A circulant graph $C_n(j_1, j_2, \ldots, j_k)$
has n vertices numbered 0, 1, 2, …, n-1
and there is an edge between vertices u and v
if $u + j_r$ (modulo n) = v for some r= 1, 2, …, k.

Draw the circulants
$C_6(1,3)$,
$C_8(1,4)$,
$C_{10}(1,5)$ and
$C_{12}(1,6)$
and find a minimum dominating set for each one.

When do circulant graphs have perfect dominating sets (perfect codes)?

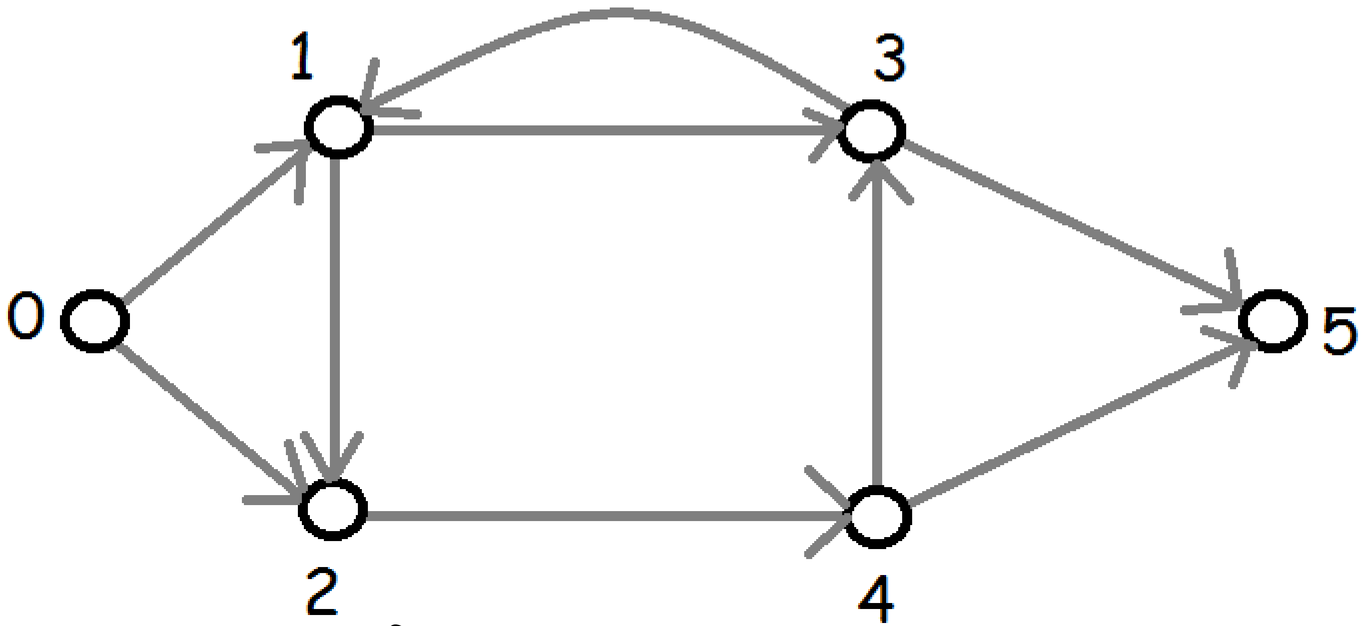
Announcements:

The literature review is due Friday June 23 at 11:55pm.

If you are searching for a paper/book in Mathscinet by author use the last name:
West
Or use
West, Douglas

Not
Douglas West

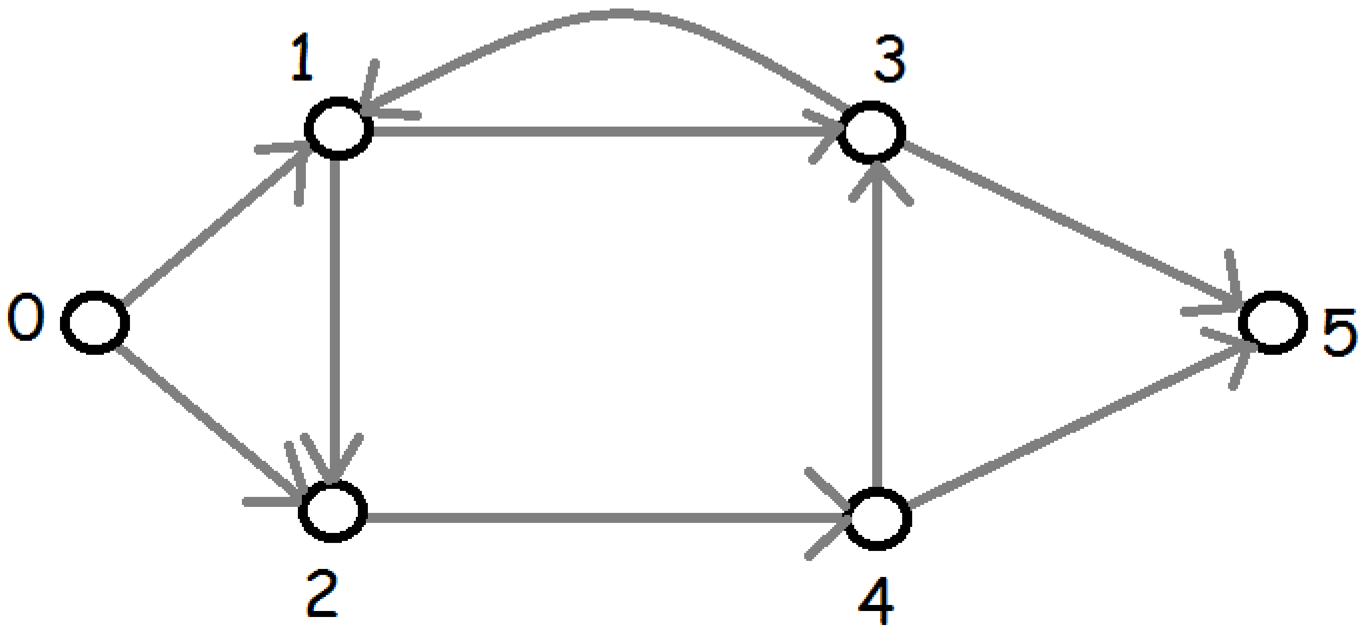
A directed graph G consists of a set V of vertices and a set E of arcs where each arc in E is associated with an ordered pair of vertices from V.



V={0, 1, 2, 3, 4, 5}

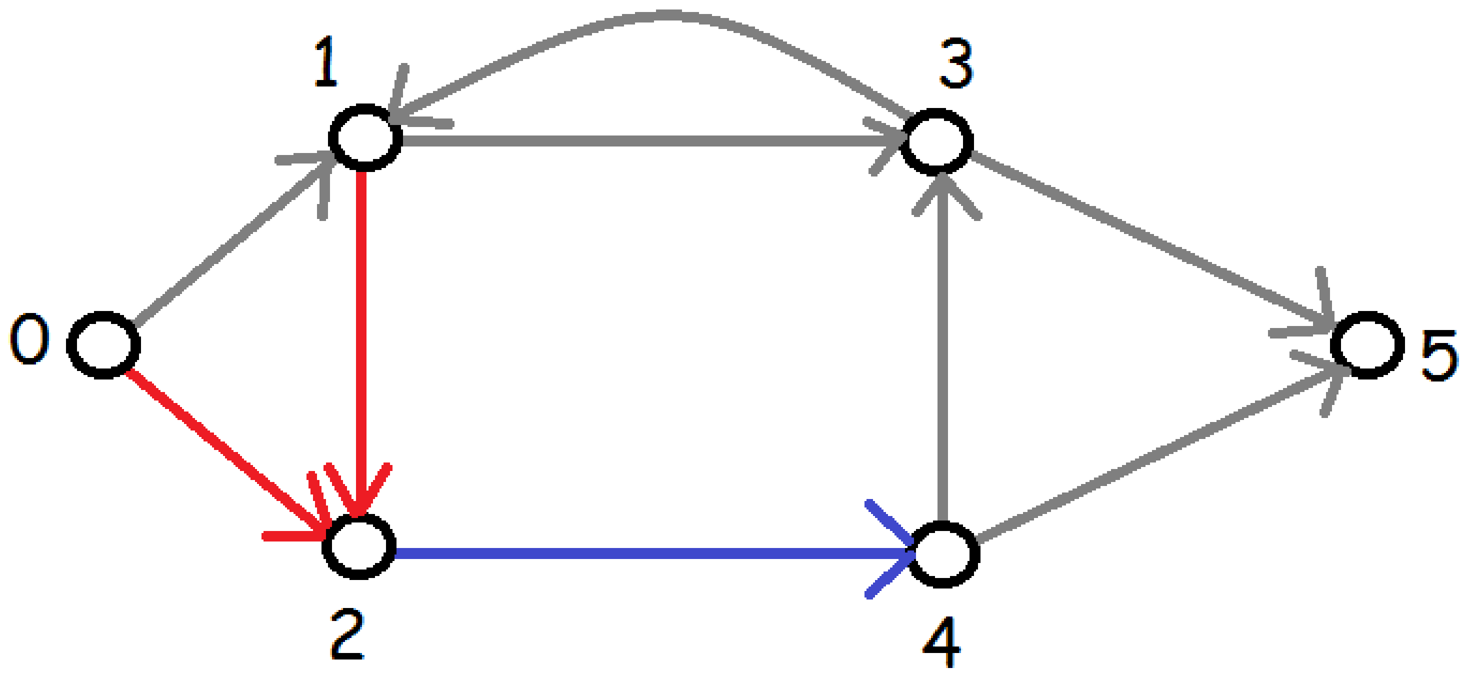
E= {(0,1), (0,2), (1,2), (1,3), (2,4),(3,1),(3,5),(4,3),(4,5)}

A directed graph G:



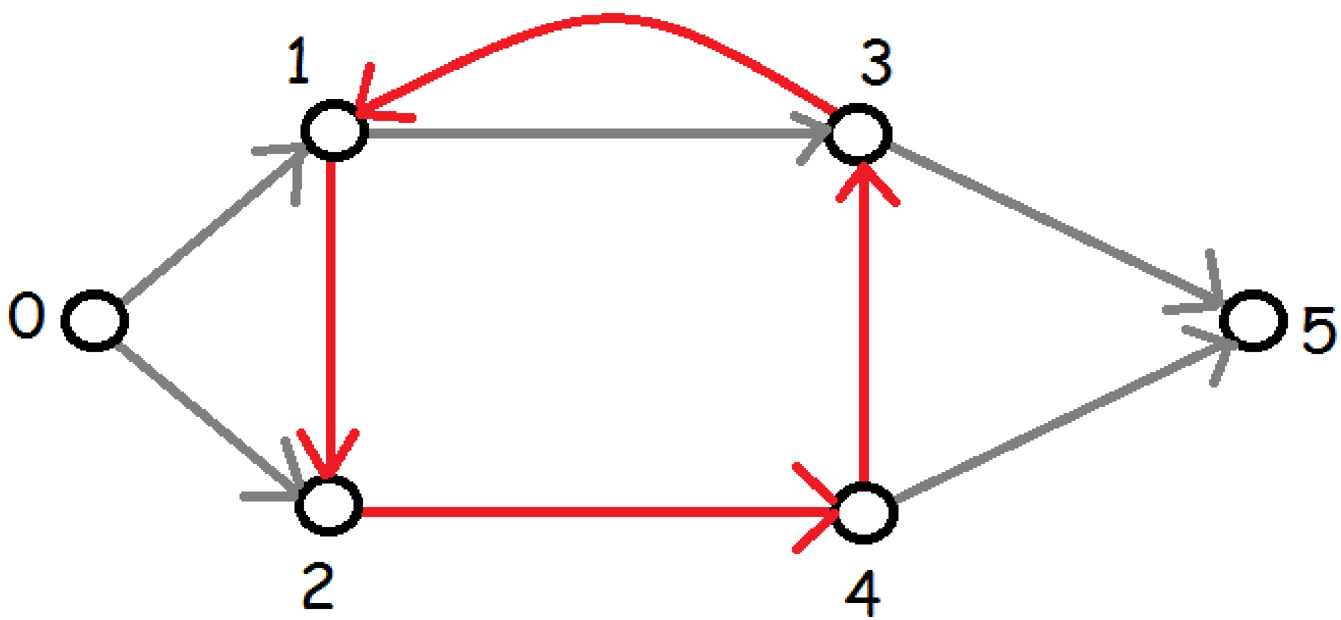
Remark: In a talk, I might just use the pictures without as many words, but I would not use words without pictures.

A directed graph G:



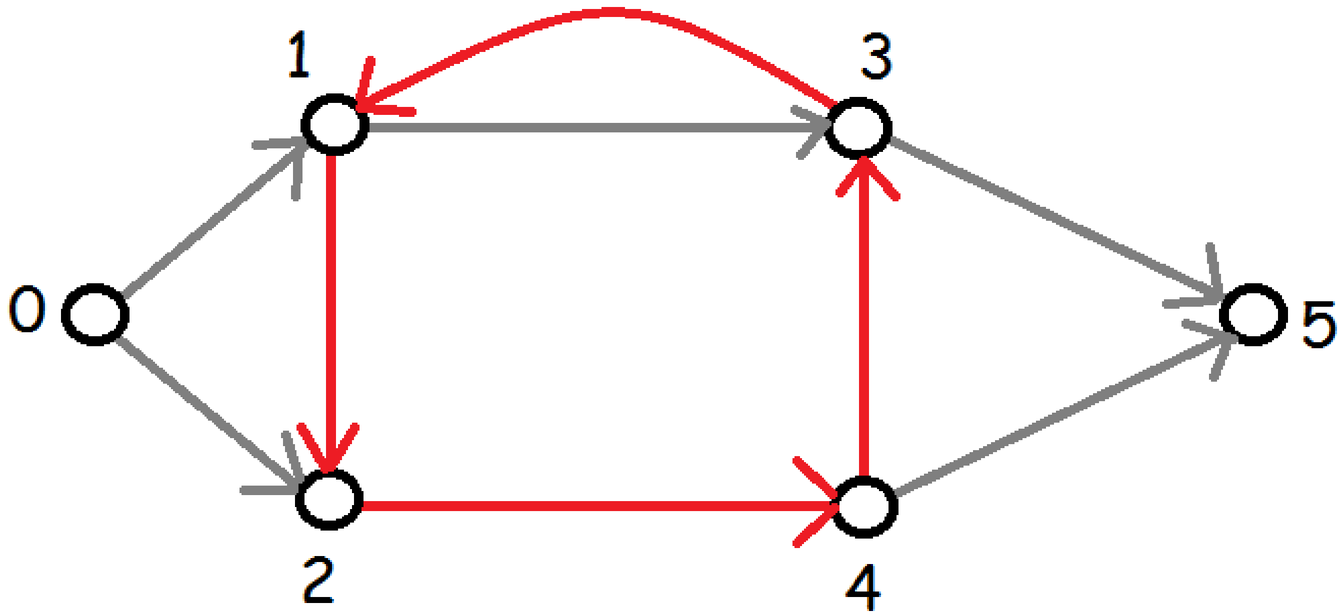
Vertex 2 has in-degree 2 and out-degree 1.

A directed cycle of length k consists of an alternating sequence of vertices and arcs of the form: $v_0, e_1, v_1, e_2, \ldots, v_{k-1}, e_k, v_k$ where $v_0 = v_k$ but otherwise the vertices are distinct and where e_{i+1} is the arc (v_i, v_{i+1}) for i= 0, 1, 2, ..., k-1.

