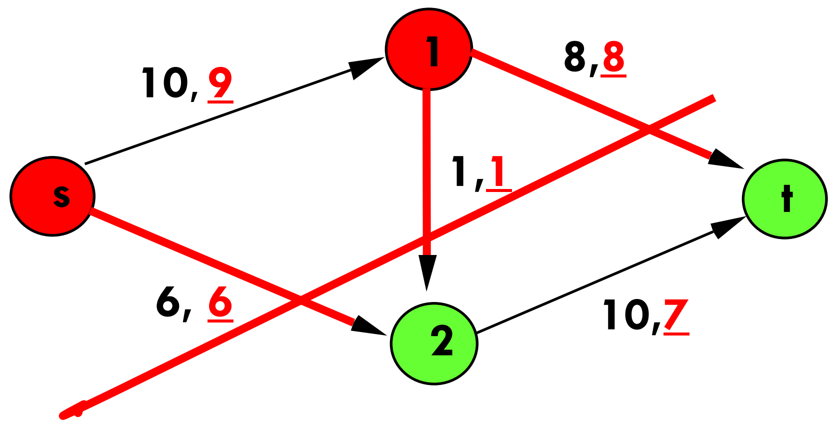
c(S,T)=10+10=20

c(S,T): Sum of weights of edges on leaving S.

The ST Min-Cut Problem

The minimum cut of a network

is a cut whose capacity c[S,T] is minimum over all cuts of the network.



S={s,1} and T={2,t}

c(S,T)=8+1+6=15

c(S,T): Sum of weights of edges on leaving S.

How to Compute ST Min-Cut?

Min-Cut / Max-Flow Theorem

In every network, the maximum flow equals the cost/capacity of the ST-Min-Cut

Learn DAA : From B K Sharma

Max-Flow Min-Cut Theorem

The value of a maximum flow is equal to the capacity of a minimum cut.

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

↓

Maximum Flow Algorithms

Max-Flow Min-Cut Theorem

f is maximum flow in G.

Equivalent to

A flow is maximum if and only if its residual network contains no augmenting paths.

Equivalent to

The value of a maximum flow is equal to the capacity of a minimum cut.

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

Learn DAA : From B K Sharma

Maximum Flow Algorithms

Edmonds-Karp Algorithm

Variation of Ford-Fulkerson's algorithm

Chooses an augmenting path *p*

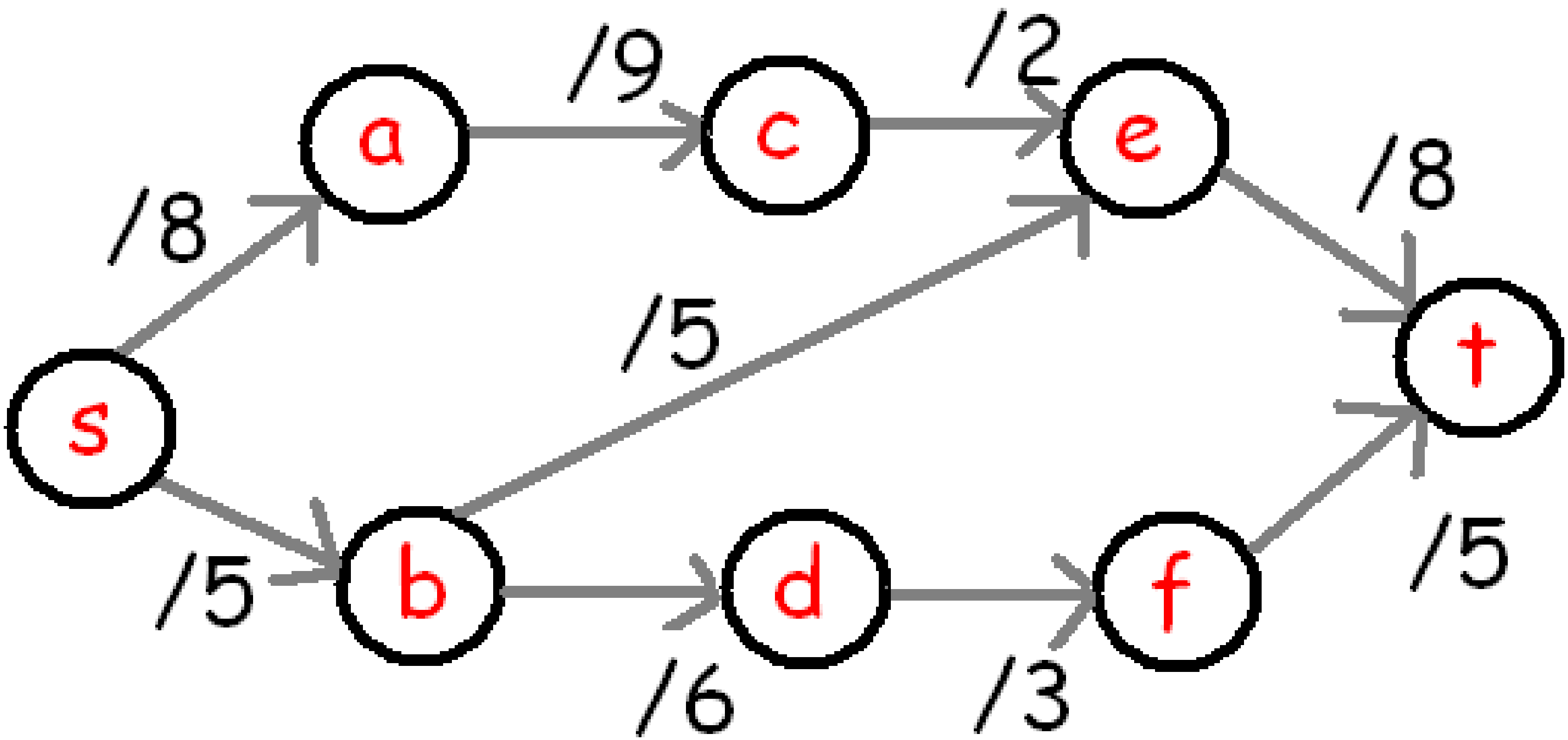
with the smallest number of edges

Uses BFS to find augmenting paths.

