Faculty of Information Technology, Monash University

FIT2004: Algorithms and Data Structures

Week 11: Network Flow

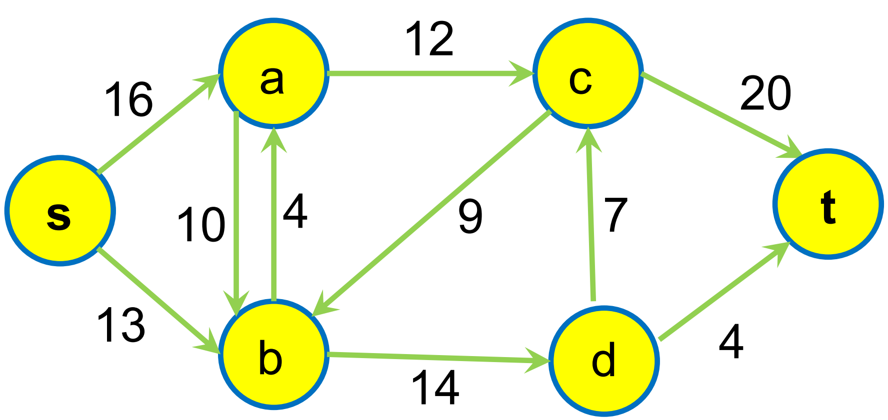
These slides are prepared by M. A. Cheema and are based on the material developed by Arun Konagurthu and Lloyd Allison.

Outline

1. Maximum Flow Problem
2. Ford-Fulkerson Algorithm
3. Min-cut Max-flow Theorem

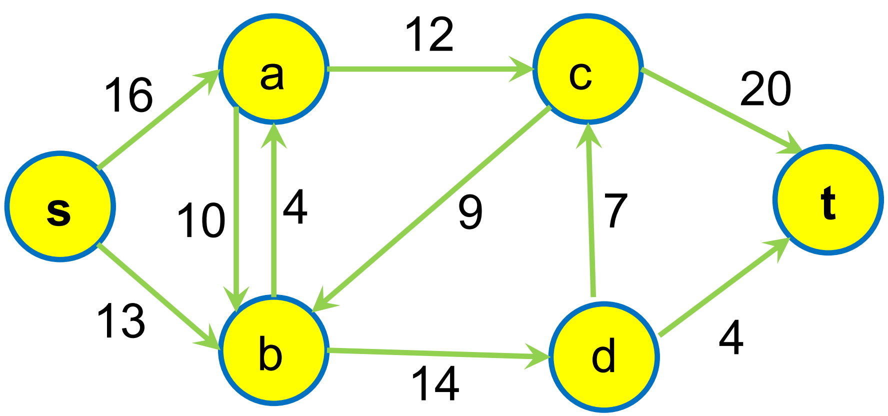
Flow Networks

- A flow network is a connected directed graph where
  - there is a single source vertex and a single sink/destination vertex;
  - each edge has a given (non-negative) capacity (usually integers)
    - giving the maximum amount/rate of flow that the edge can carry;
- Flow networks model many real-world problems
  - Water flowing through an assembly of pipes.
  - Electric current flowing through electrical circuits.
  - Information flowing through communication networks
  - Can be applied to many scenarios (unrelated to physical flows).



Some basic notations

- Set of all incoming edges to a vertex v: denoted as $E_{in}(v)$
  - $E_{in}(b) = s \rightarrow b, c \rightarrow b, a \rightarrow b$
  - $E_{in}(a) = ?$
- Set of all outgoing edges from a vertex v: denoted as $E_{out}(v)$
  - $E_{out}(b) = b \rightarrow a, b \rightarrow d$
  - $E_{out}(a) = ?$
- Source Vertex: denoted as s (has no incoming edges)
- Sink/target vertex: denoted as t (has no outgoing edges)

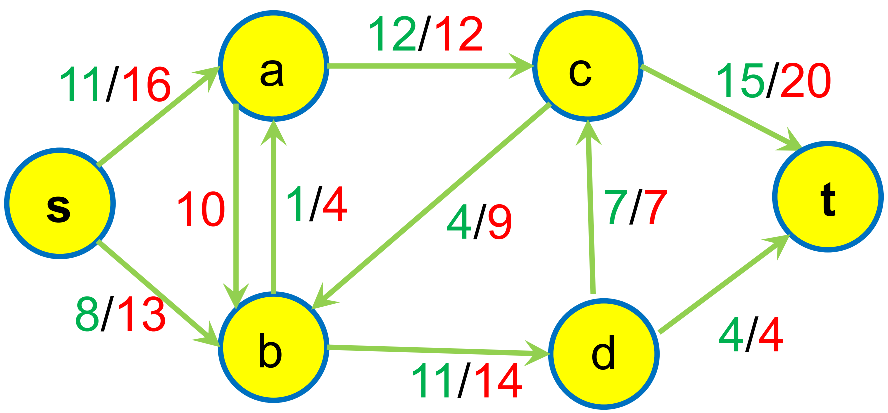


Flow

- Flow is an **assignment** of how much material is flowing through each edge in the flow network given its stated edge capacity.
- All vertices (except source and sink) conserve their flow. That is
  - The total amount flowing into any vertex (through incoming edges)