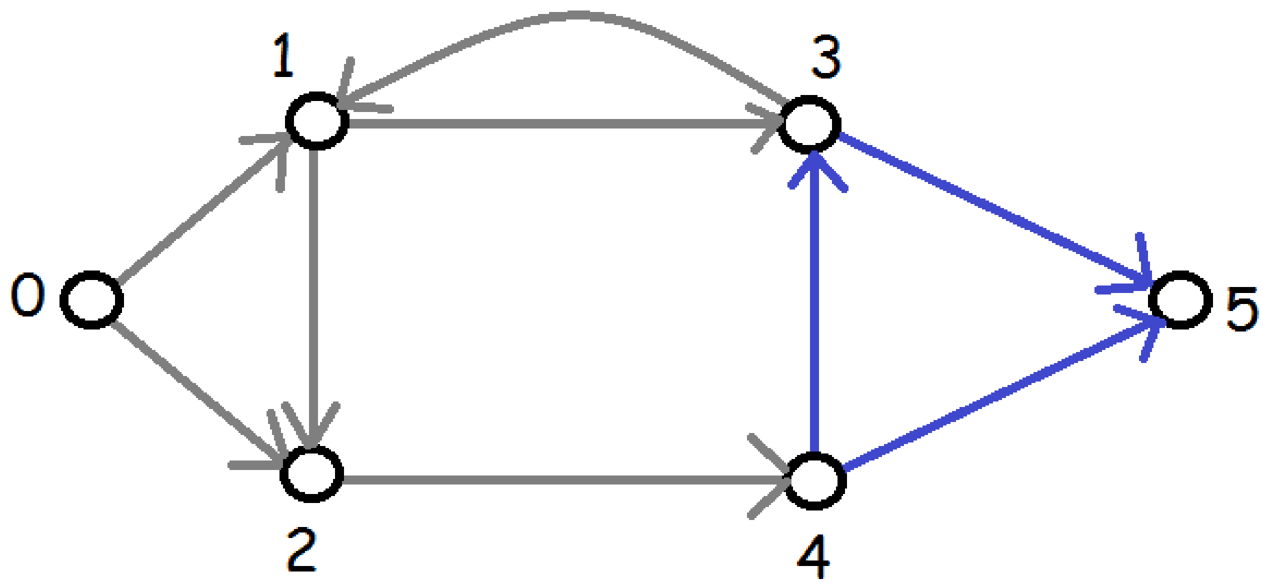


1, (1,2), 2, (2,4), 4, (4,3), 3, (3,1), 1

A directed cycle of length 4:

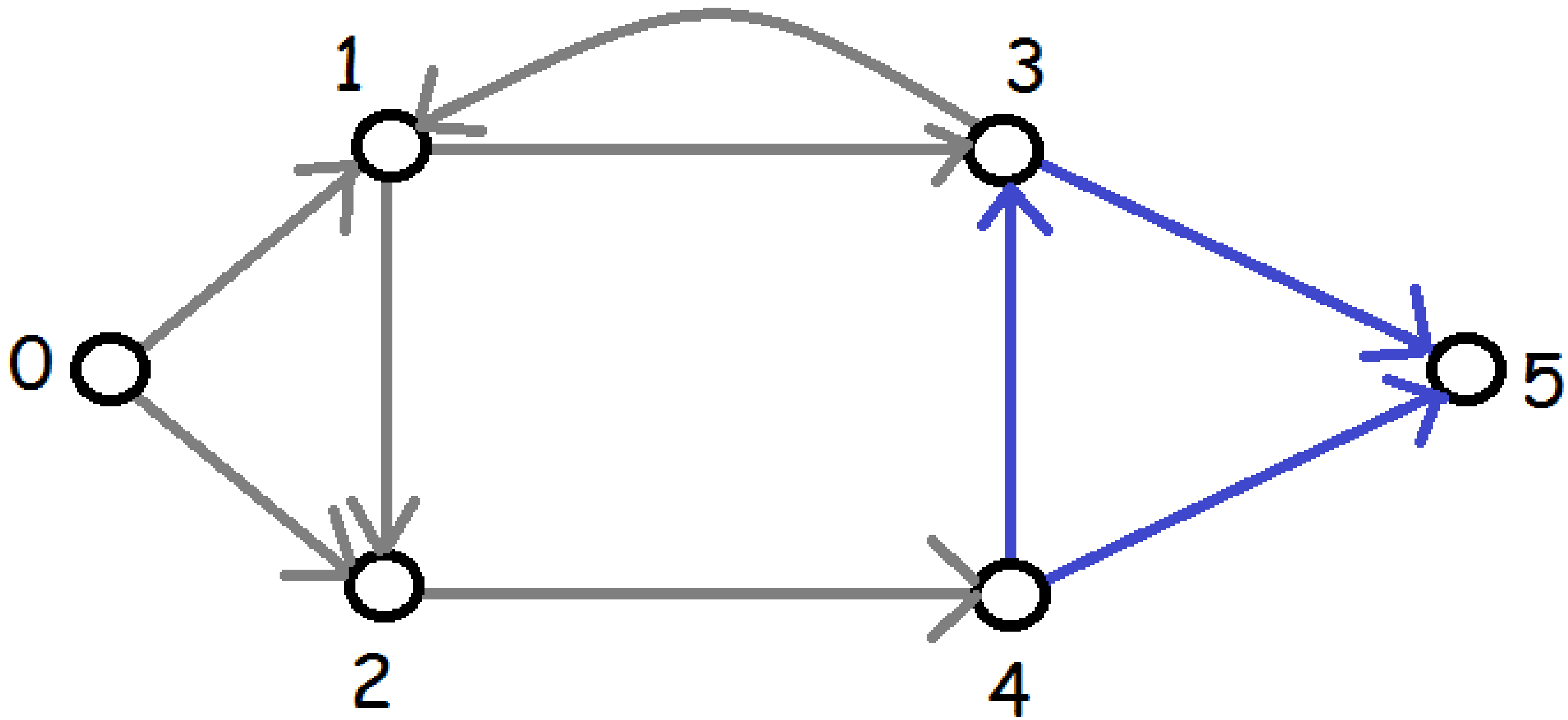


A cycle of length k consists of an alternating sequence of vertices and arcs of the form: v_0, e_1, v_1, e_2, … ,v_{k-1}, e_k, v_k where $v_0 = v_k$ but otherwise the vertices are distinct and where e_{i+1} is either the arc (v_i, v_{i+1}) or (v_{i+1}, v_i) for i= 0, 1, 2, …, k-1.

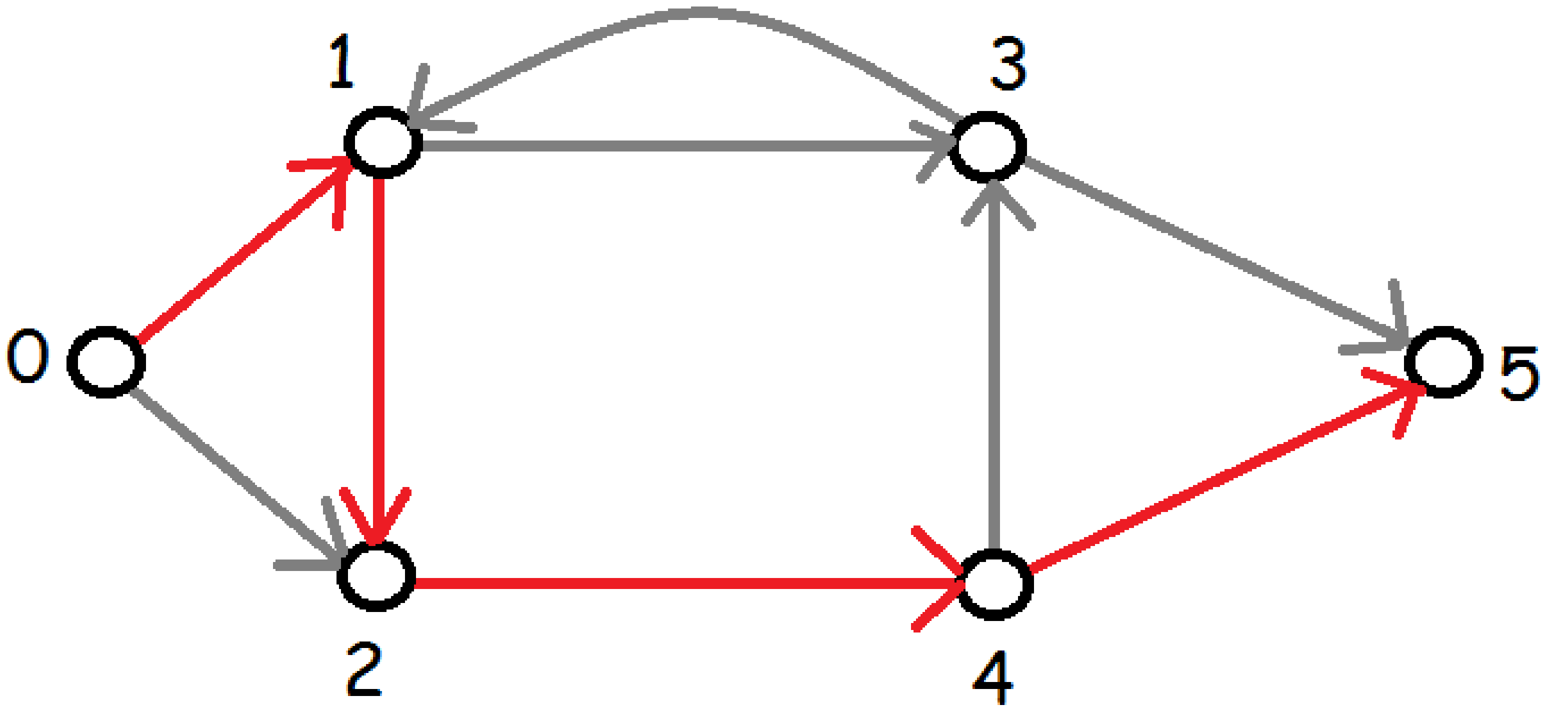


3, (3,5), 5, (4,5), 4, (4,3), 3

A cycle of length 3 which is not a directed cycle (arcs can be traversed in either direction):

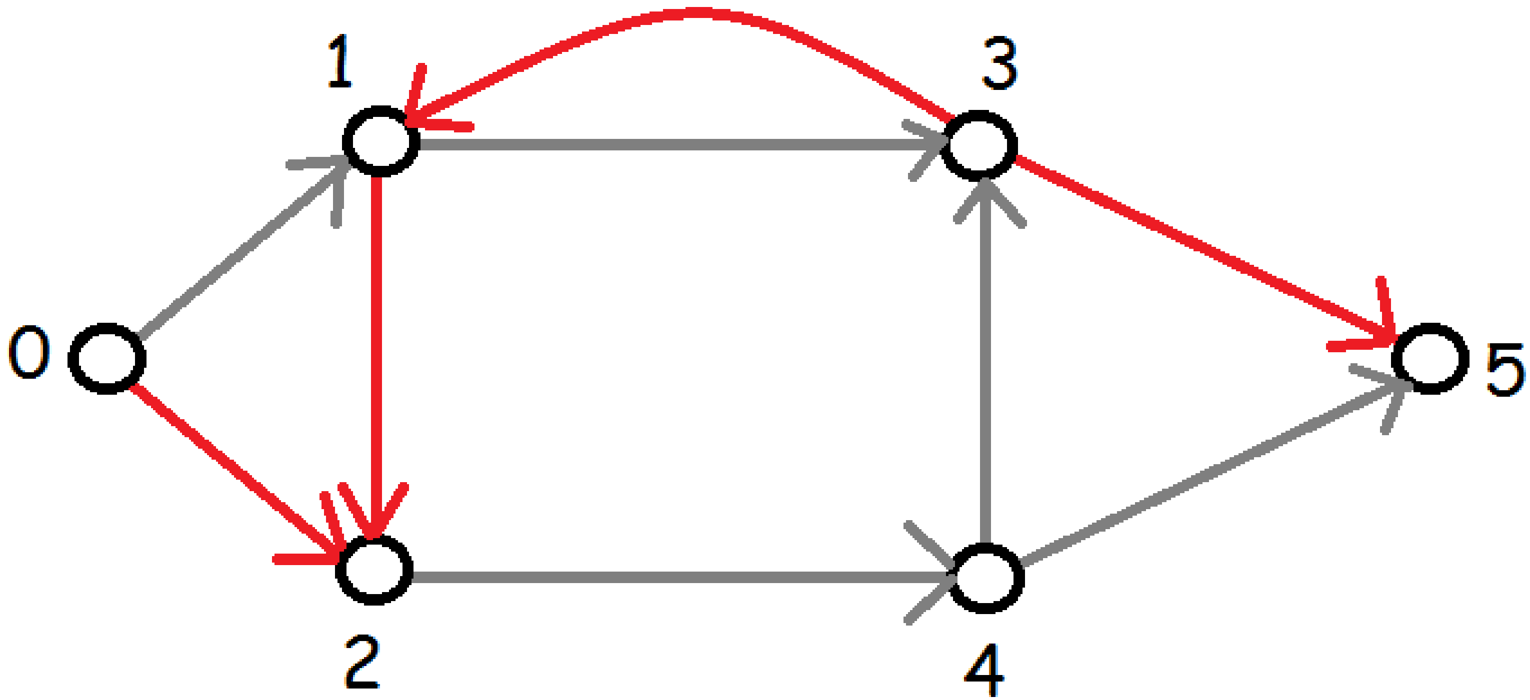


A directed path of length 4 from vertex 0 to vertex 5:



0, (0,1), 1, (1,2), 2, (2,4), 4, (4,5), 5

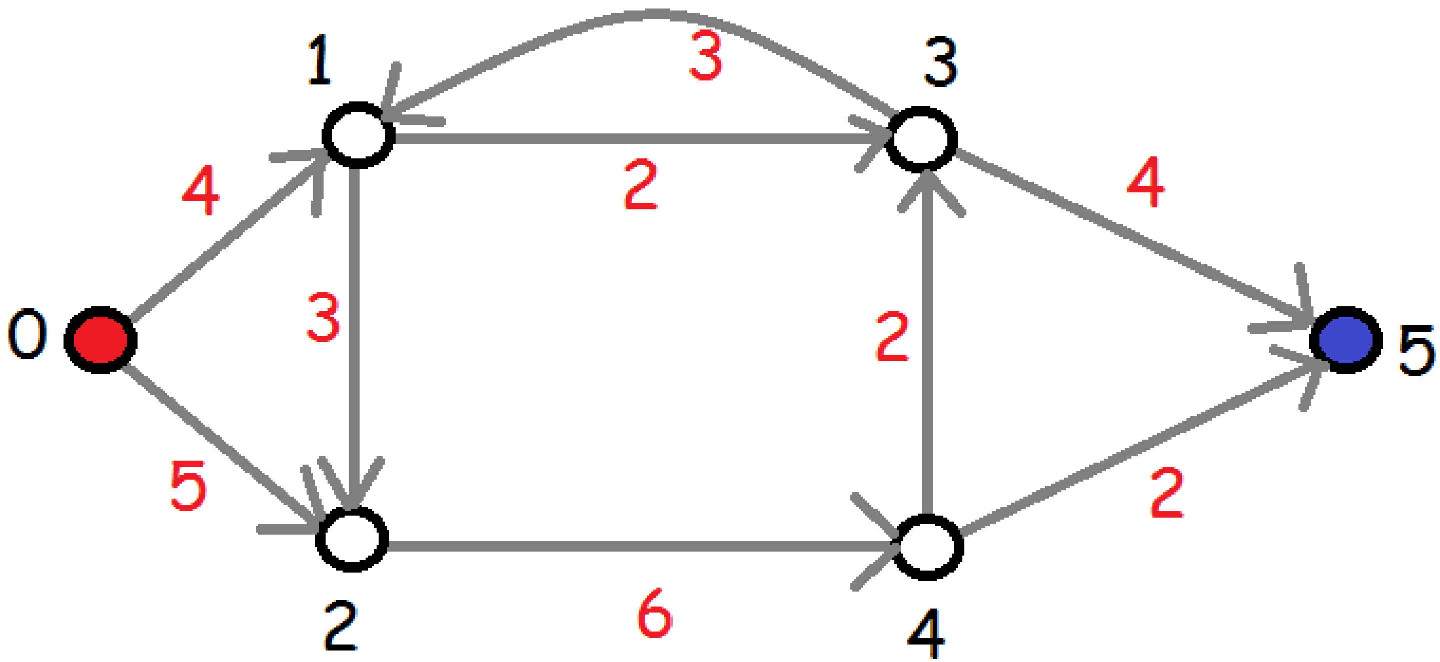
A path of length 4 which is not a directed path (arcs can be traversed in either direction) from vertex 0 to vertex 5:



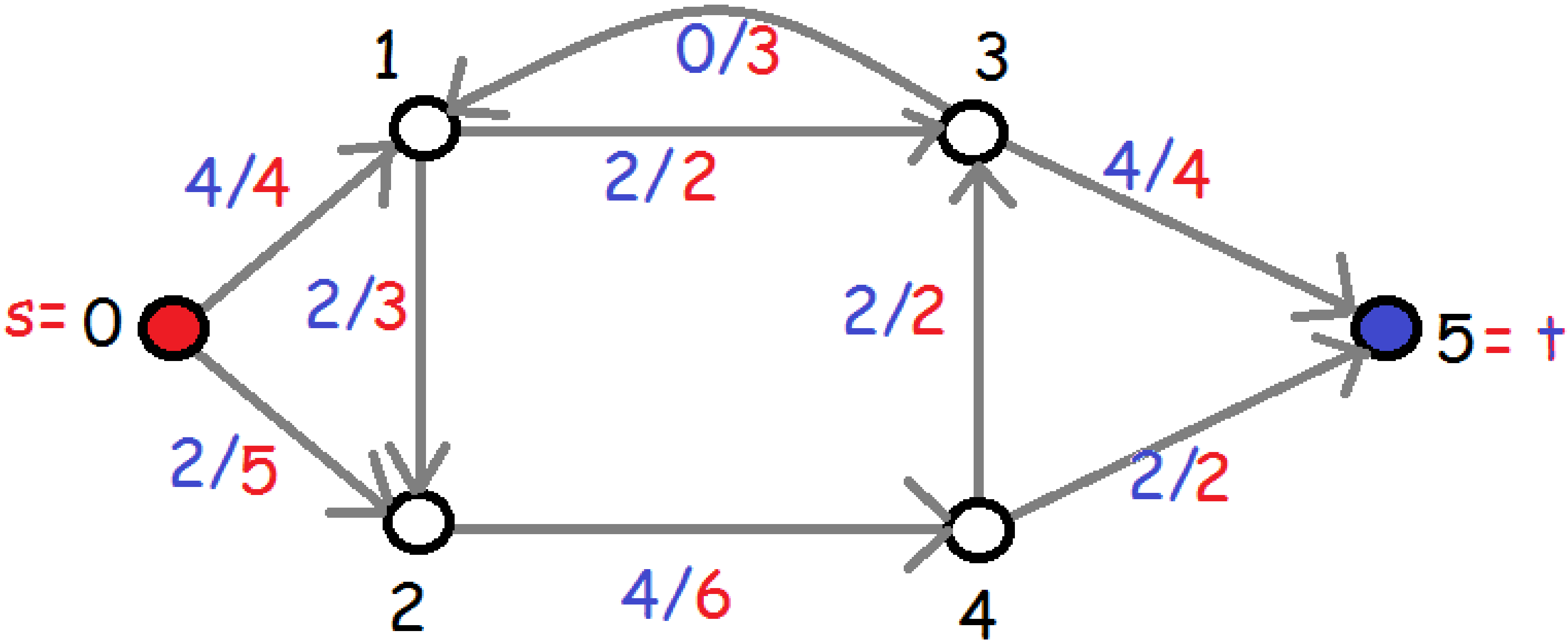
0, (0,2), 2, (1,2), 1, (3,1), 3, (3,5), 5

The maximum flow problem:

Given a directed graph G, a source vertex s and a sink vertex t and a non-negative capacity c(u,v) for each arc (u,v) , find the maximum flow from s to t.