Learn DAA : From B K Sharma

Edmonds-Karp Algorithm

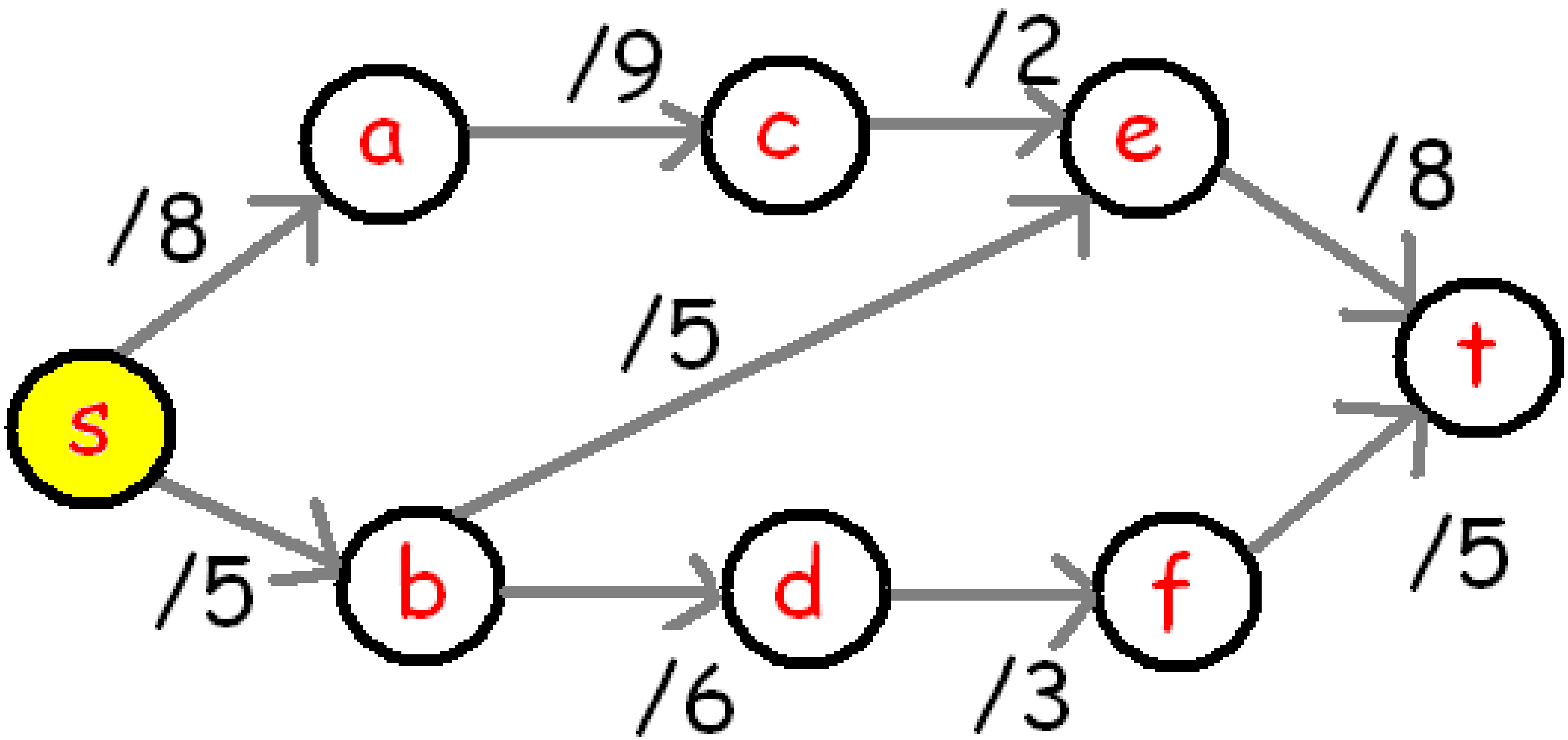
Example



Learn DAA : From B K Sharma

Edmonds-Karp Algorithm

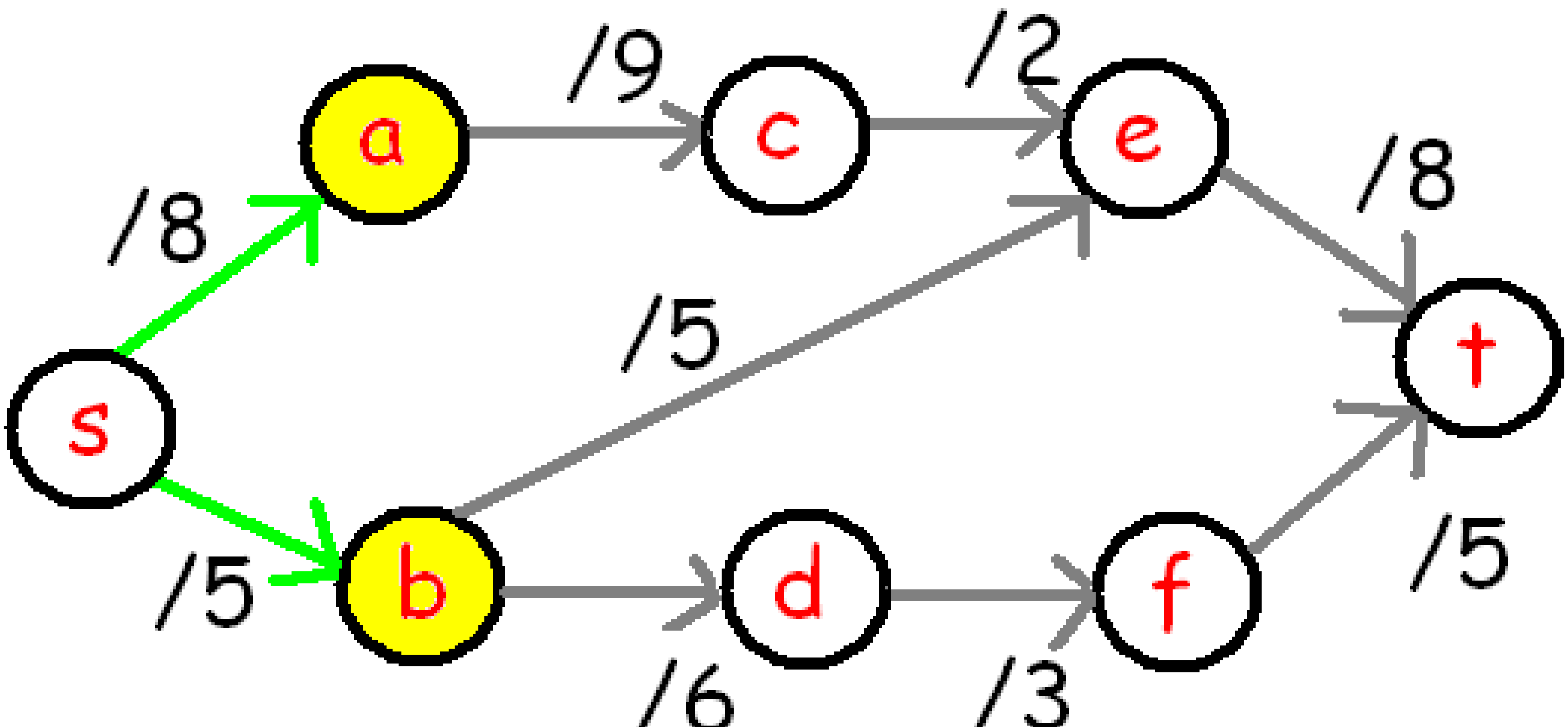
Example



Learn DAA : From B K Sharma

Edmonds-Karp Algorithm

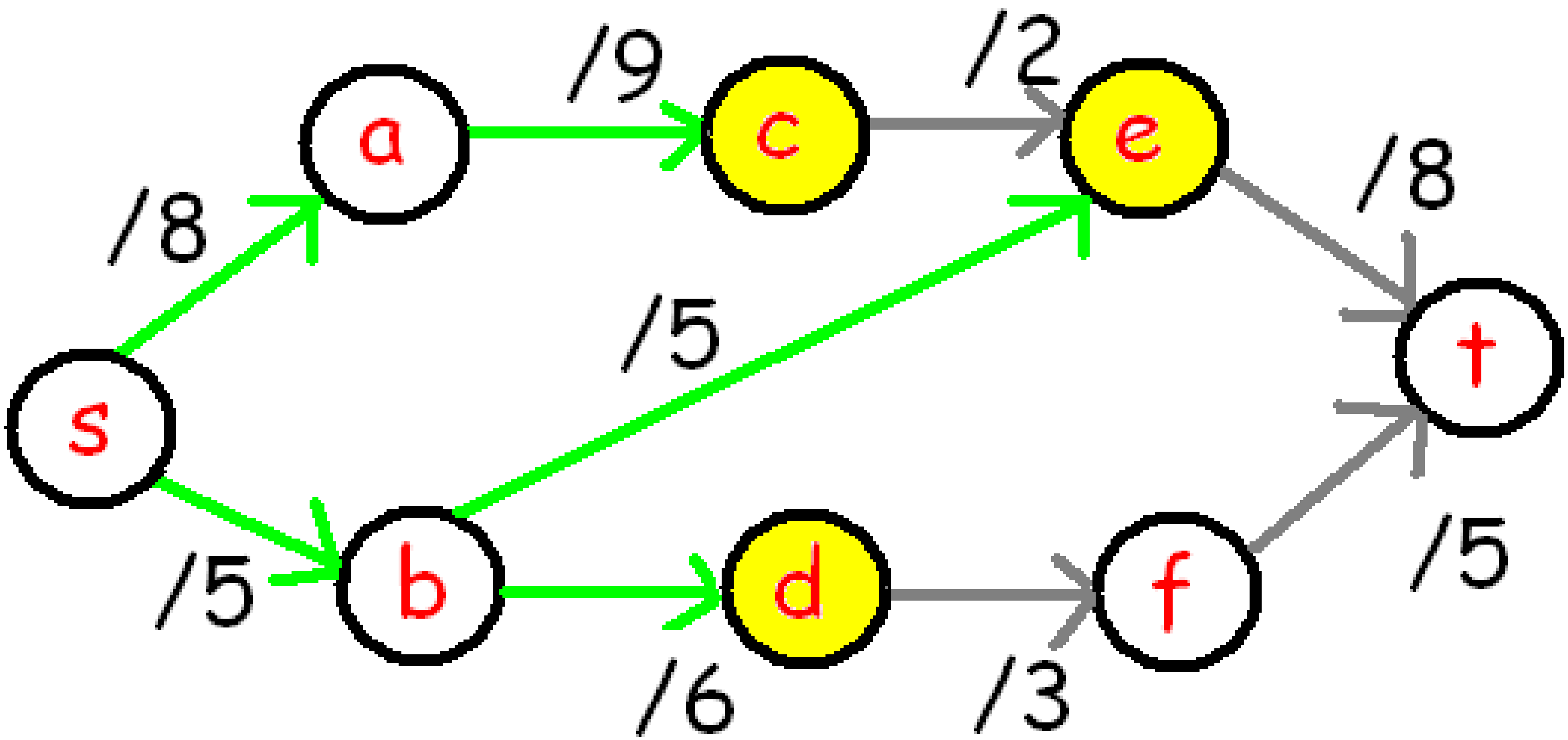