    IS EQUAL TO

    the total amount flowing out of that vertex (through outgoing edges).
  - I.e., Total incoming flow at a vertex = total outgoing flow at a vertex
  - This key property is called flow conservation.



Green numbers indicate flow and red indicate capacity. Flow is not shown if 0

Properties of a Flow Network

A flow network must satisfy the following two properties.

Property 1: Capacity Constraint

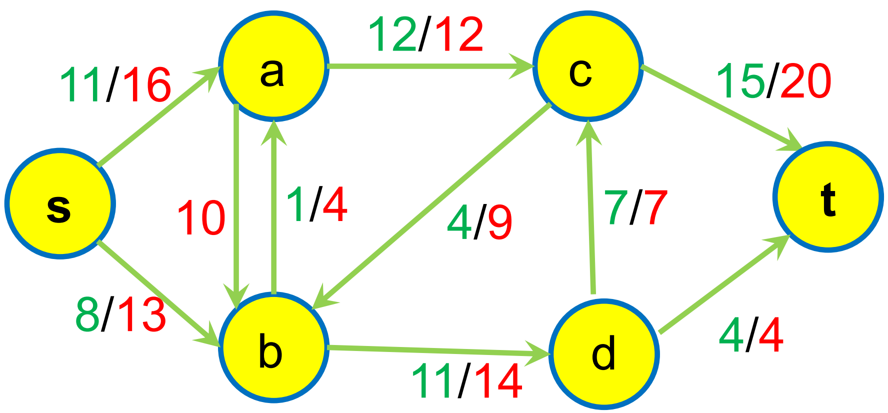
- For each edge e, its flow, denoted as f(e), is bounded by the capacity of its edge, i.e., $0 \leq f(e) \leq c(e)$ where c(e) is the capacity of the edge

Property 2: Flow Conservation

- For any vertex v (except source and sink), the total flow coming into the vertex must be equal to the total flow going out from this vertex – formally

$$\sum_{\forall e_{in} \in E_{in}(v)} f(e_{in}) = \sum_{\forall e_{out} \in E_{out}(v)} f(e_{out}).$$

- What is total outgoing flow of b?
- What is total incoming flow of b?