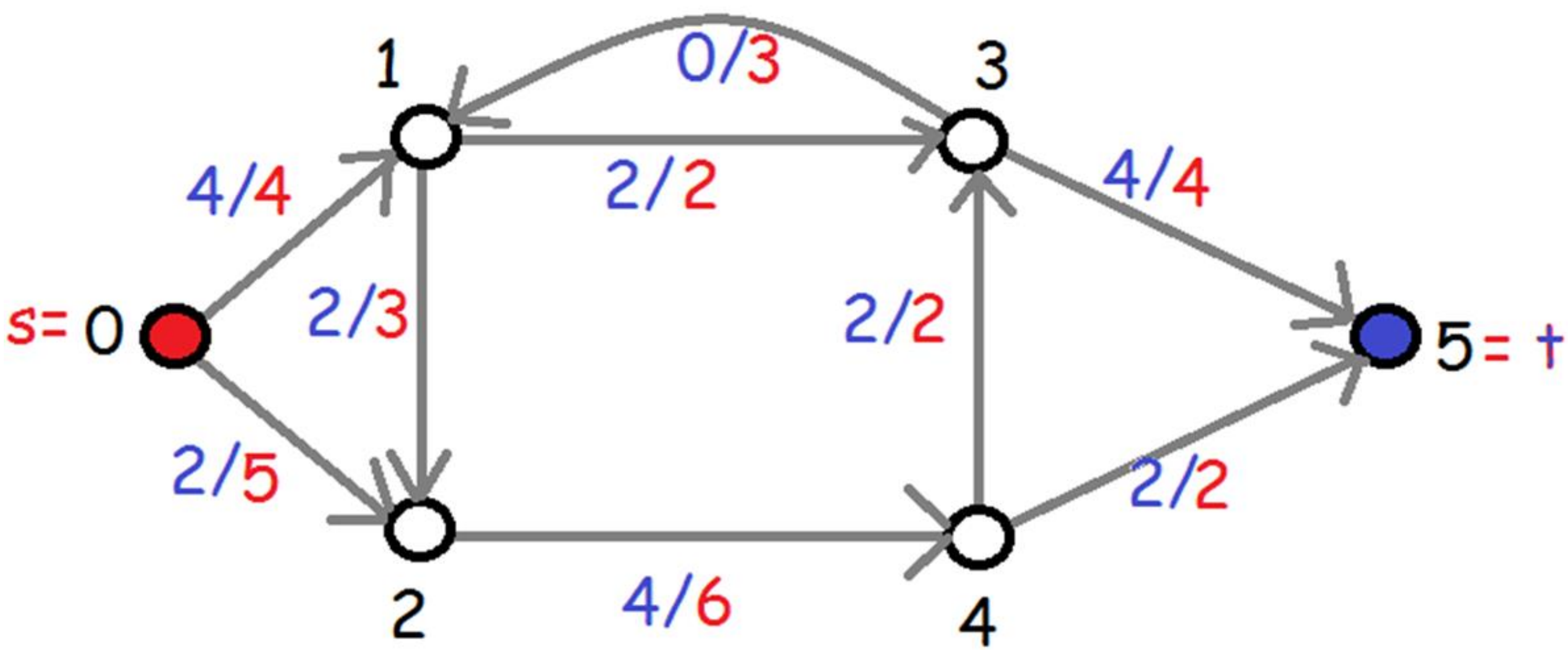


An example of a maximum flow:



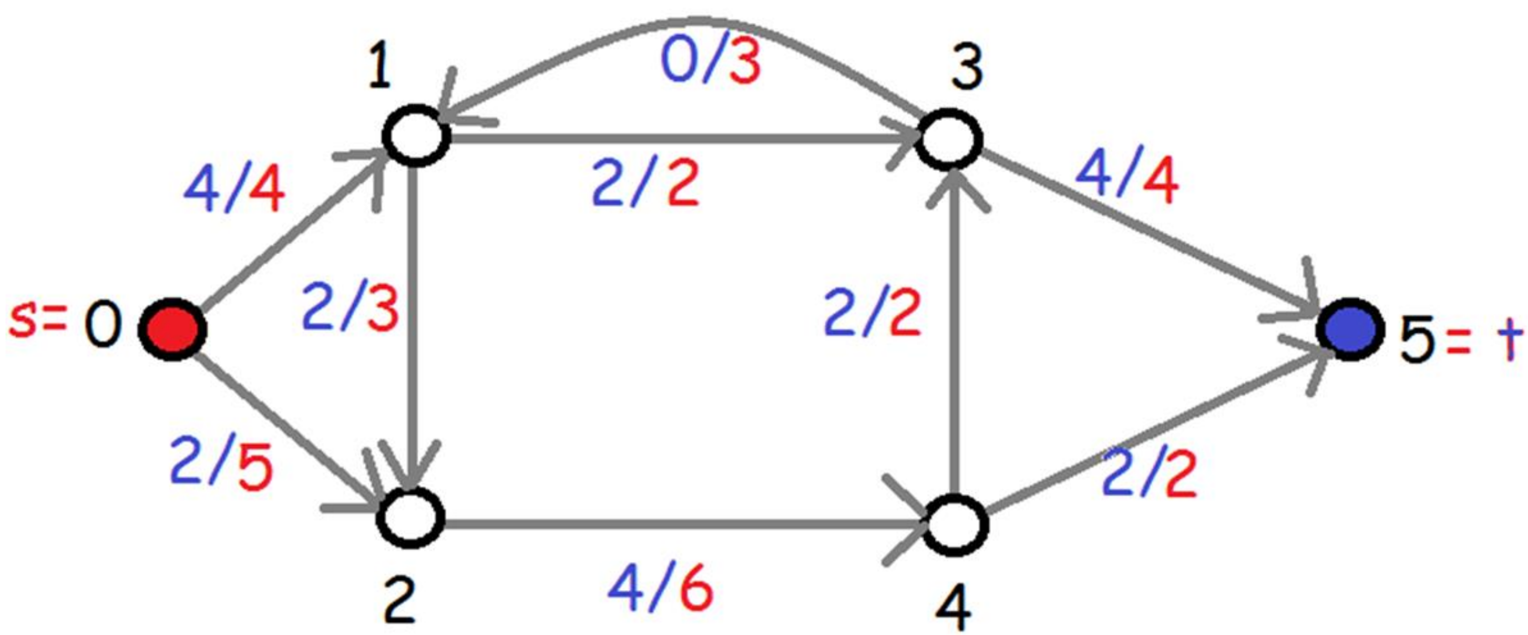
A flow function f is an assignment of flow values to the arcs of the graph satisfying:

1. For each arc (u,v), $0 \leq f(u, v) \leq c(u,v)$.
2. [Conservation of flow] For each vertex v except for s and t, the flow entering v equals the flow exiting v.



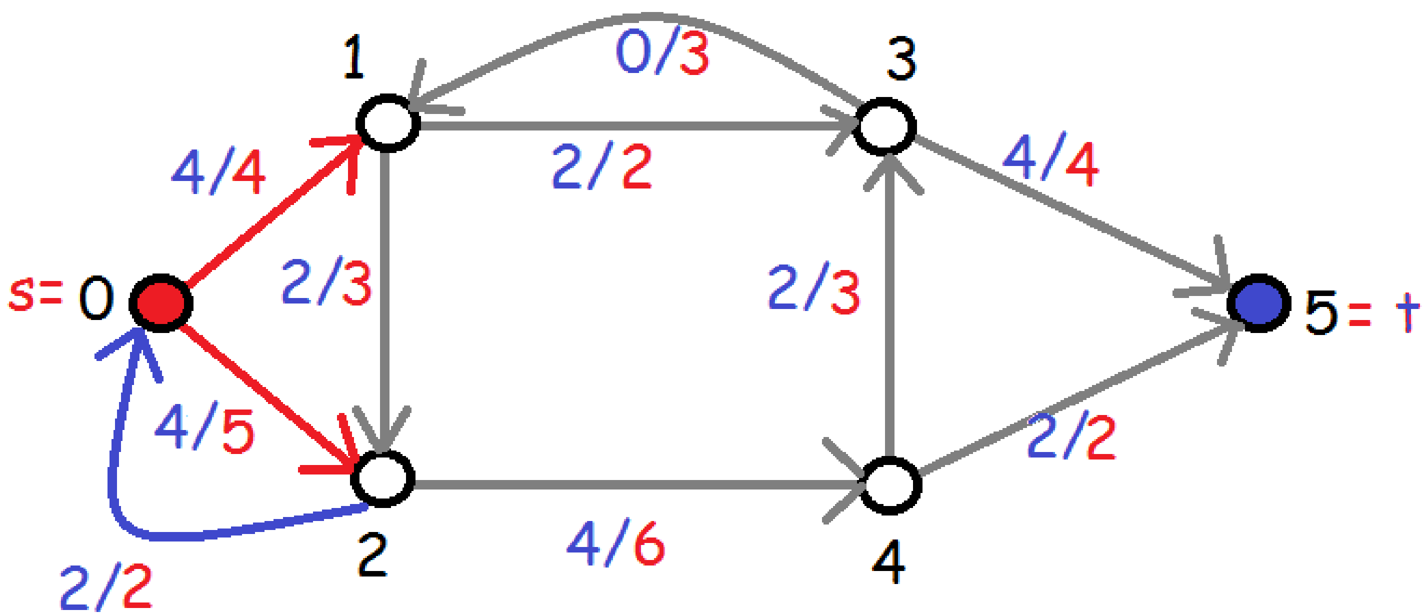
The amount of flow from s to t is equal to the net amount of flow exiting s
= sum over arcs e that exit s of f(e) –
sum over arcs e that enter s of f(e).

Flow = 6.

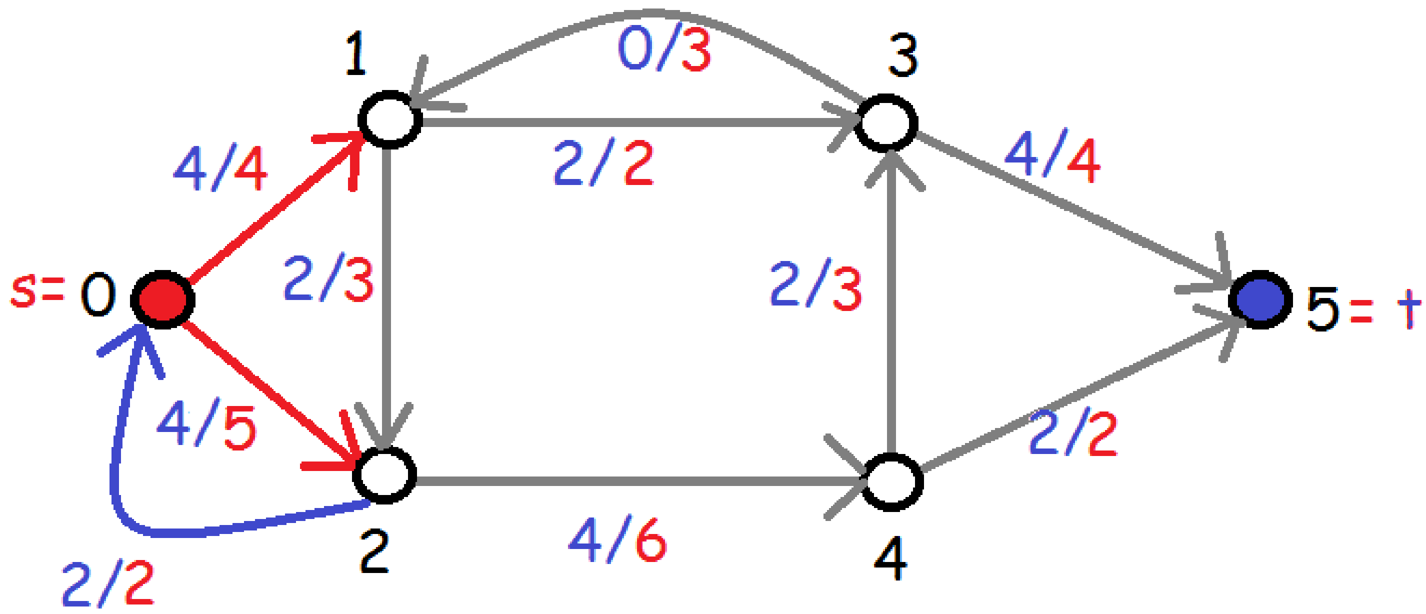


A slightly different example:

Flow= 4 + 4 - 2 = 6.



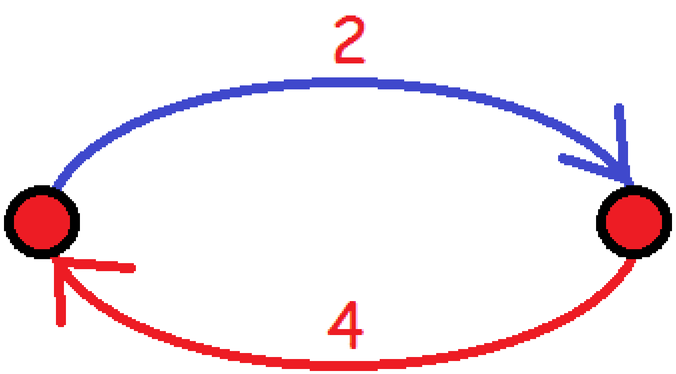
Because of conservation of flow,
the amount of flow from s to t is also equal to
the net amount of flow entering t.



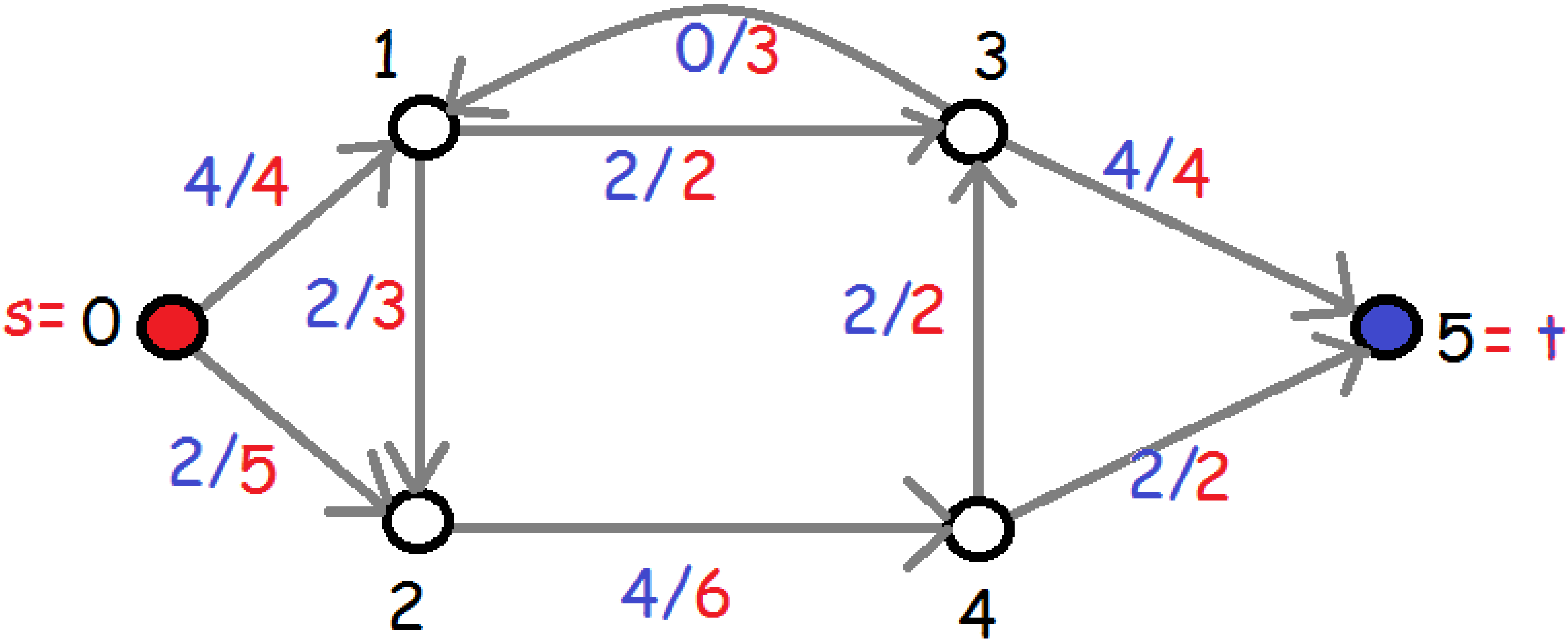
Form an auxillary graph as follows:
For each arc (u,v) of *G*:

1. Add an arc (u,v) with capacity c(u, v) - f(u,v).
2. Add an arc (v,u) with capacity f(u,v).

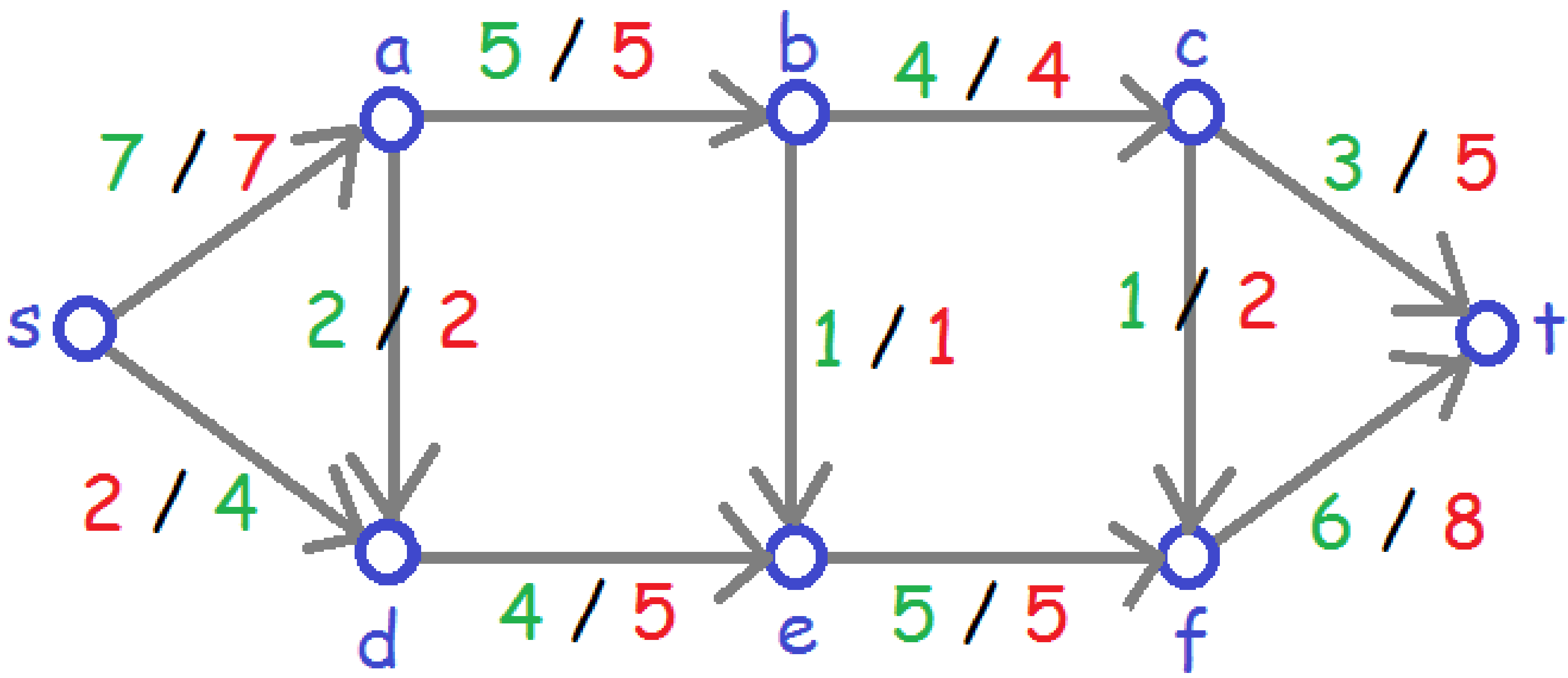
Auxillary graph:



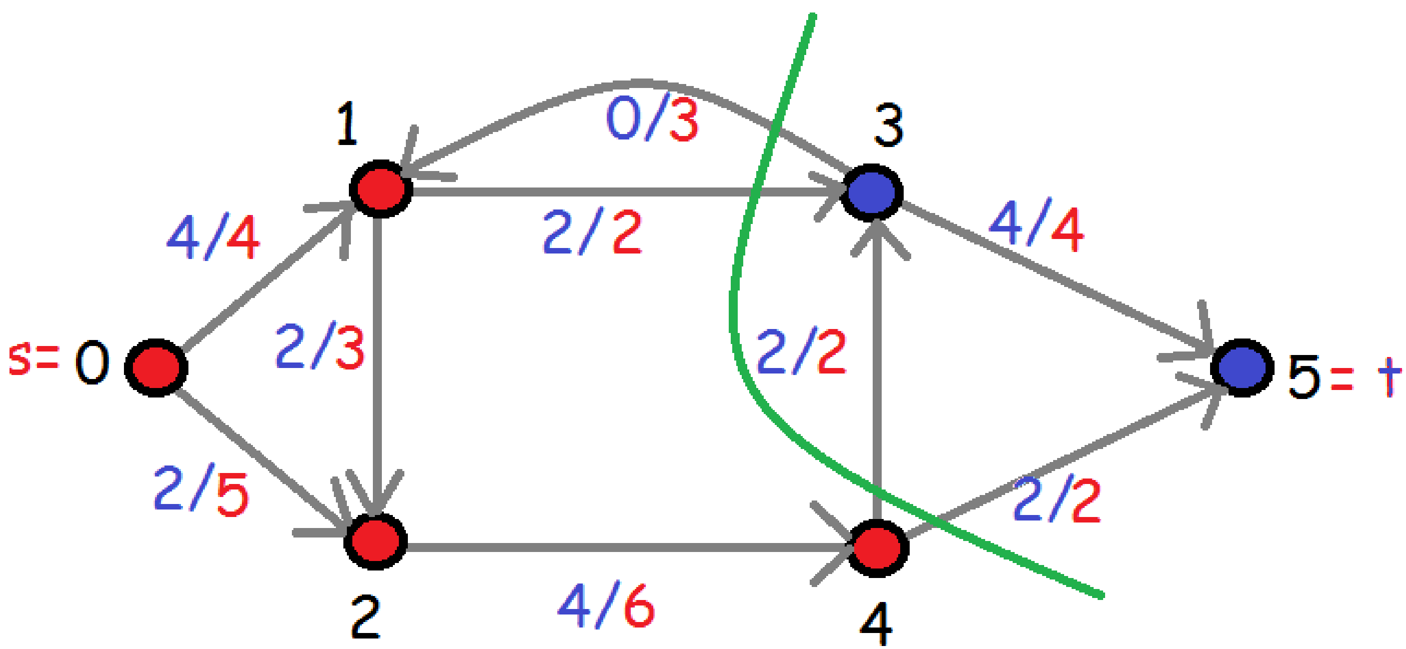
Make the auxillary graph for this example:



1. Create the auxillary graph for this flow.
2. Apply BFS to find the set of vertices reachable from s in the auxillary graph.
3. Which arcs are in the corresponding cut?



When the flow is maximum:
S= {v: v is reachable from s on a directed path of non-zero weighted arc s}
T= V-S. Then (S,T) is a minimum capacity s,t-cut of the graph.

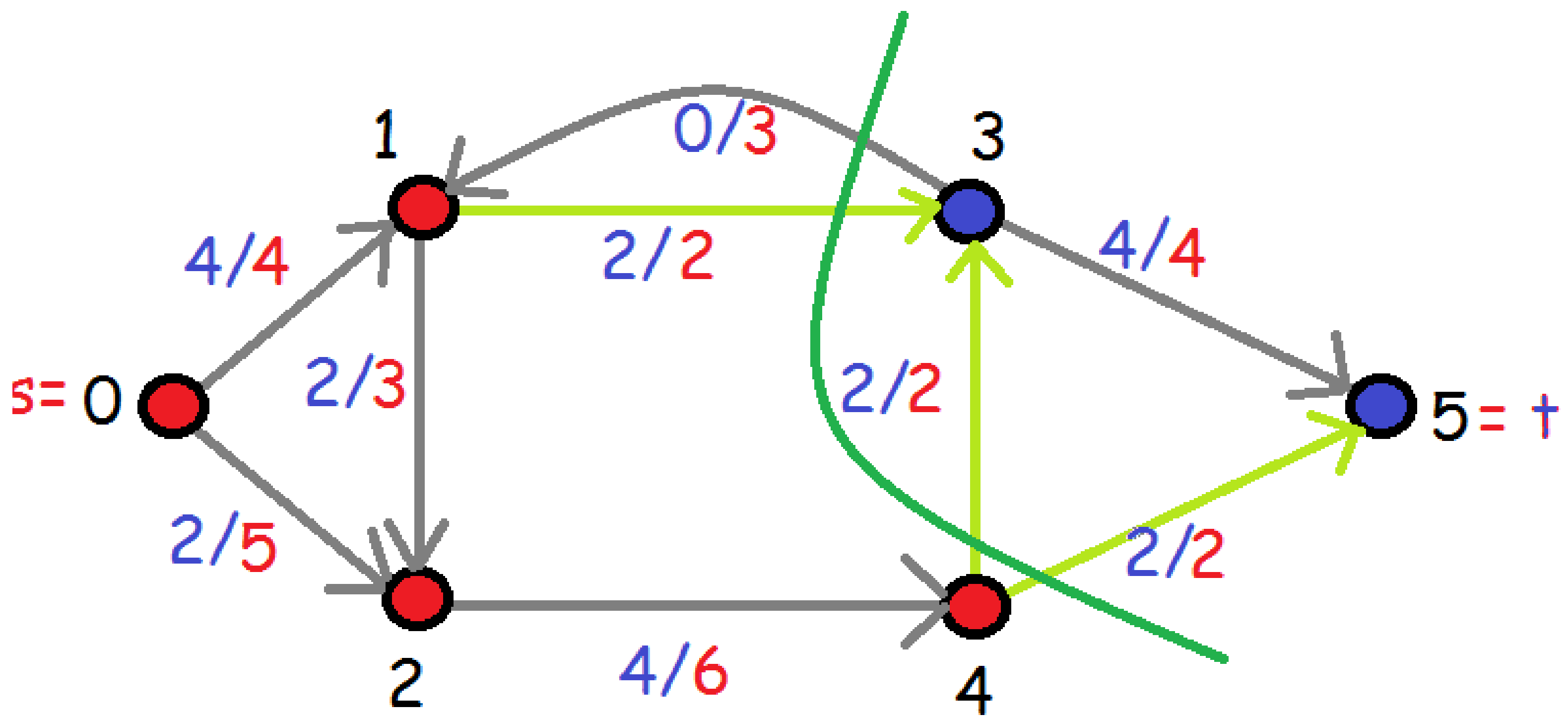


S= {0, 1, 2, 4}, T= {3, 5}

(S, T)= {(u, v): u ∈ S and v ∈ T}.

(S,T)= {(1,3), (4,3), (4,5)}

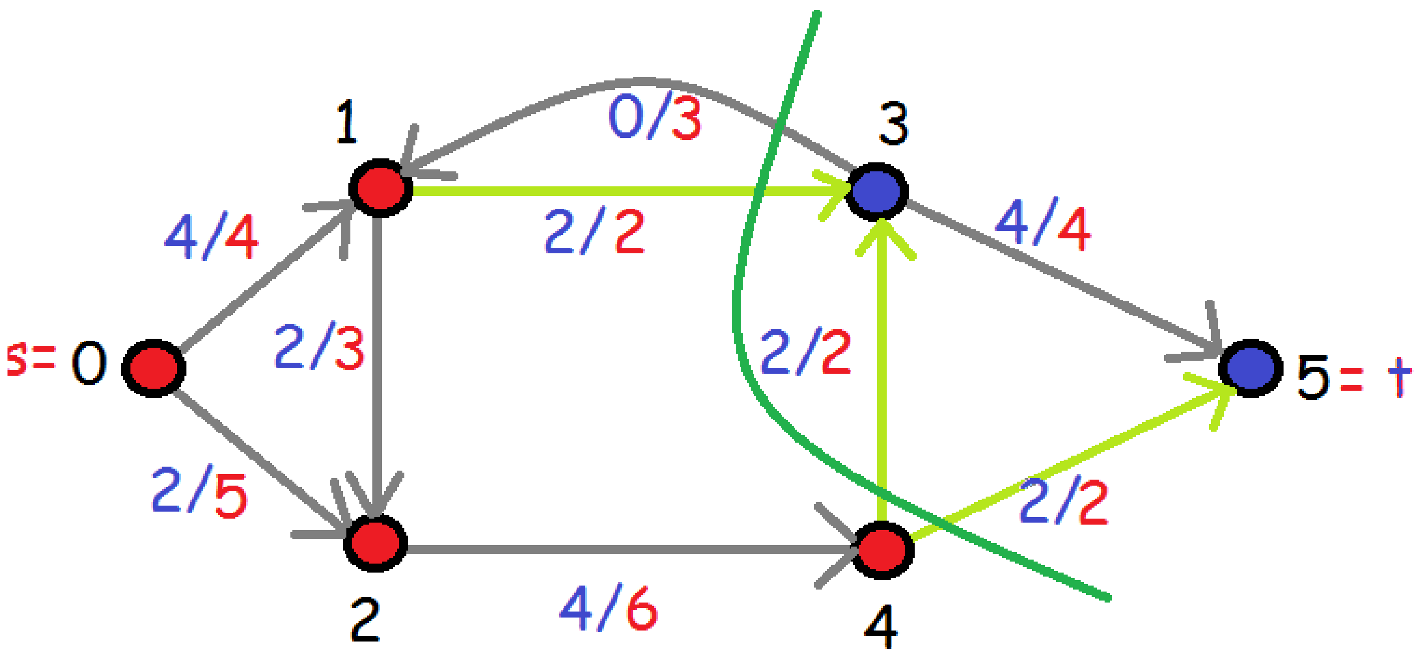
This is a cut because if you remove these edges there are no directed paths anymore from s to t.



The capacity of a cut (S,T) is the sum of the capacities of the arcs in the cut.

(S,T)= {(1,3), (4,3), (4,5)}

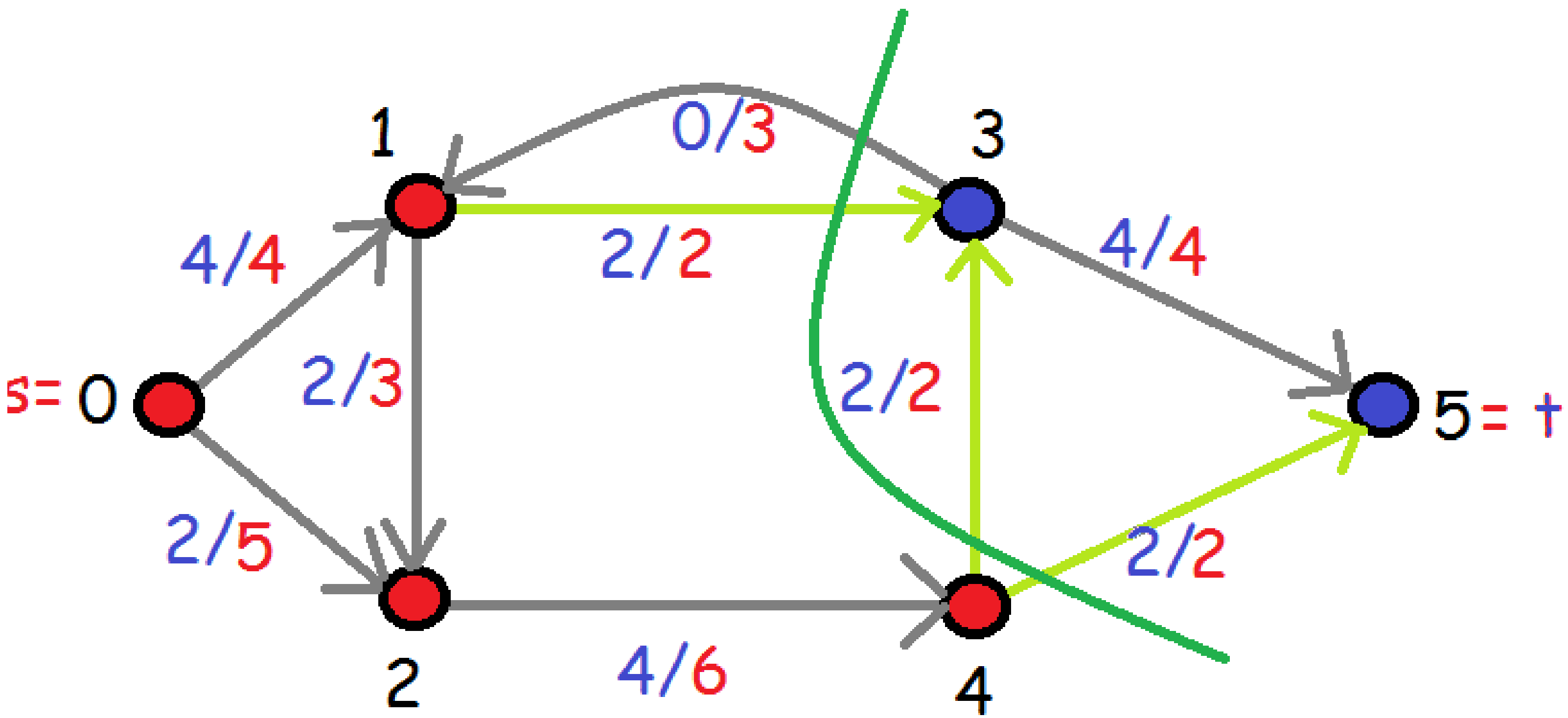
Capacity(S,T)= 2 + 2 + 2 = 6.



The maximum flow from s to t cannot be more than the capacity of any of the s,t-cuts.

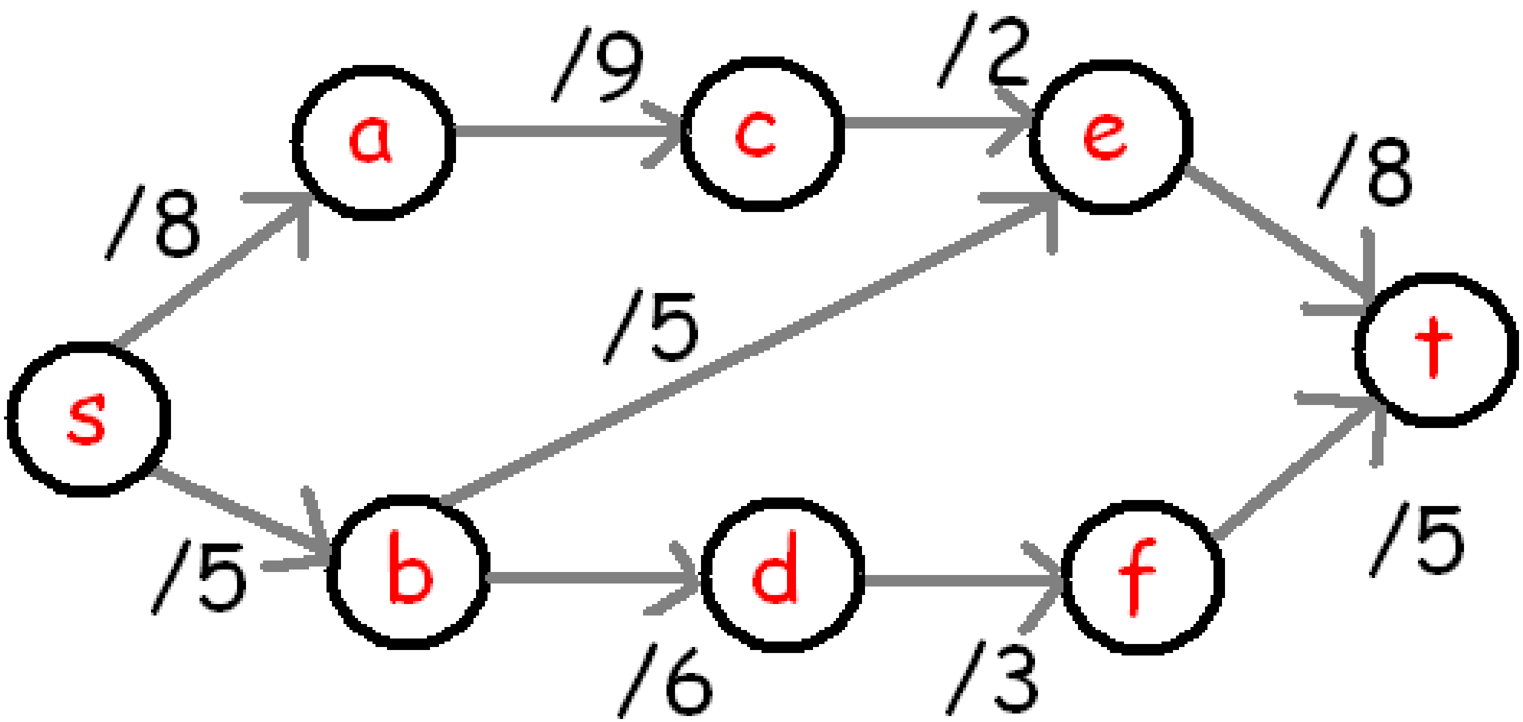
Theorem: the maximum flow equals the minimum capacity of an s,t-cut.