Example



Learn DAA : From B K Sharma

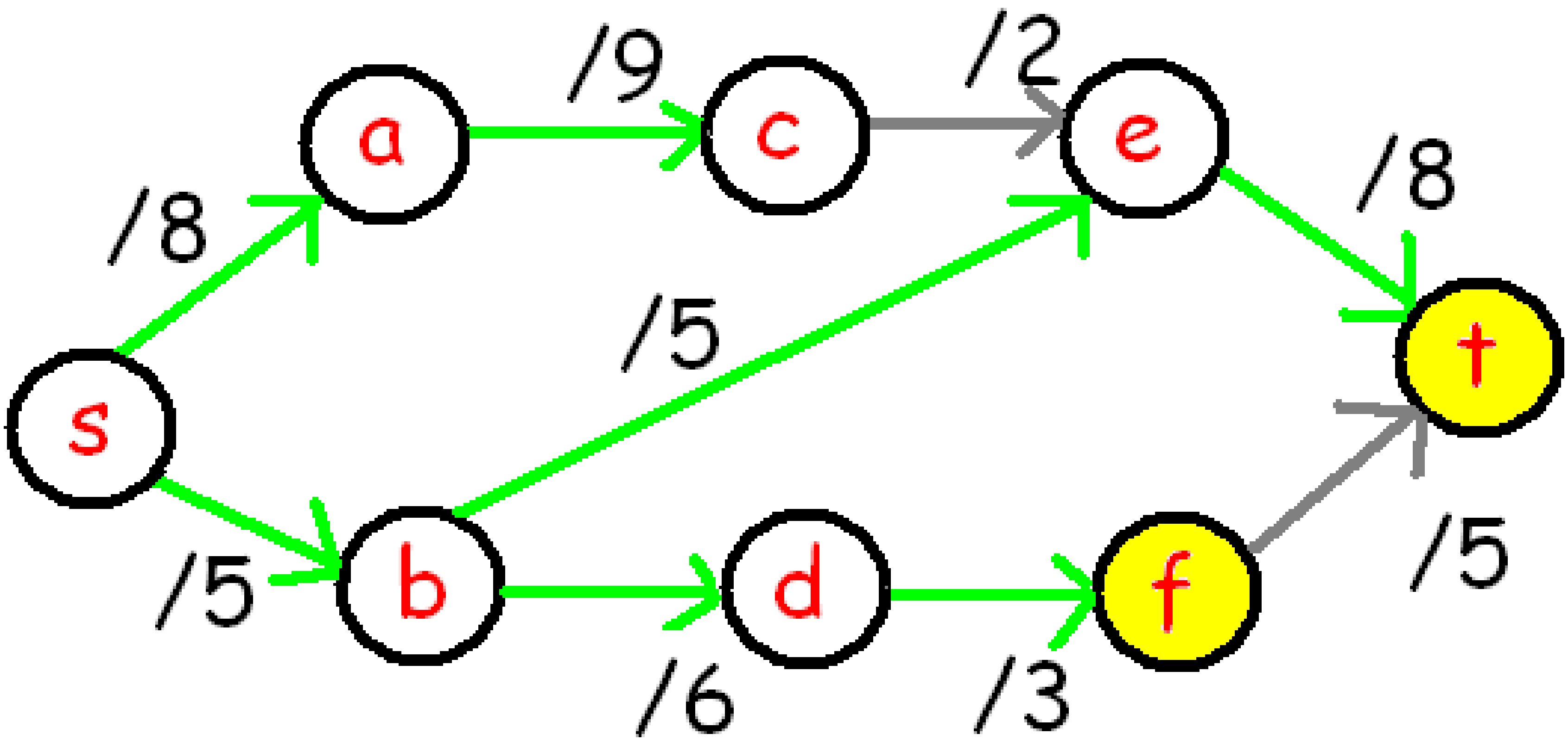
Edmonds-Karp Algorithm

Example



Edmonds-Karp Algorithm

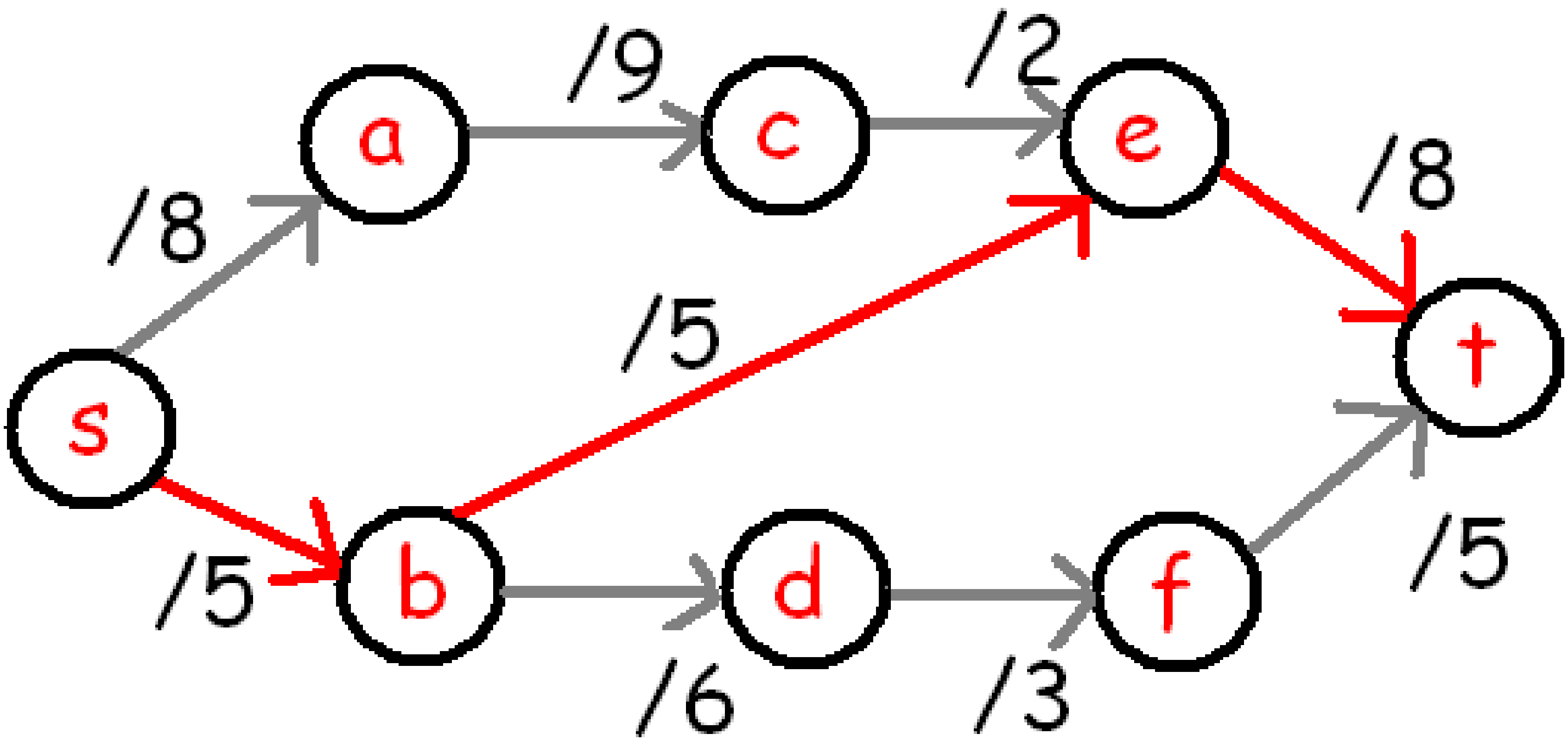
Example



Learn DAA : From B K Sharma

Edmonds-Karp Algorithm

Example

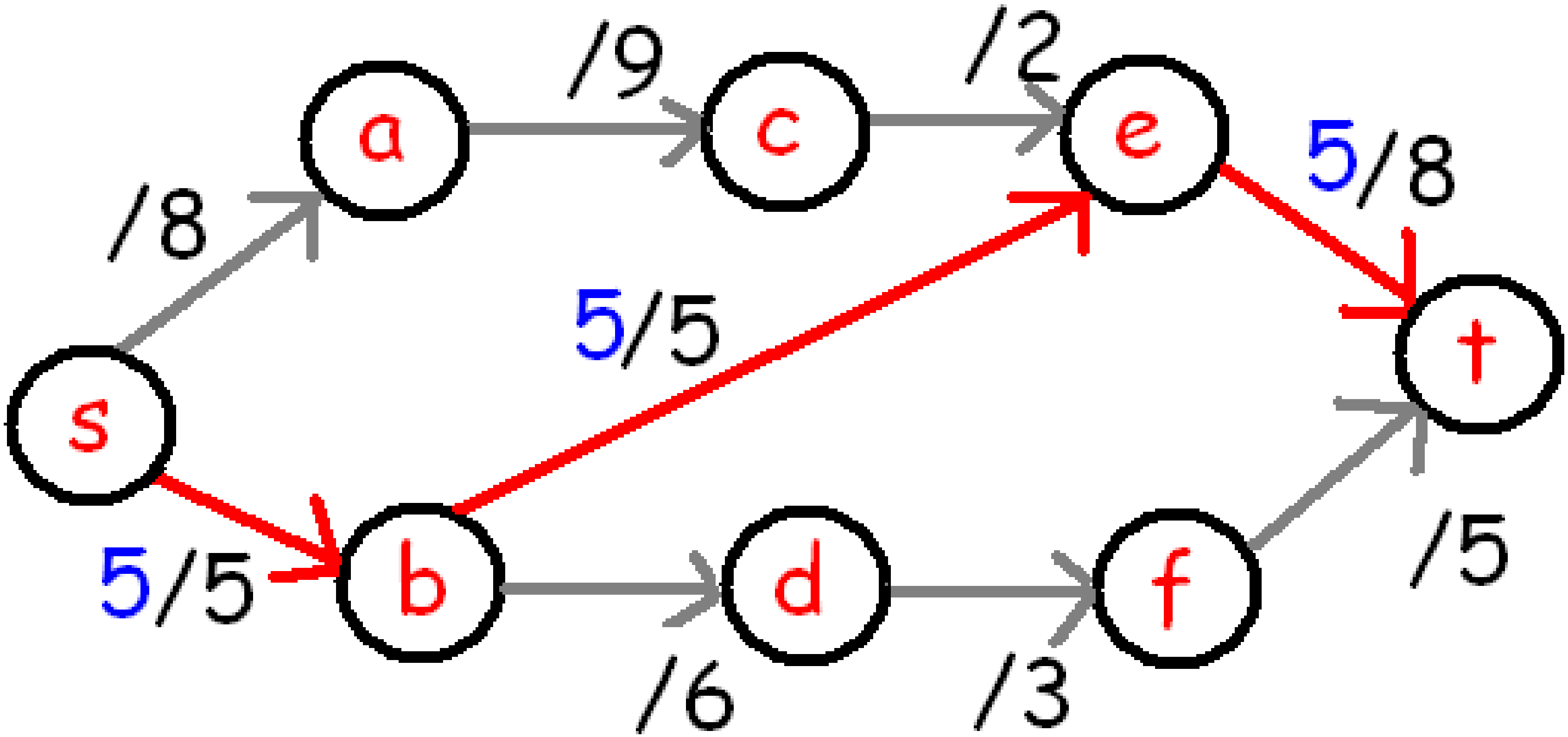


Augmenting path: s, b, e, t