Green numbers indicate flow and red indicate capacity. Flow is not shown if 0

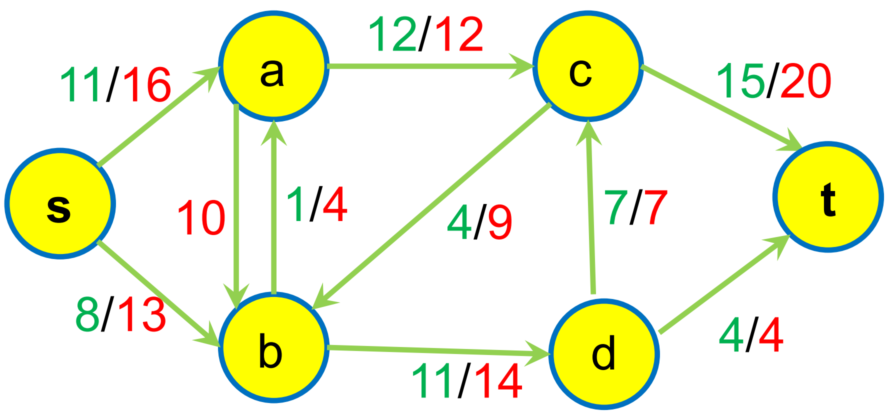
Maximum-flow Problem

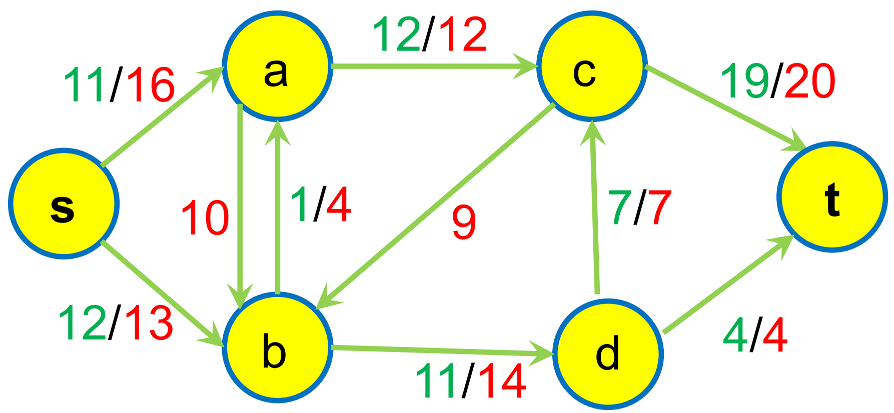
Value of a flow in a network:

- Given that flow network satisfies the capacity constraint and flow conservation properties, flow of a network is the total flow out of the source vertex. Equivalently, this is the same as the total flow into sink vertex.
  - What is the flow value in the flow network at bottom right?
  - What is the flow value in the flow network at bottom left?

Maximum-flow problem

- Given a flow network, what is the maximum value of the flow that can be sent from source s to sink t without violating the flow network properties.





Green numbers indicate flow and red indicate capacity. Flow is not shown if 0

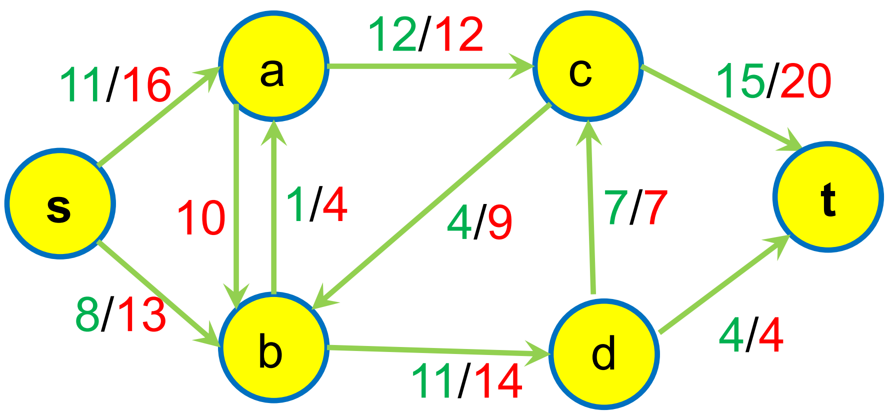
Outline

1. Maximum Flow Problem
2. Ford-Fulkerson Algorithm
3. Min-cut Max-flow Theorem

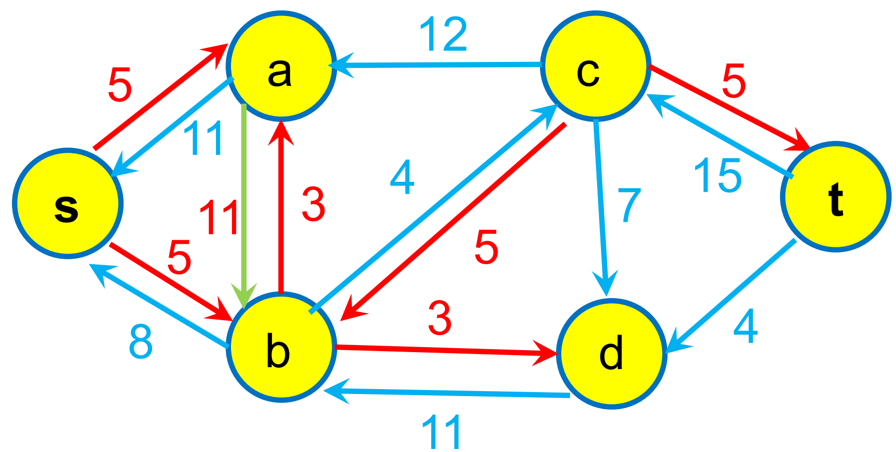
Residual Network

- Residual network has the same vertices as the original network.
- For every directed edge u→v in flow network, we add two edges in the residual network:
  - Forward edge/Residual edge: An edge in the same direction as u→v with the residual/remaining capacity
    - Indicates the remaining capacity (remaining amount of flow) that can be sent via the edge u→v
    - Edge is not created if remaining capacity is 0.
  - Backward edge/Reversible flow edge: An edge in the direction opposite to u→v (i.e., v→u) with weight equal to the current flow of u→v in the flow network
    - Indicates the flow that can be reversed/cancelled
    - Edge is not created if reversible flow is 0
  - Edges in the same direction are merged into a single edge with total weight shown

MARS: Weight of ?