How can we find a maximum flow?



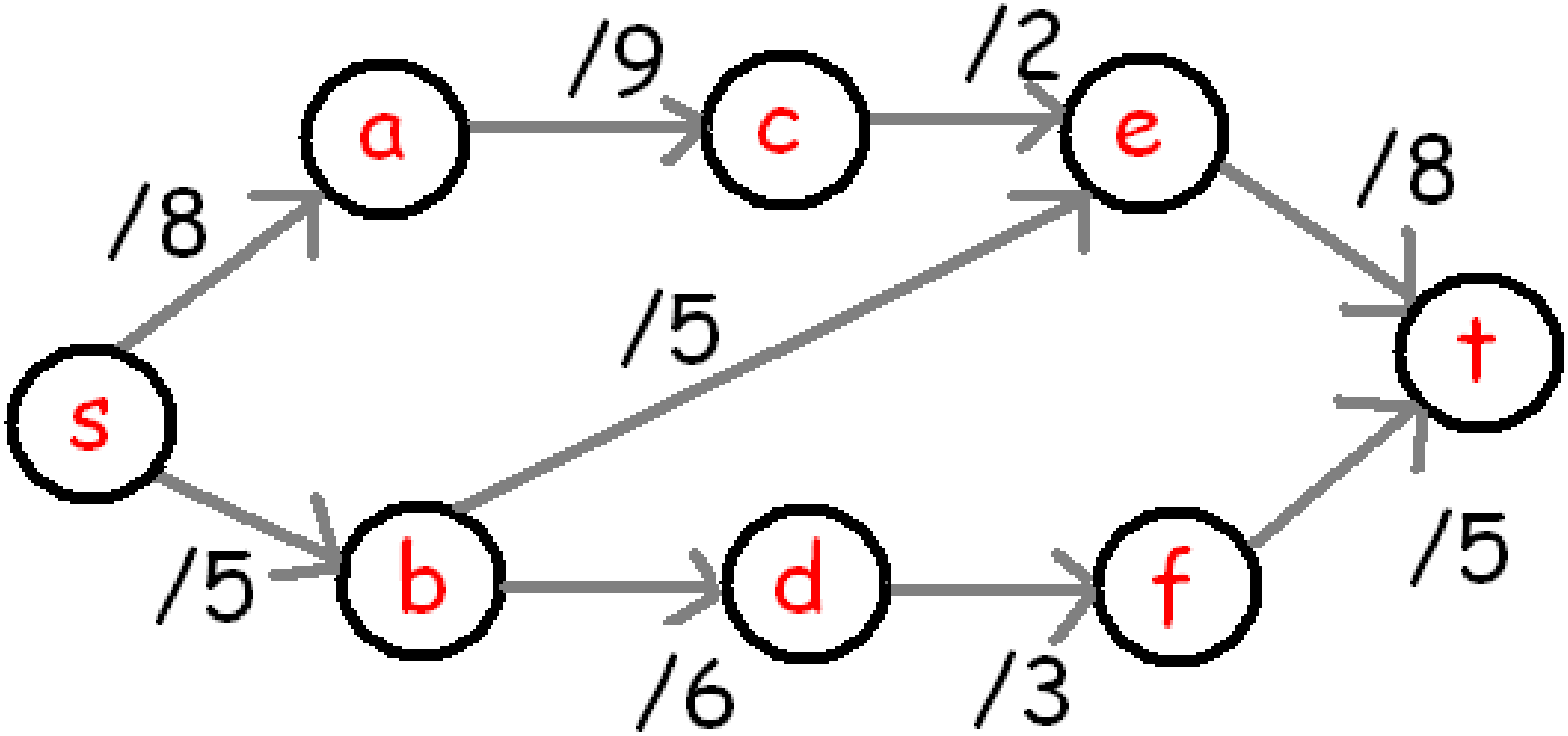
Use the Edmonds-Karp Algorithm to find the maximum flow in this network.

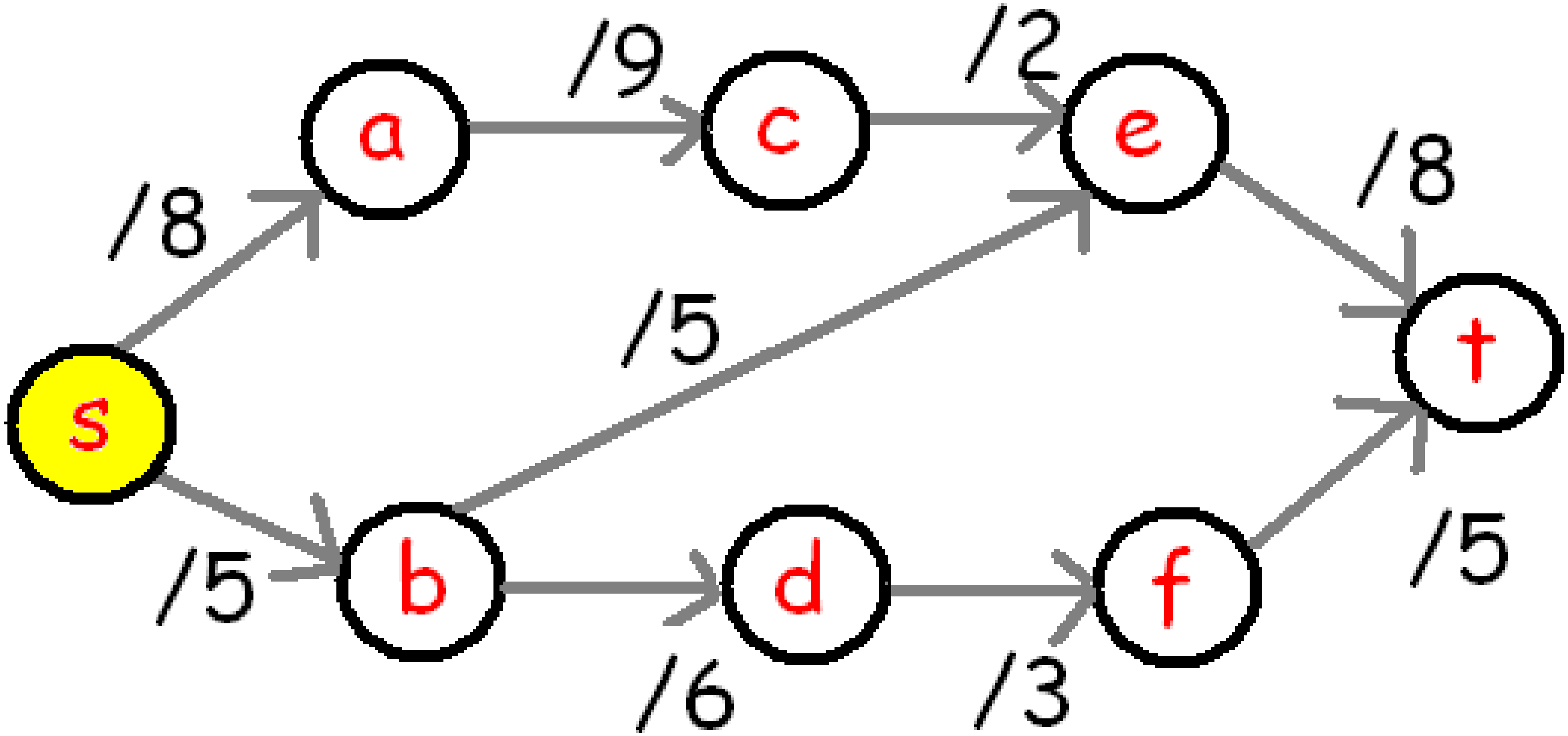
Edmonds-Karp: Use BFS to find augmenting paths in the auxillary graph.