Learn DAA : From B K Sharma

Edmonds-Karp Algorithm

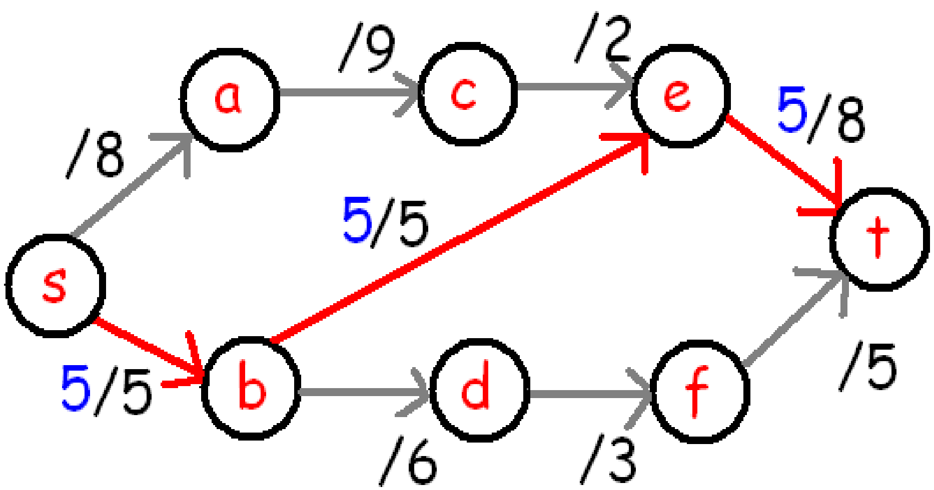
Example



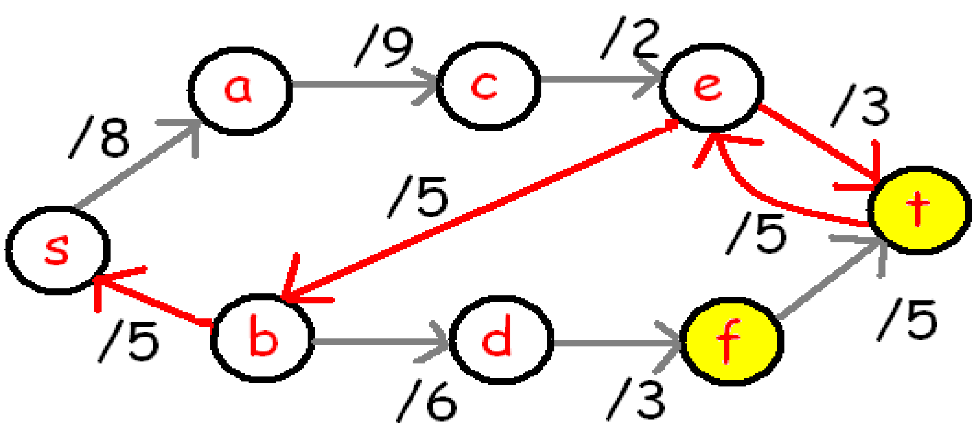
Send 5 units of flow along augmenting path.

Edmonds-Karp Algorithm

Example



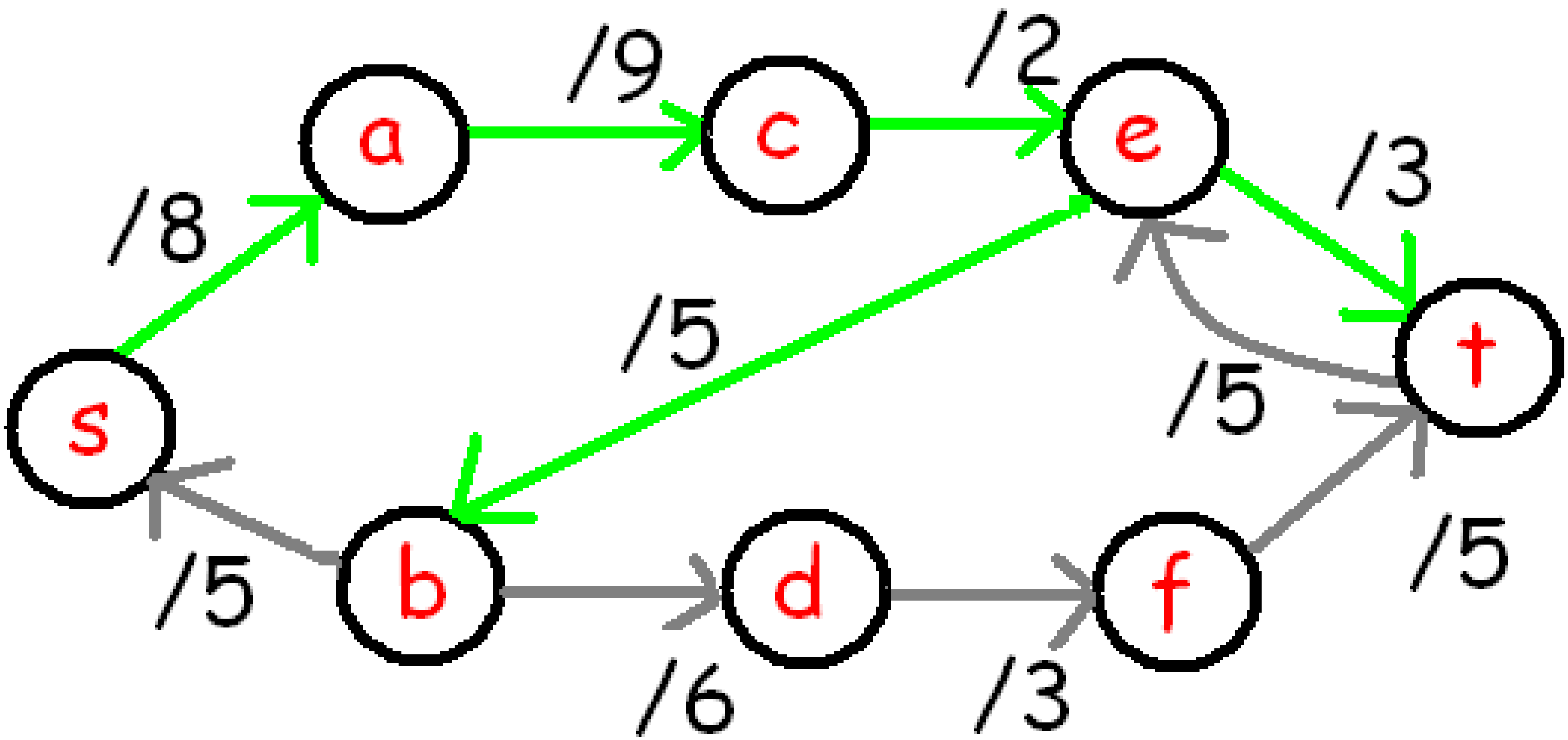
Create new auxiliary graph:



Learn DAA : From B K Sharma

Edmonds-Karp Algorithm

Example

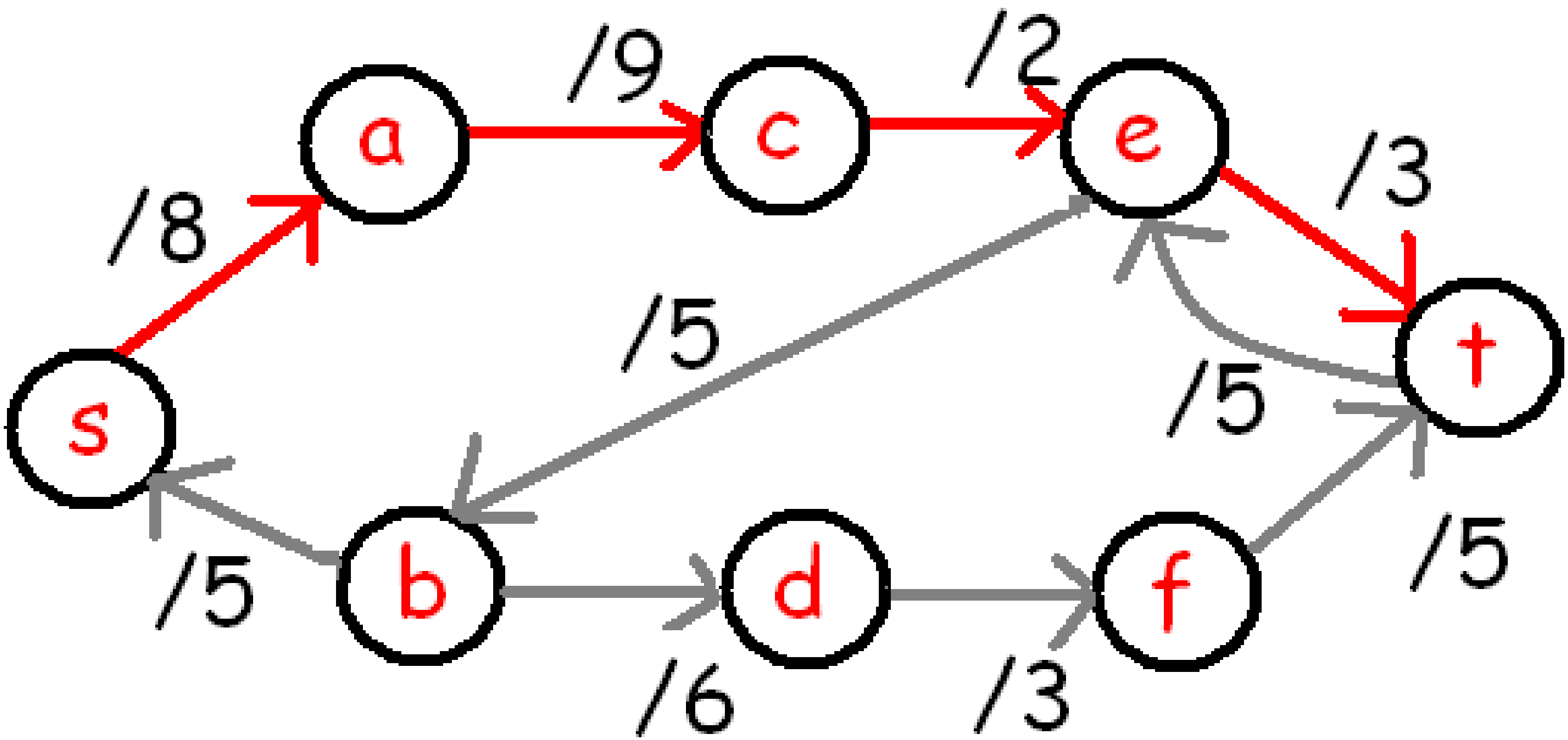


Do BFS starting at s of auxiliary graph.

Learn DAA : From B K Sharma

Edmonds-Karp Algorithm

Example

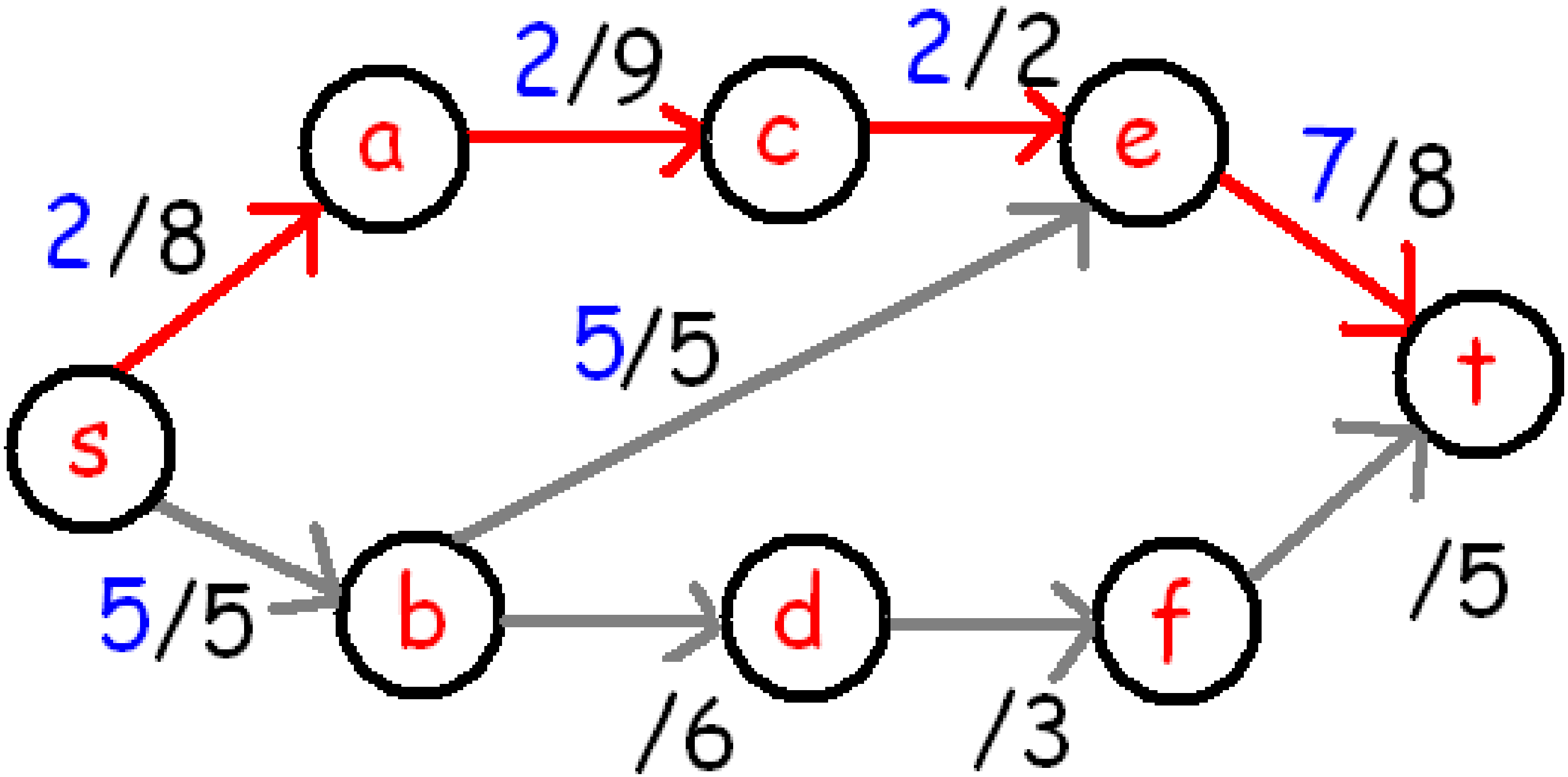


Send 2 units of flow along augmenting path.

Learn DAA : From B K Sharma

Edmonds-Karp Algorithm

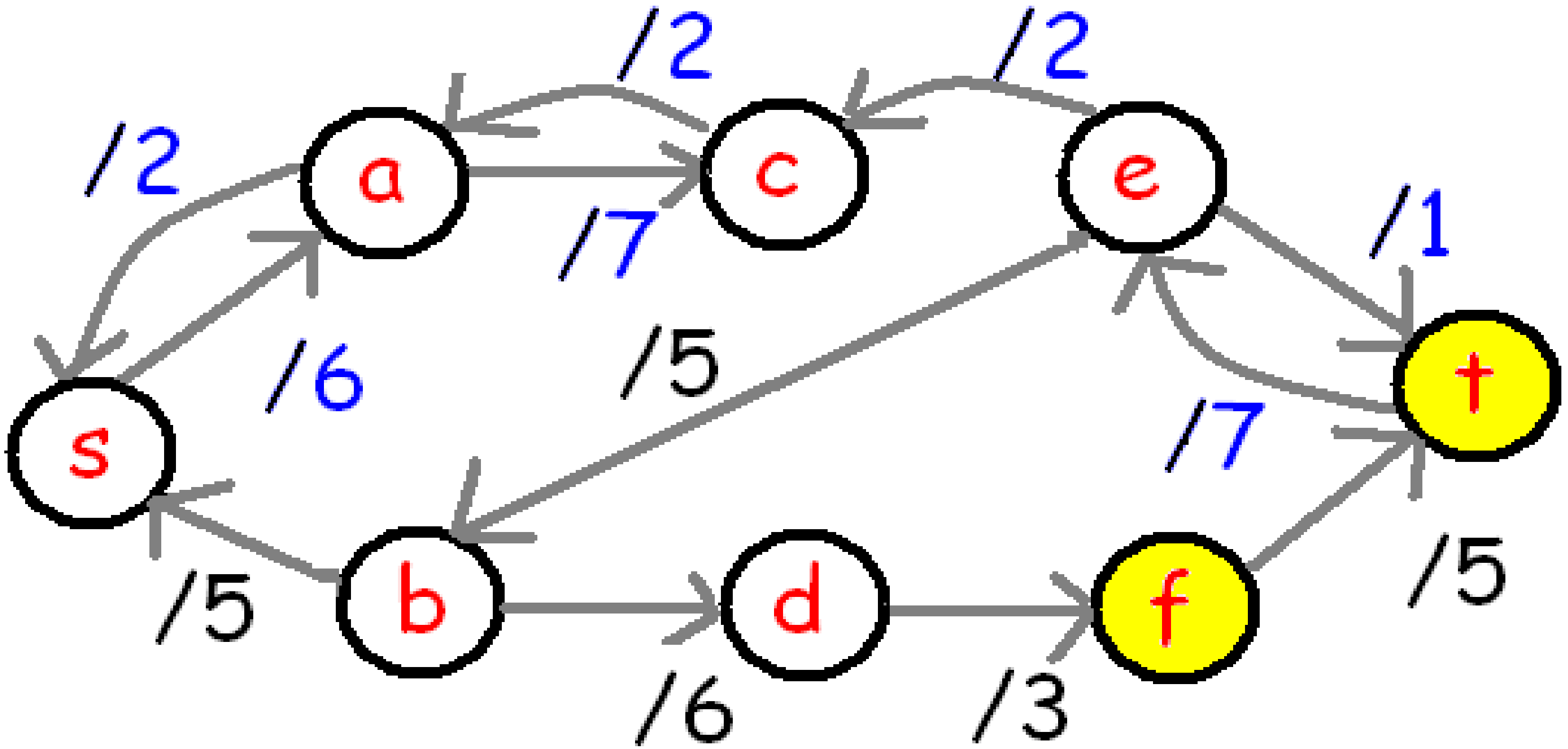
Example



Learn DAA : From B K Sharma

Edmonds-Karp Algorithm

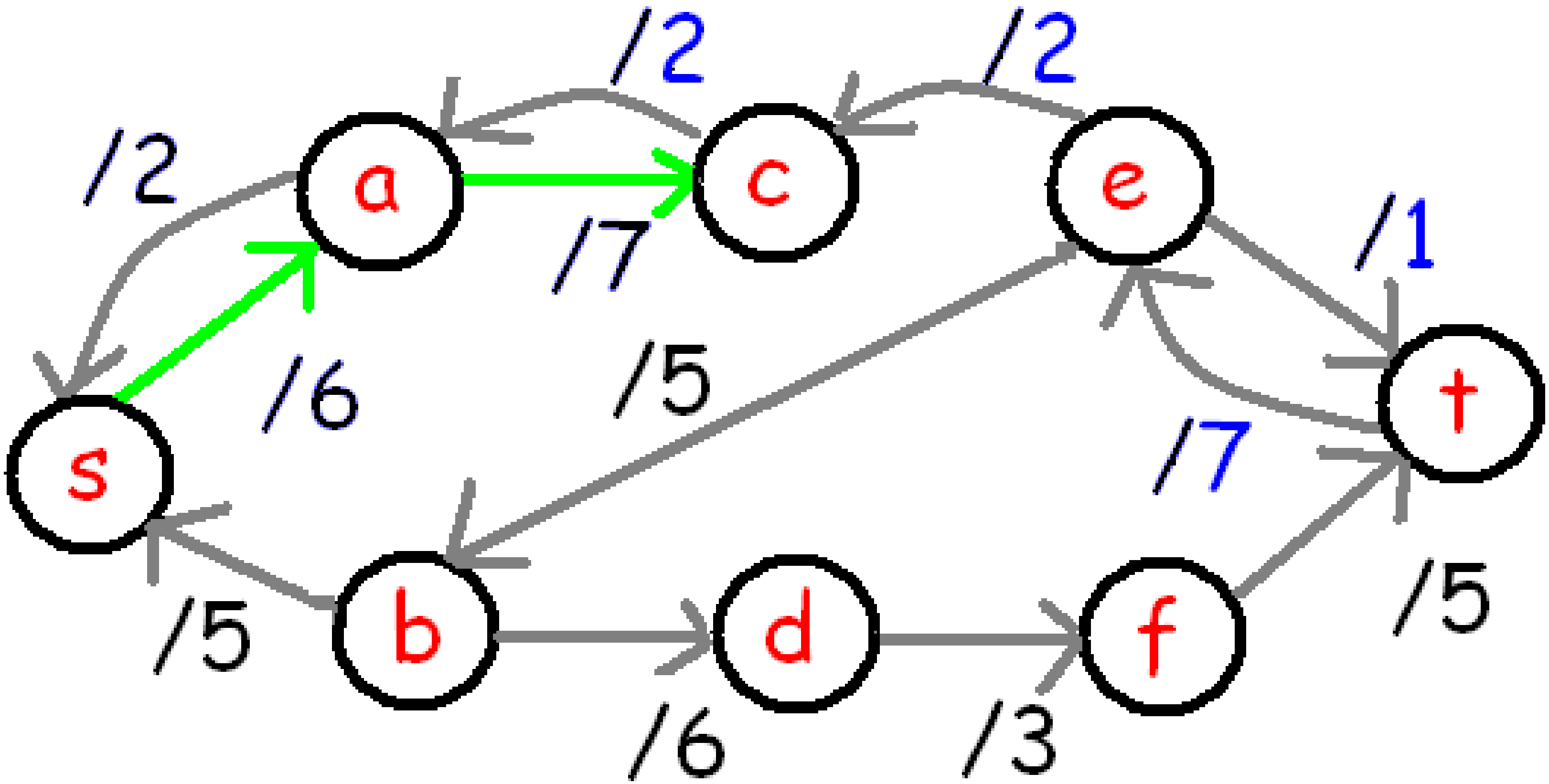
Example



Learn DAA : From B K Sharma

Edmonds-Karp Algorithm

Example

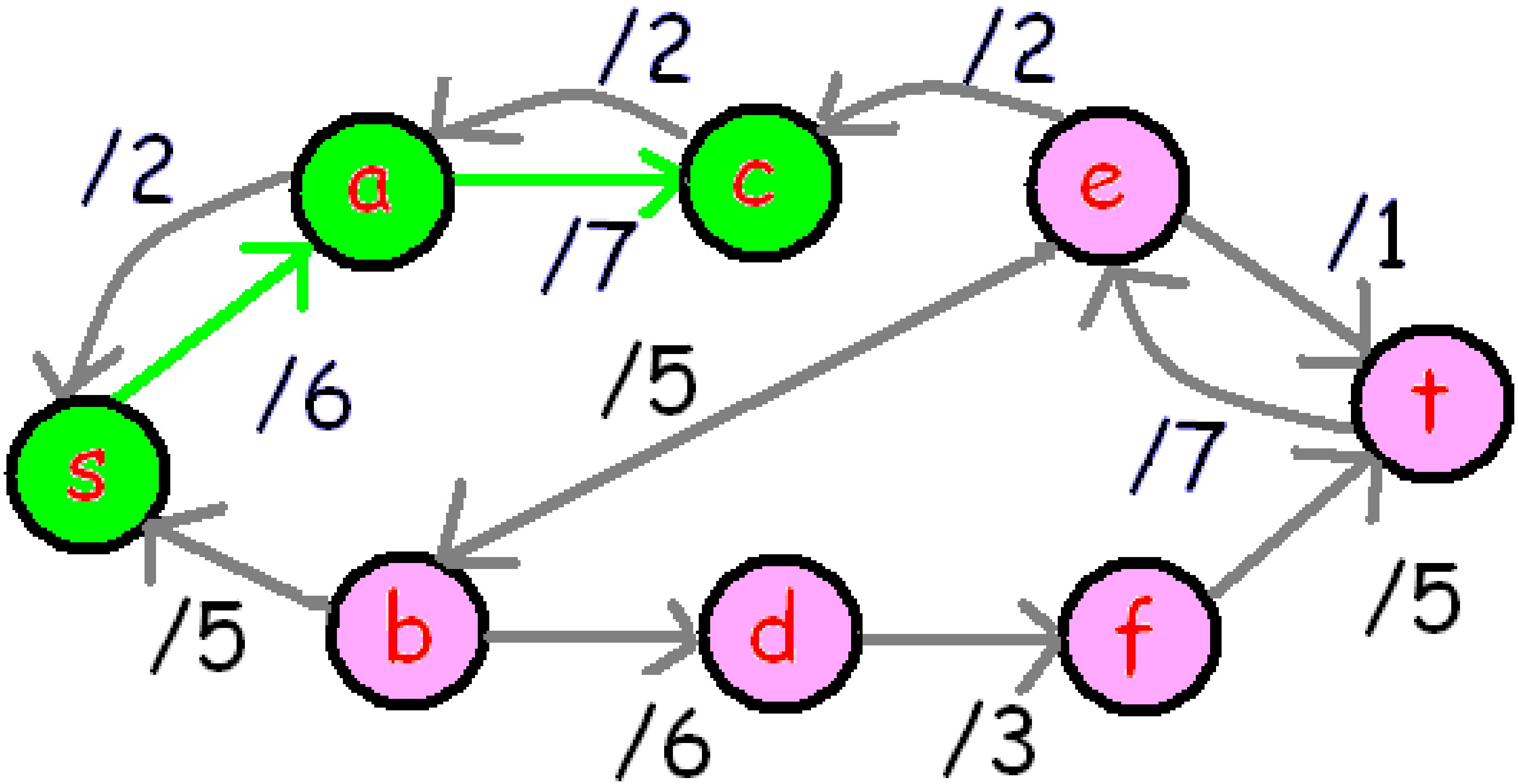


Apply BFS: Can not reach t

Learn DAA : From B K Sharma

Edmonds-Karp Algorithm

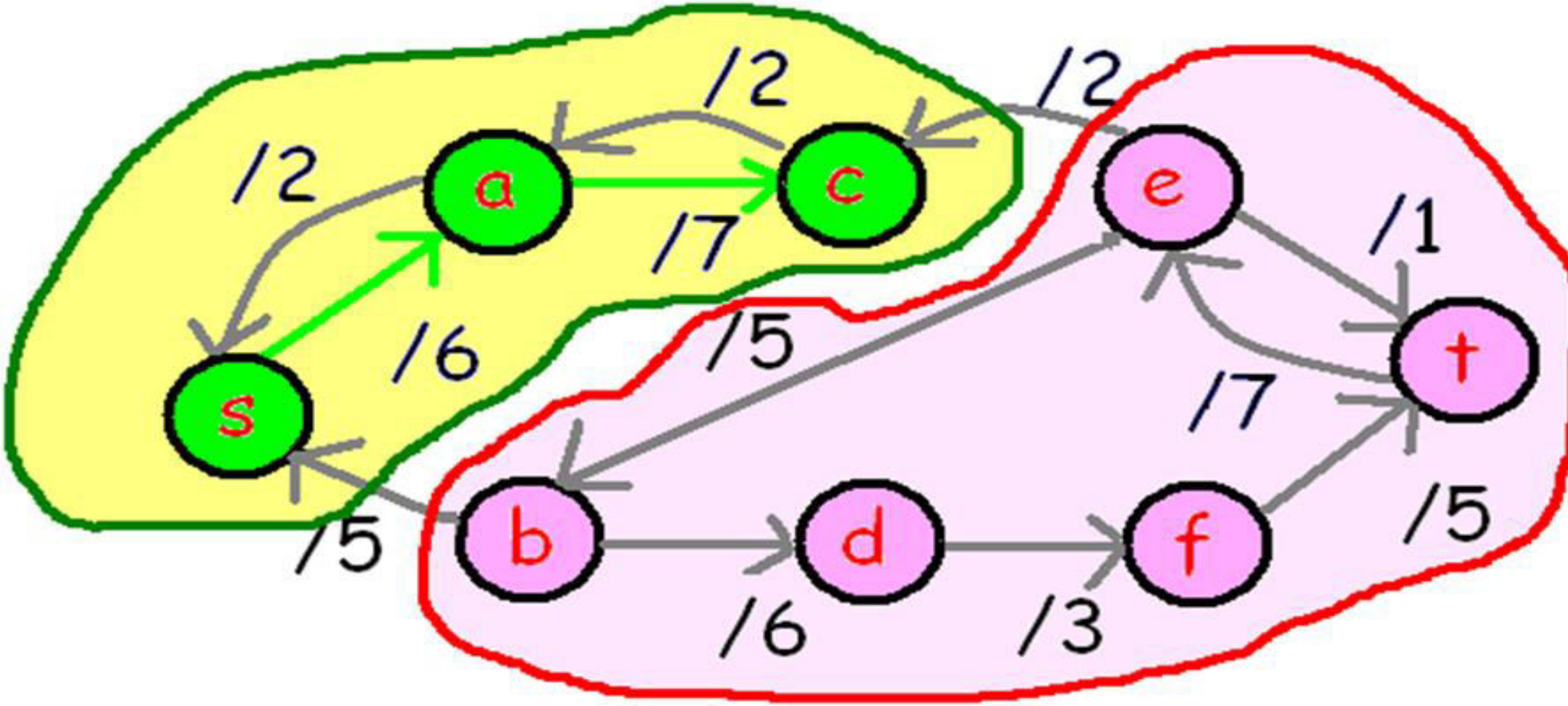