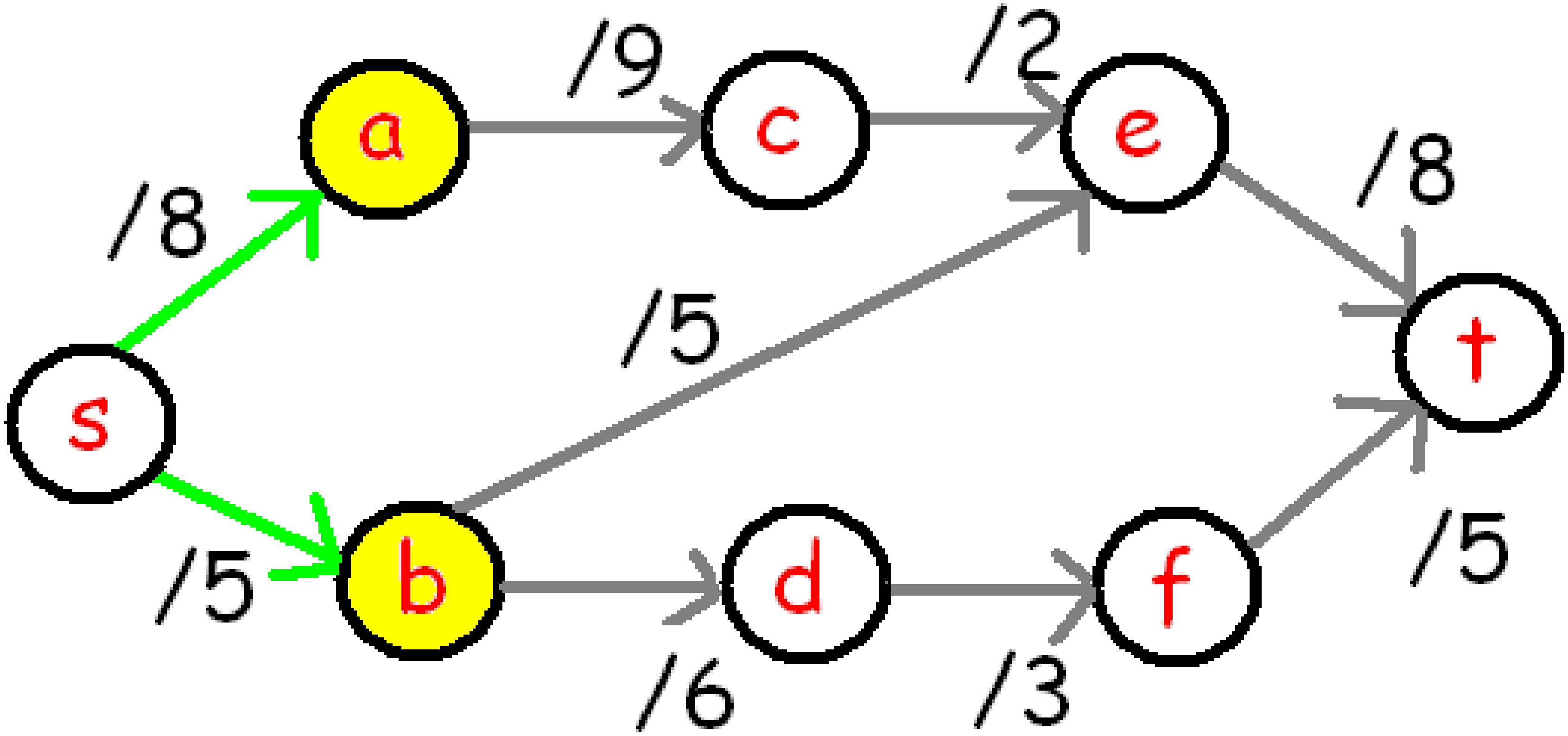


Jack Edmonds

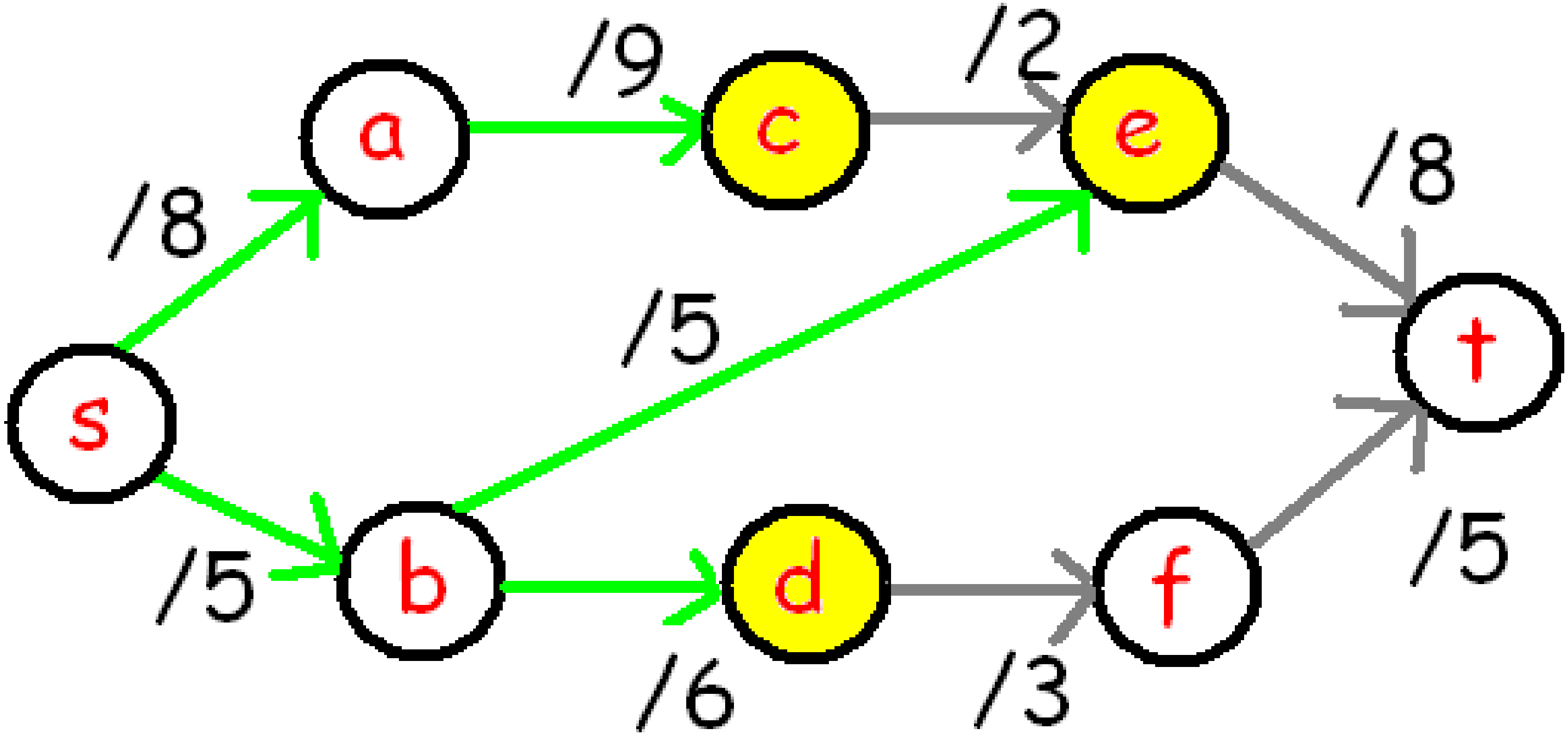
Richard Karp


/8
/9
/2
/8
/5
/5
/6
/3
/5
s
a
b
c
d
e
f
t

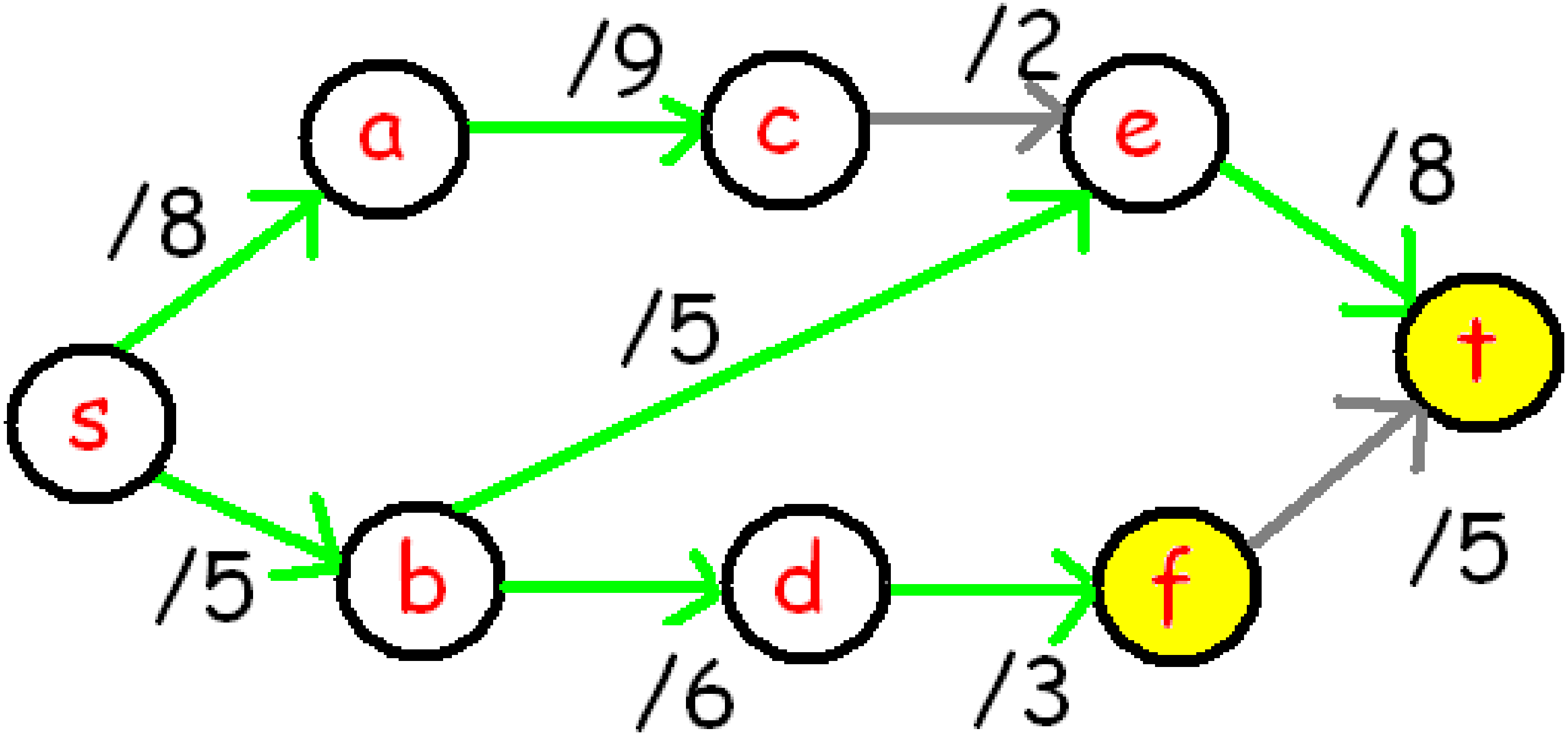

/8
/9
/2
/8
/5
/5
/6
/3
/5
a
c
e
s
t
b
d
f

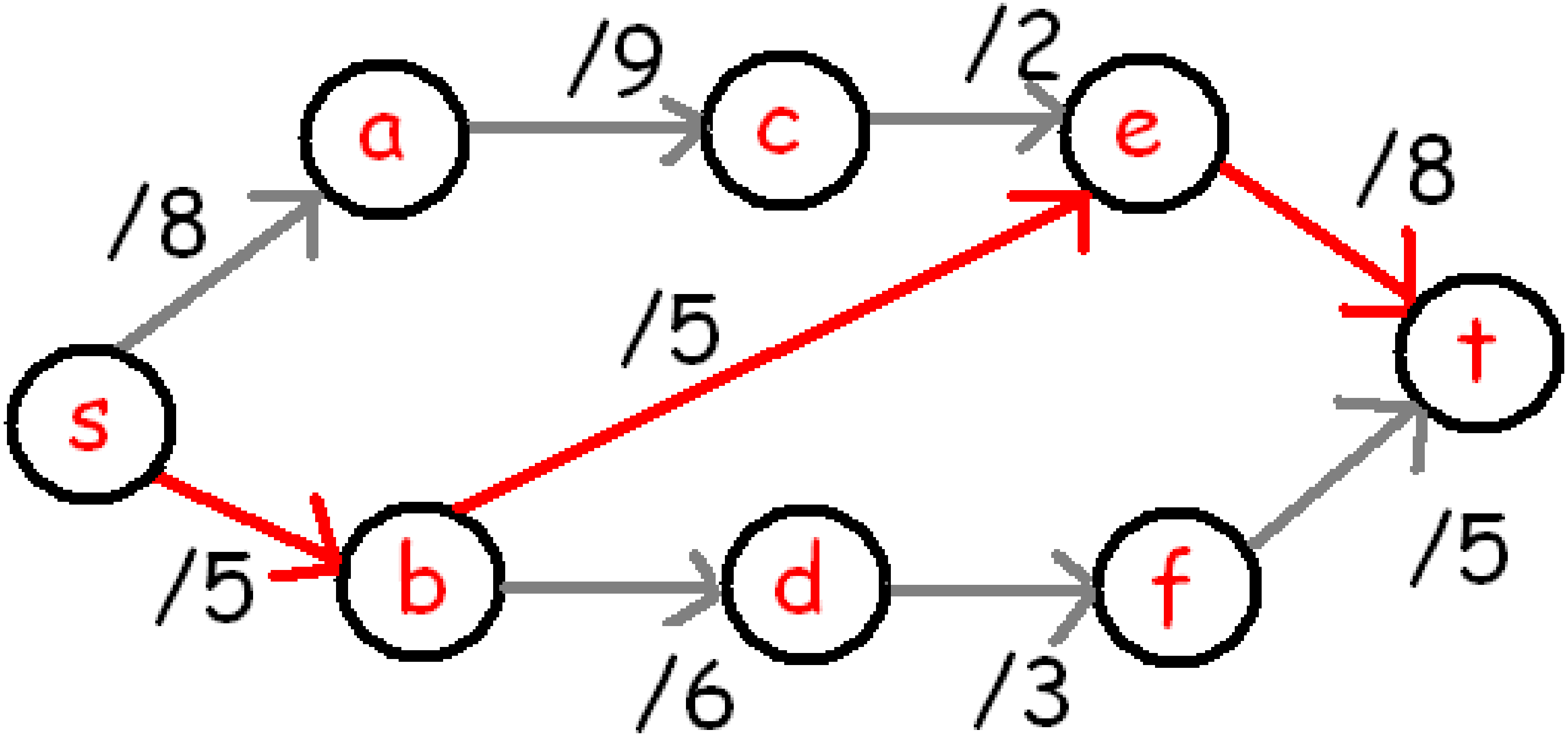


a
c
e
s
b
d
f
t
/8
/9
/2
/5
/8
/5
/6
/3
/5

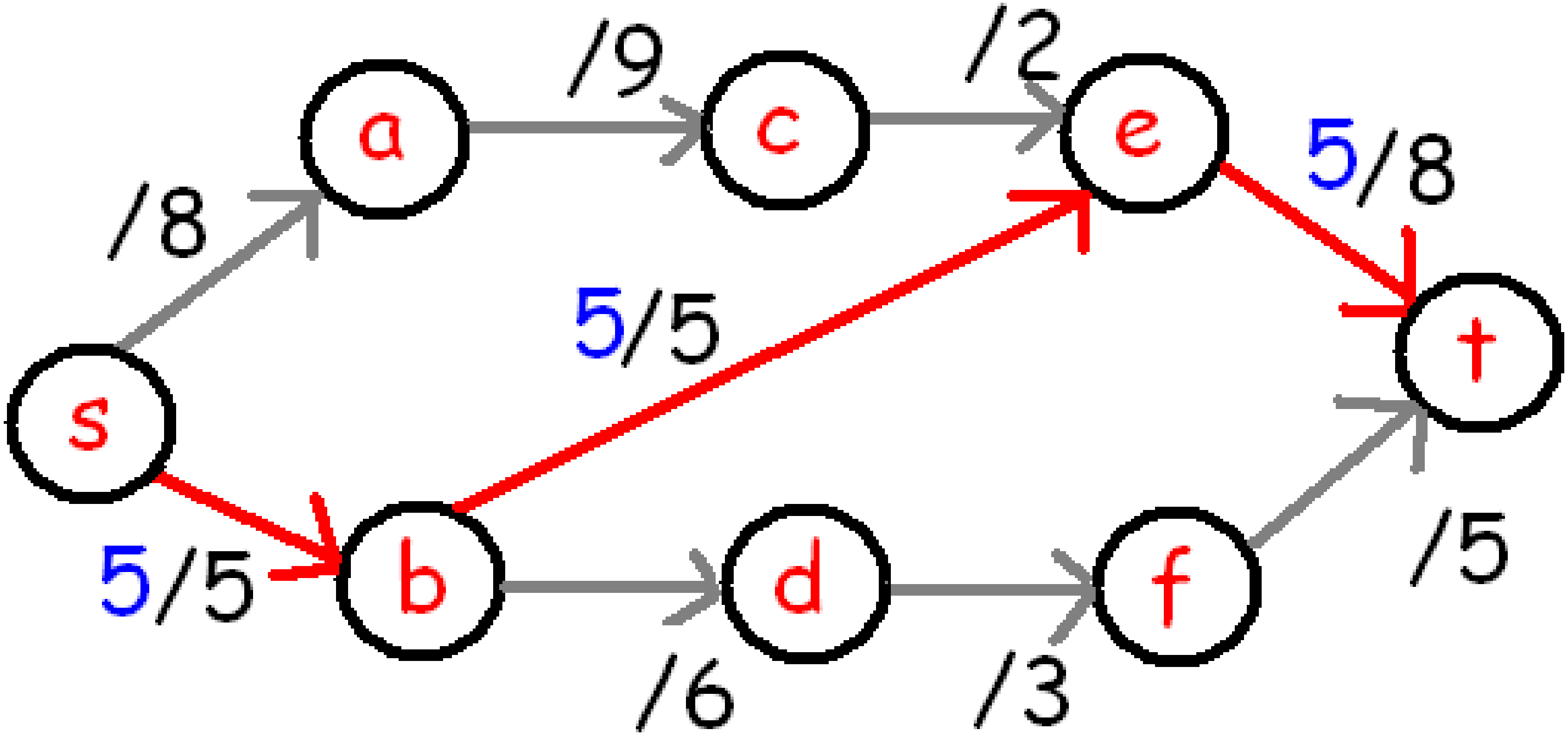


/8
/9
/2
/8
/5
/5
/6
/3
/5
a
c
e
s
t
b
d
f

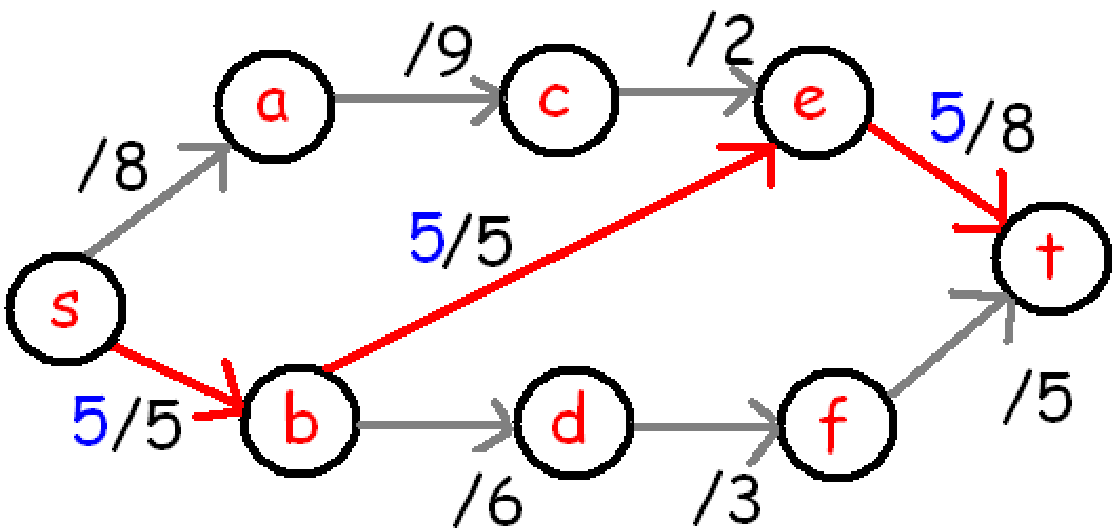

/9
/2
/8
/8
/5
s
a
c
e
t
b
d
f
/5
/5
/6
/3



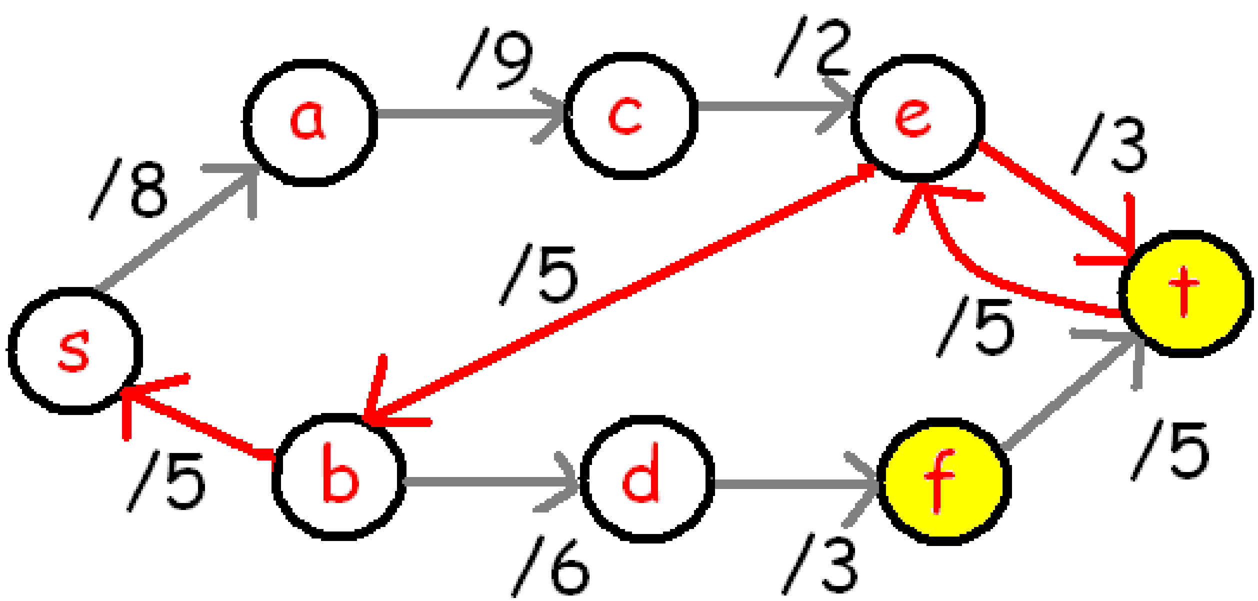
Augmenting path: s, b, e, t

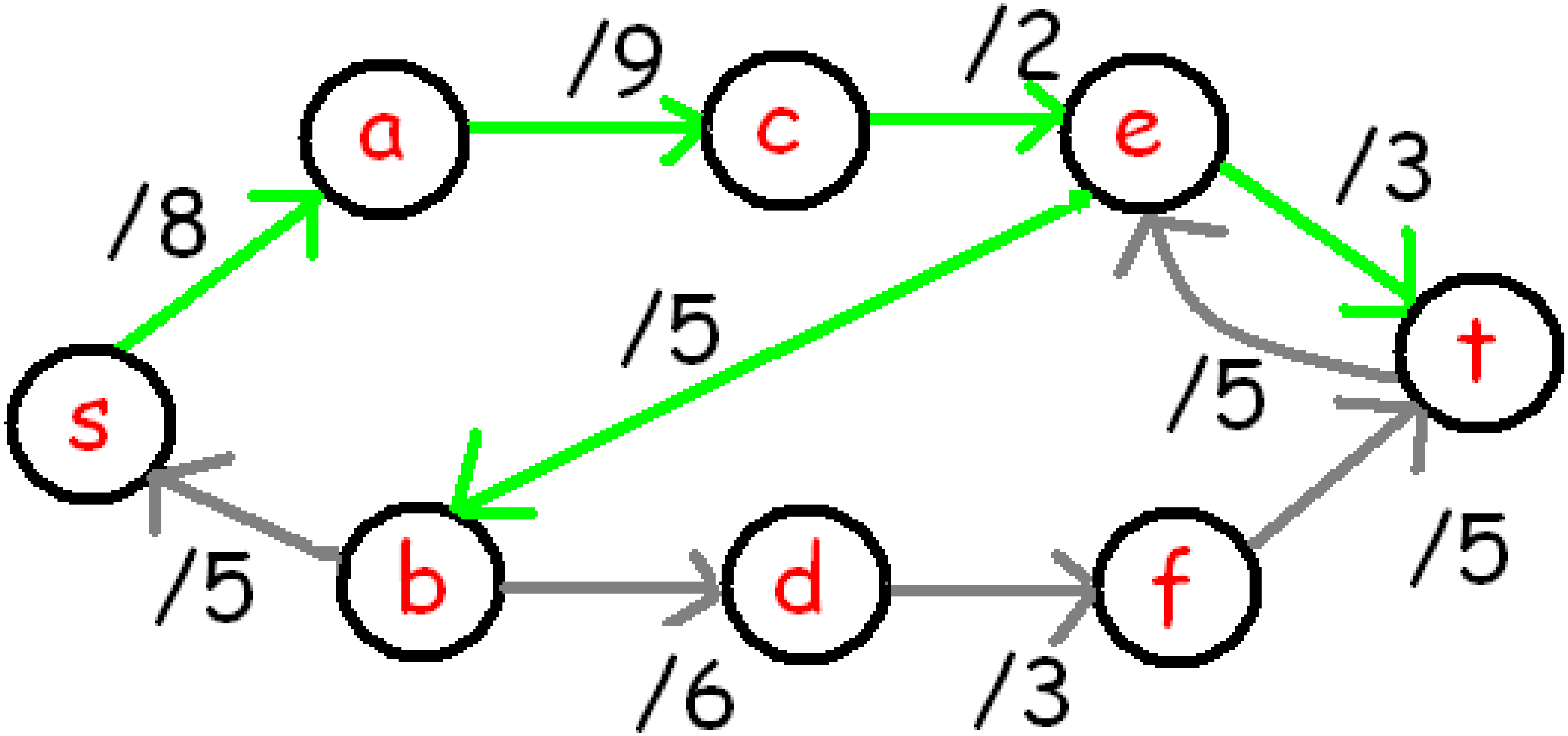


Send 5 units of flow along augmenting path.

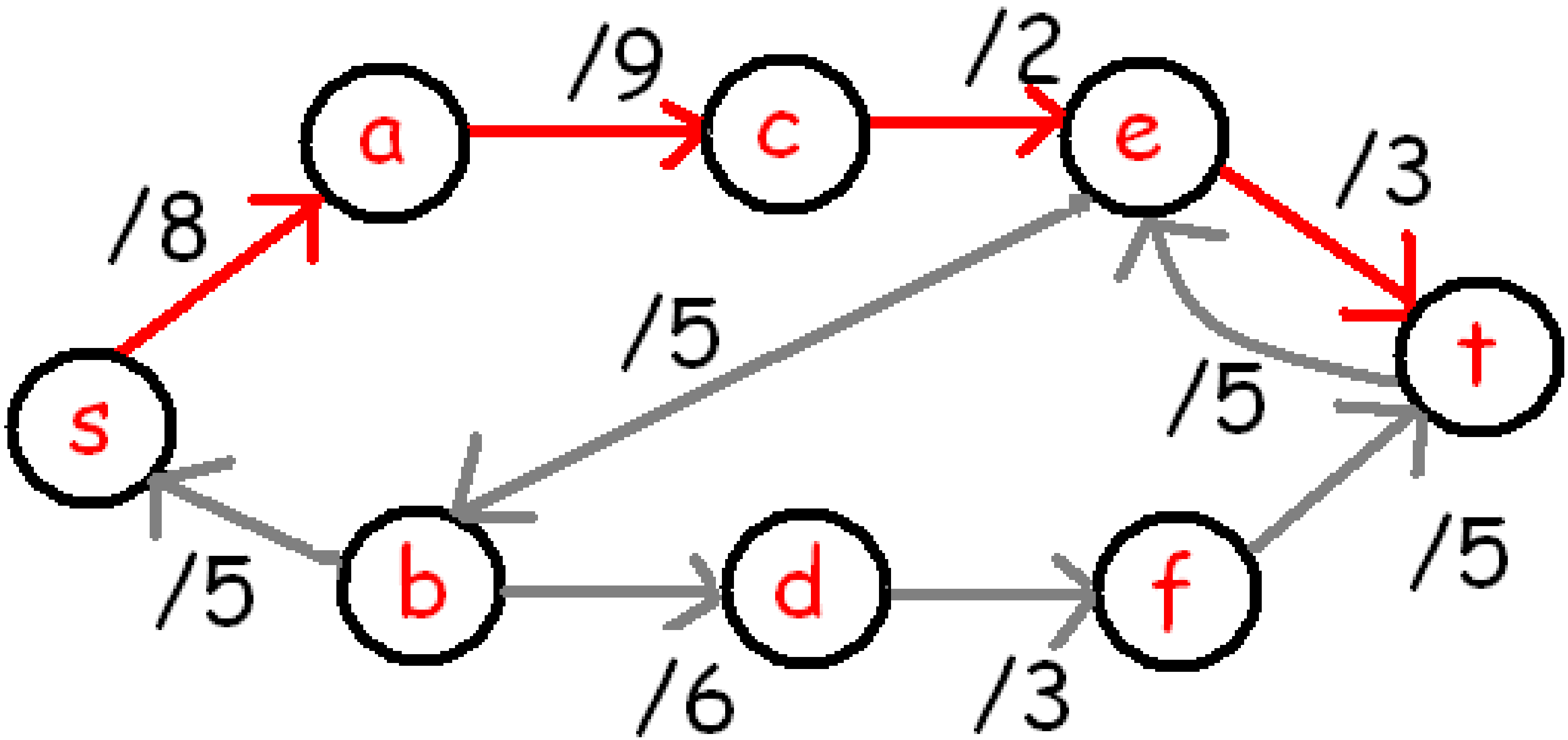


Create new auxillary graph:



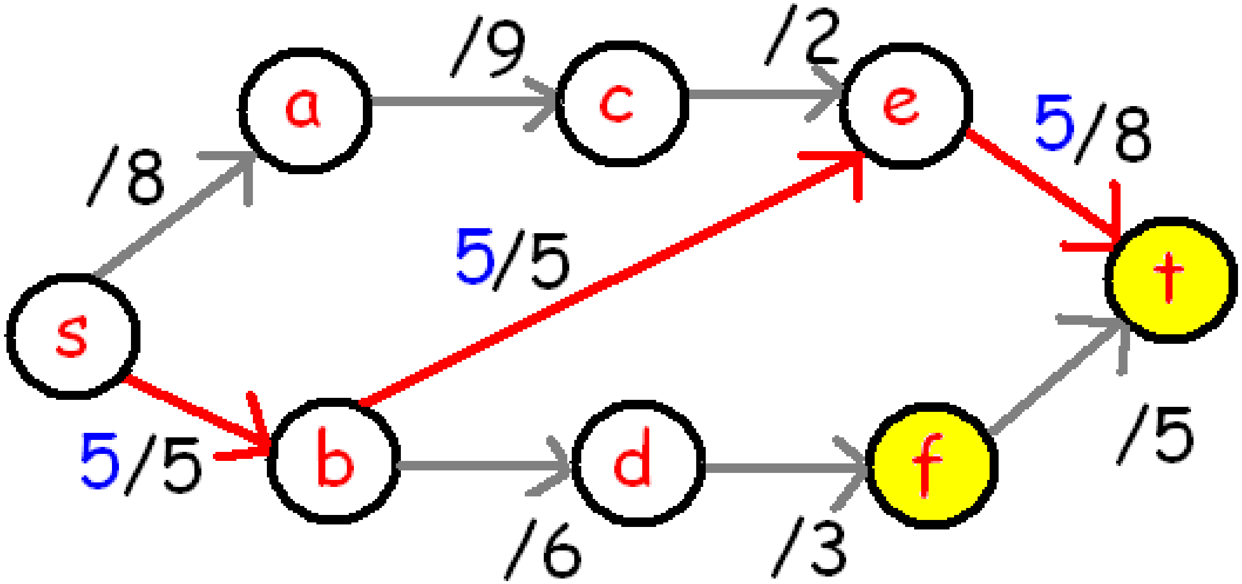


Do BFS starting at s of auxillary graph.

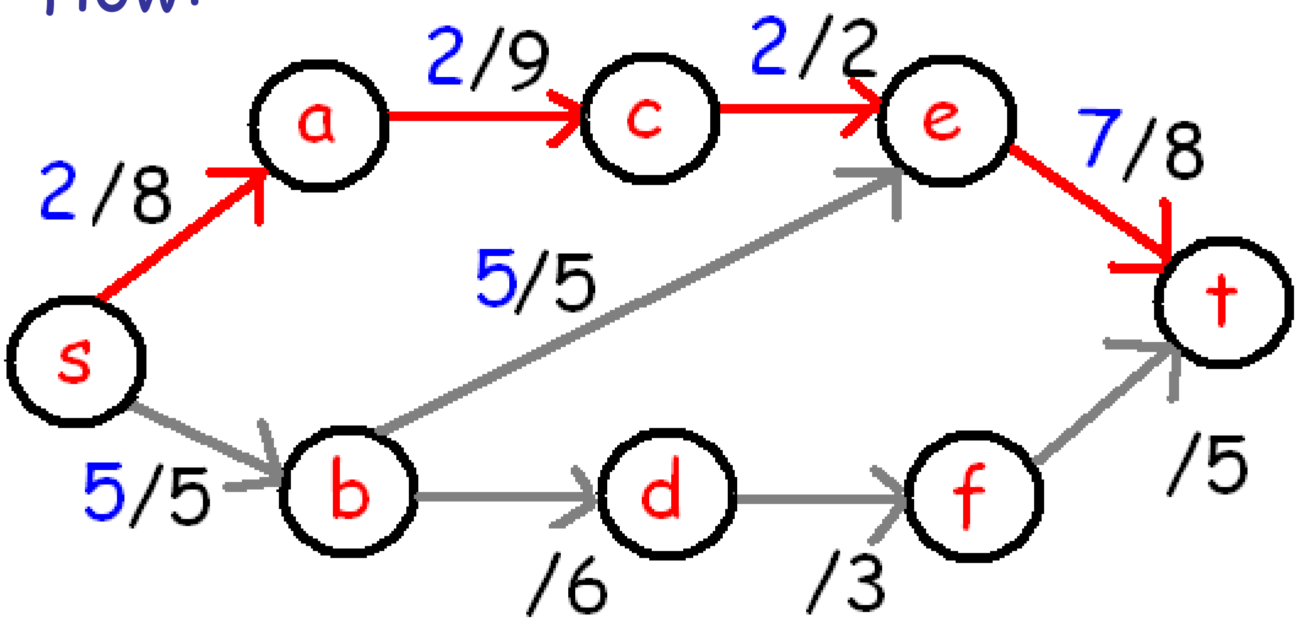


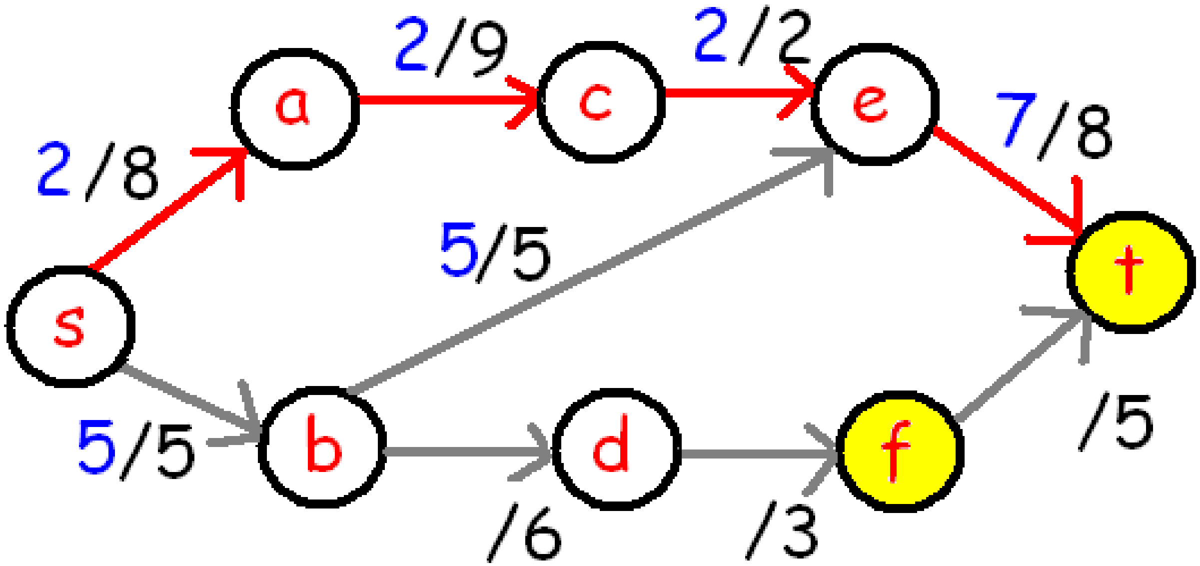
Identify augmenting path: s, a, c, e, t

Capacity of path is 2.

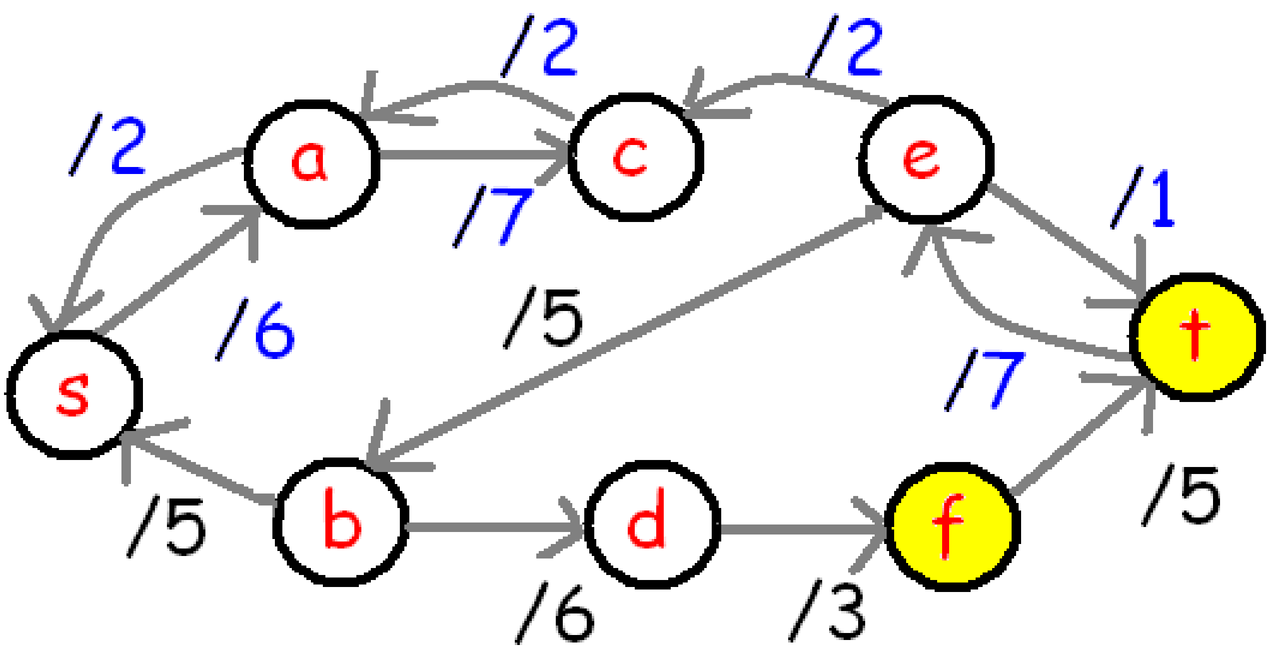


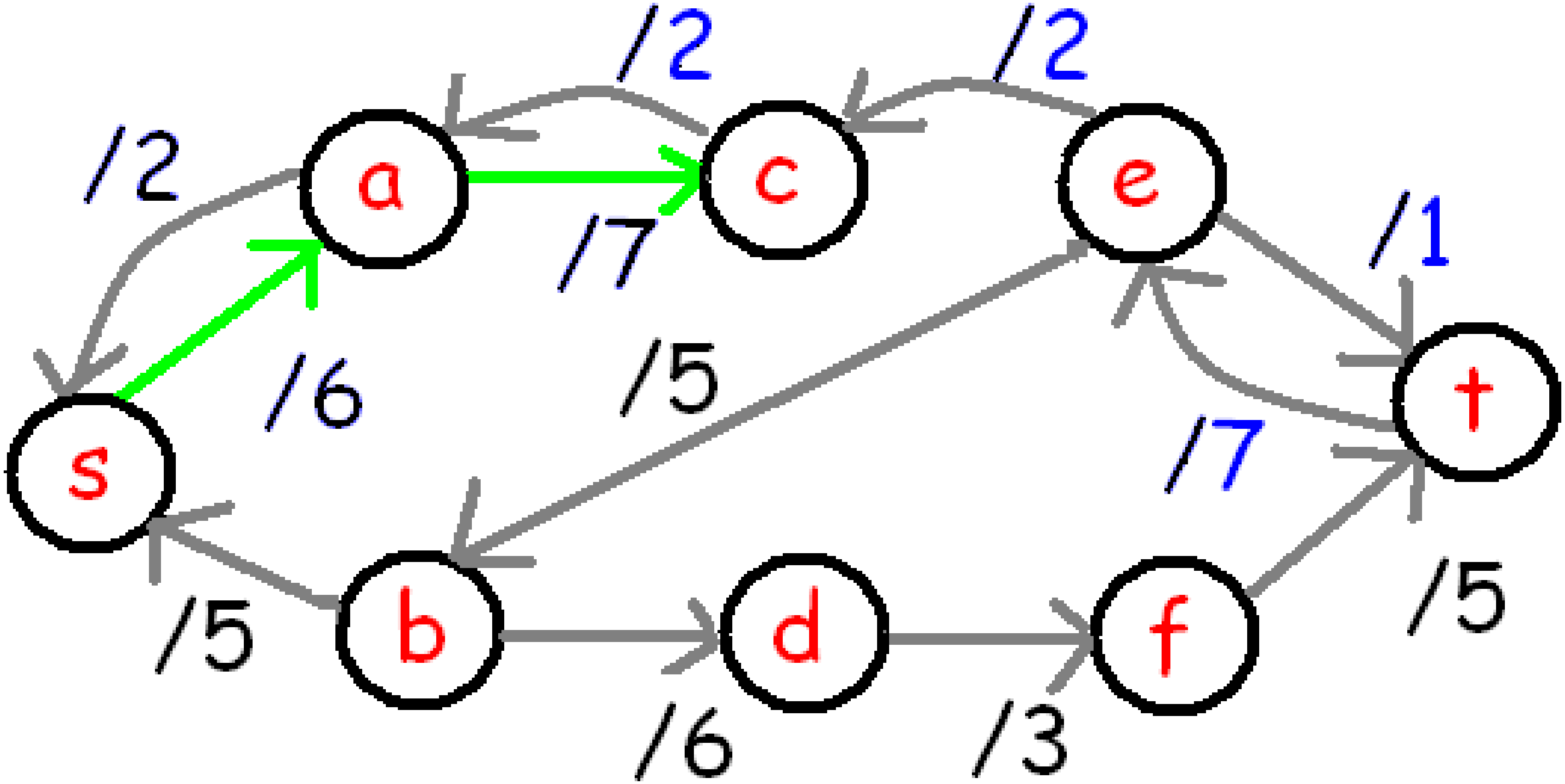
Augment flow:



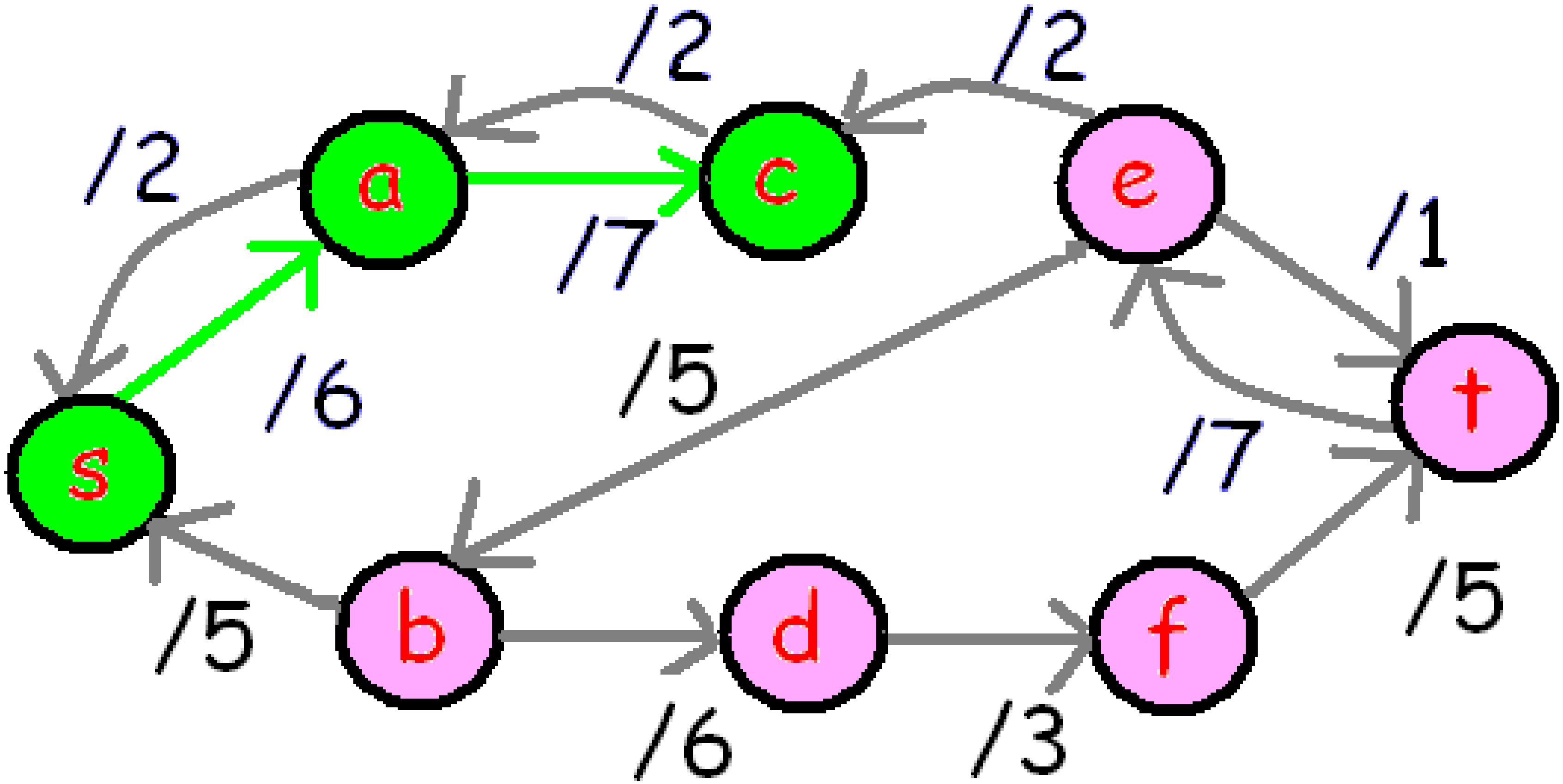


Make new auxillary graph:



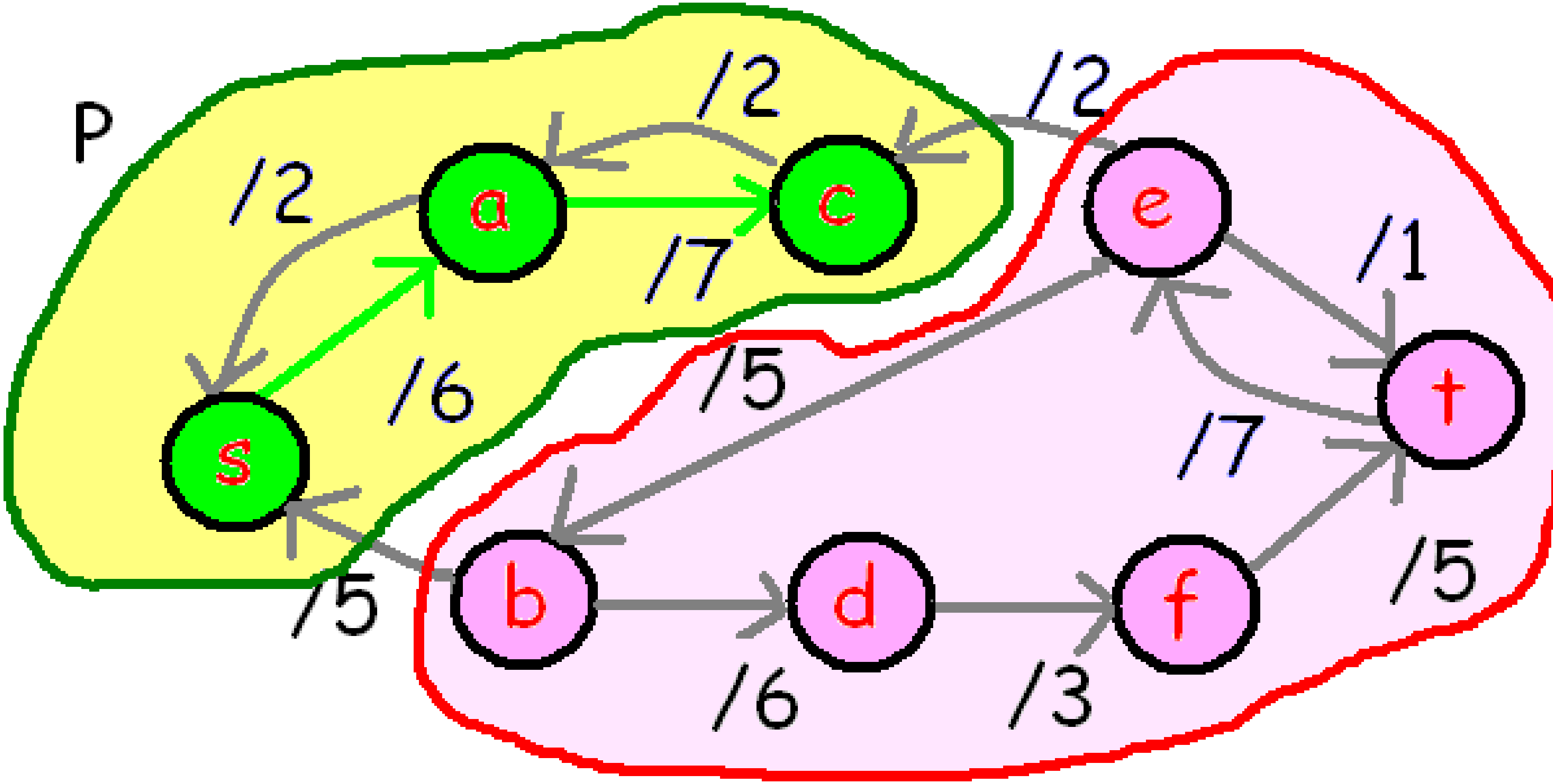


Apply BFS: Cannot reach t.



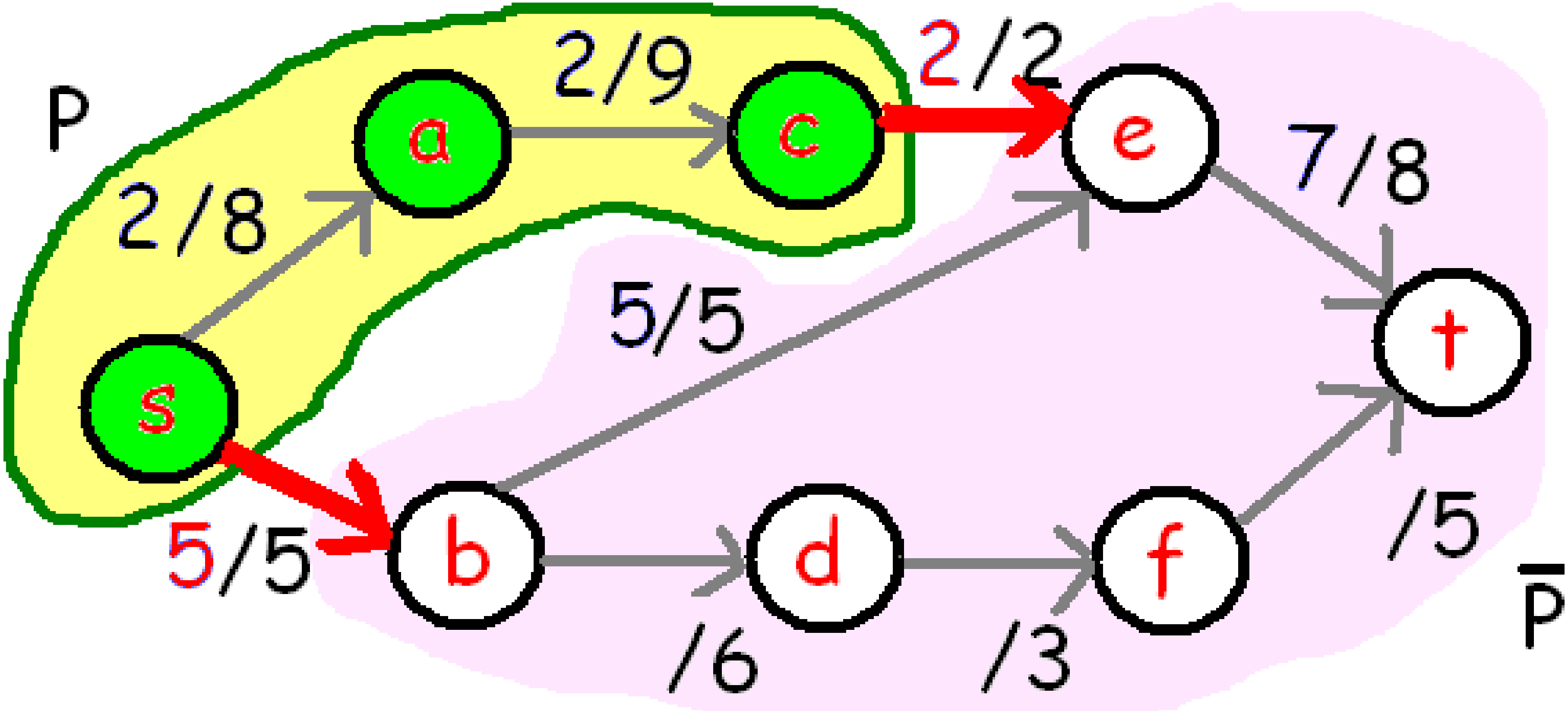
P= {u: u is reachable from s on BFS}

(P, V-P)= { (u,v) : u $\in$ P and v $\notin$ P}.



In the auxillary:

P={s, a, c}   V-P = {b, d, e, f, t}

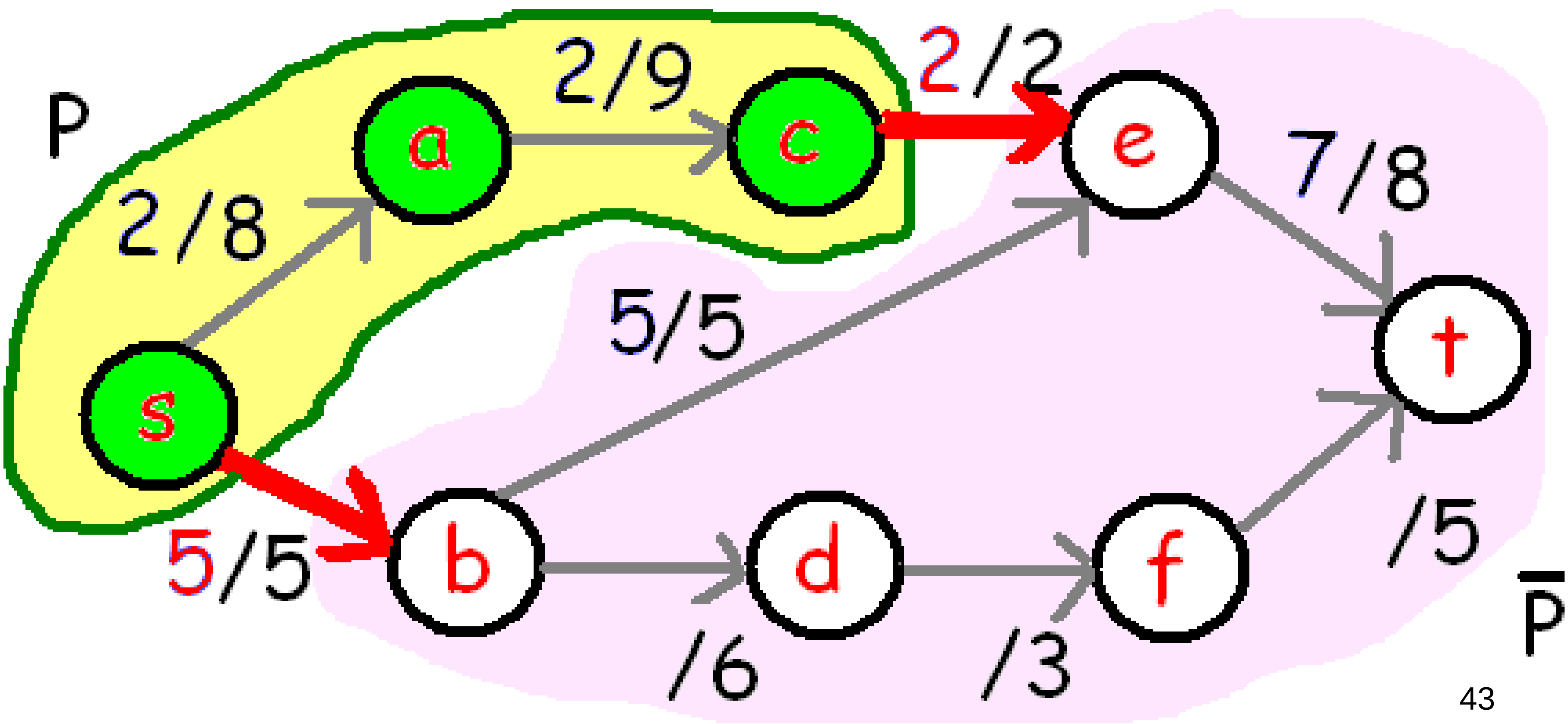


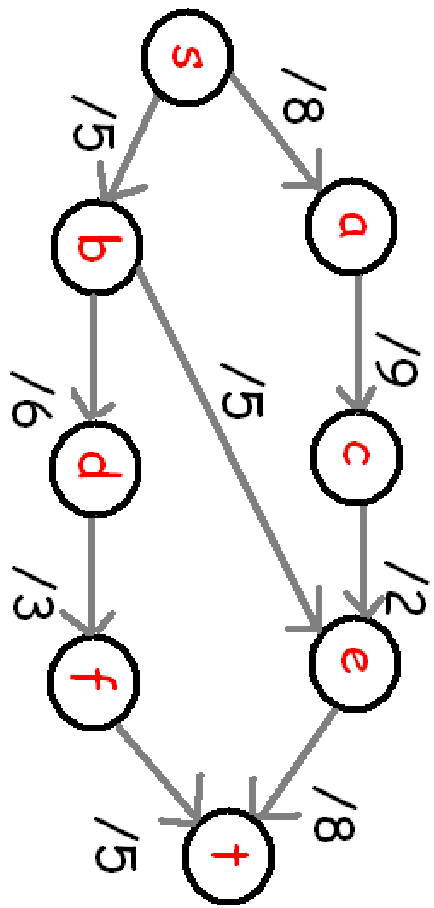
$(P, V\text{-}P) = \{ (u,v) : u \in P \text{ and } v \notin P \}$.

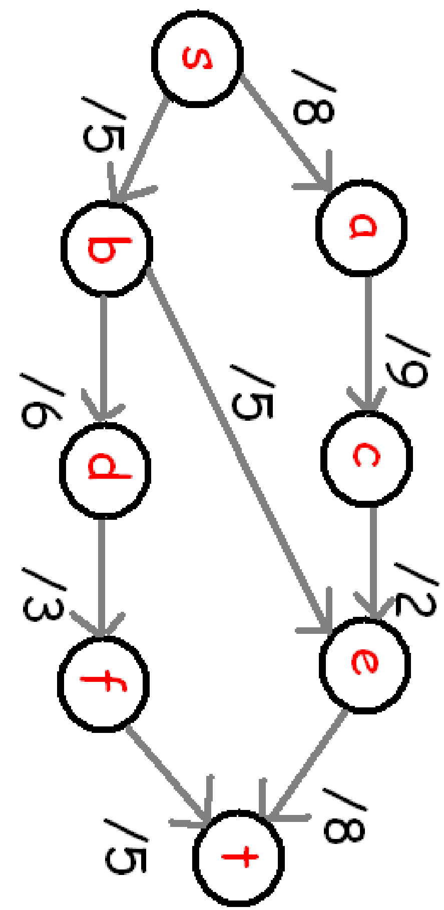
So $(P, V\text{-}P) = \{ (s, b), (c, e) \}$

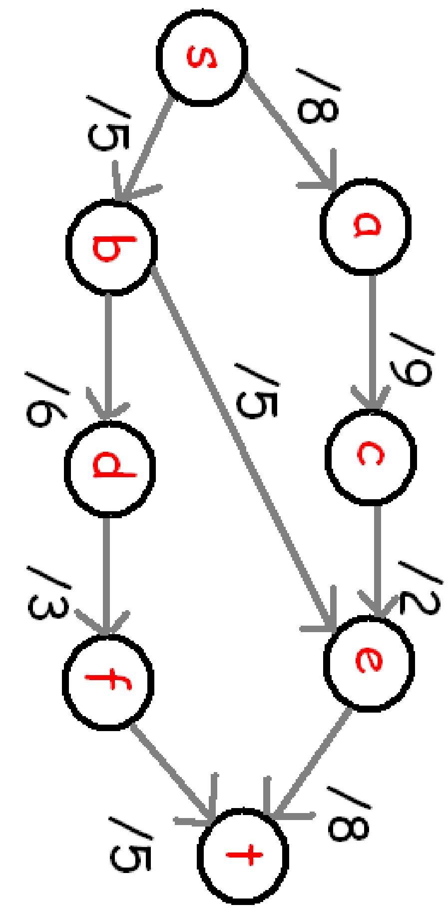
Max Flow Min Cut Theorem:

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




s
a
b
c
d
e
f
t
/8
/5
/9
/6
/5
/2
/3
/8
/5


s
a
b
c
d
e
f
t
/8
/5
/9
/6
/5
/2
/3
/8
/5


s
a
b
c
d
e
f
t
/8
/5
/9
/6
/5
/2
/3
/8
/5