Example



Learn DAA : From B K Sharma

Edmonds-Karp Algorithm

Example

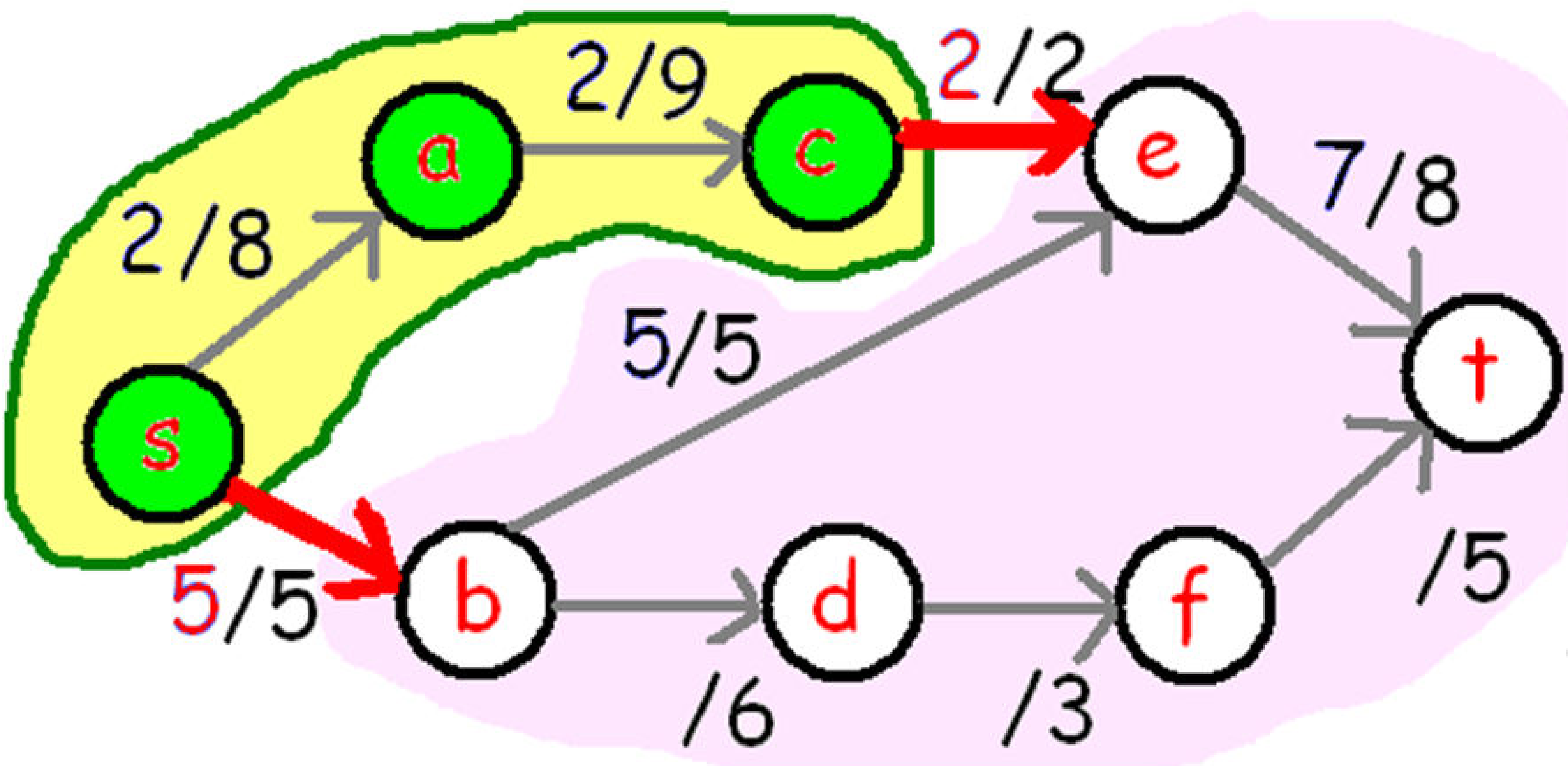


S={s, a, c}   T = {b, d, e, f, t}

Learn DAA : From B K Sharma

Edmonds-Karp Algorithm

Example



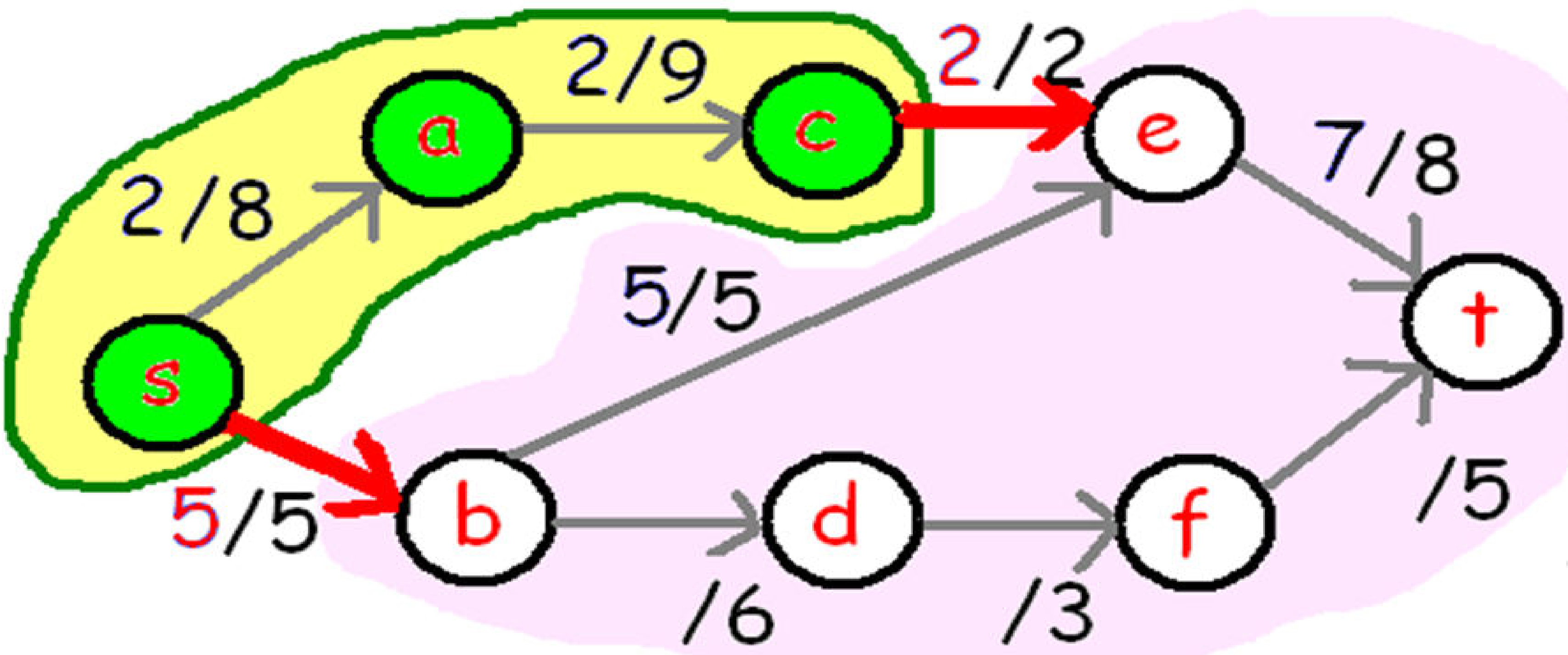
So (S, T)= { (s, b), (c, e)}

Learn DAA : From B K Sharma

Edmonds-Karp Algorithm

Min Cut /Max-Flow Theorem:

In any network, the value of a maximum flow is equal to the capacity of a minimum cut.



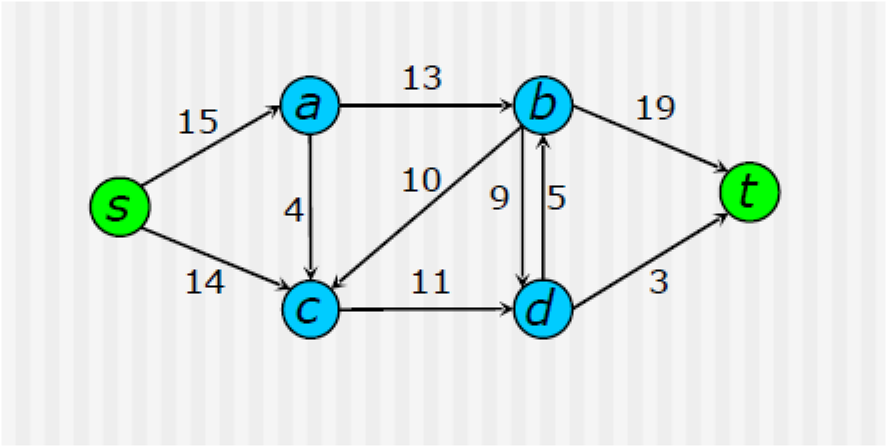
Edmonds-Karp Algorithm

The running time of Ford-Fulkerson's algorithm is $O(|f^*|(n + m))$ where f* is a maximum flow, n is number of vertices and m is number of edges

The Running Time of Edmonds- Karp Algorithm is $O(n\ m^2)$

Edmonds-Karp Algorithm

- ***Run the Edmonds-Karp algorithm on the following graph:***



END