Flow Network: Green numbers indicate flow and red indicate capacity. Flow is not shown if 0

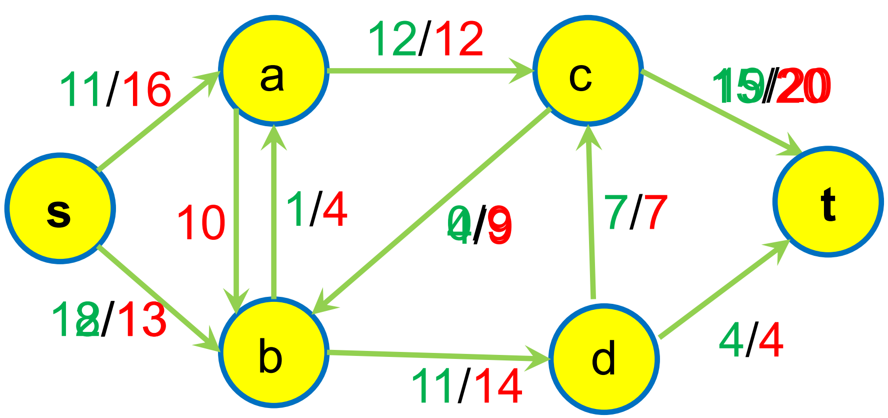


Residual Network: Where possible blue edges indicate reversible flow and red indicate residual capacity

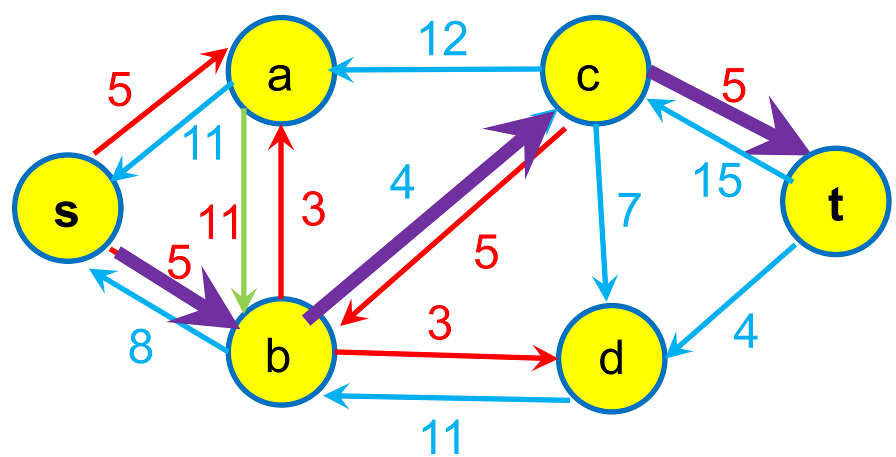
Augmenting Path in Residual Network

Augmenting path is any simple path (a path without repeating vertices) from source s to target t.

- E.g., s → b → c → t (shown in purple edges)
- Residual capacity of a path is the minimum edge weight on this path (e.g., 4 in the example)
- For each edge along this path, we can push additional flow equal to the "residual capacity of the path" in the flow network, e.g., 4 along each edge on s → b → c →t



Flow Network: Green numbers indicate flow and red indicate capacity. Flow is not shown if 0



Residual Network: Where possible blue edges indicate reversible flow and red indicate residual capacity

Ford-Fulkerson Method

Initialize flow f to zero

Create residual network

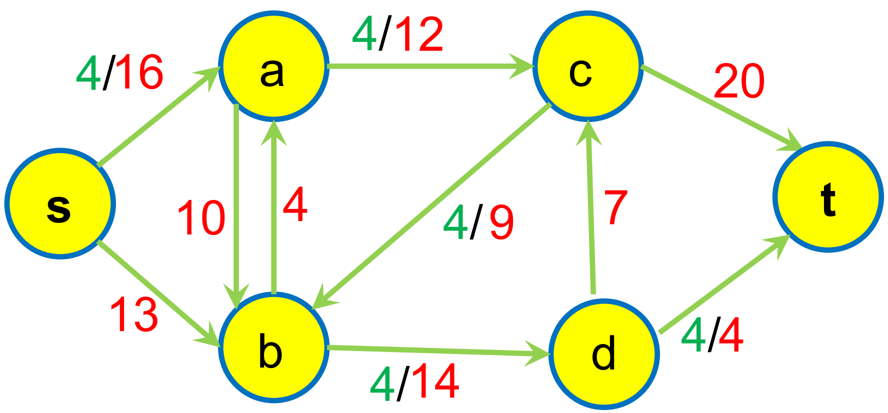
While there exists an augmenting path in the residual network:

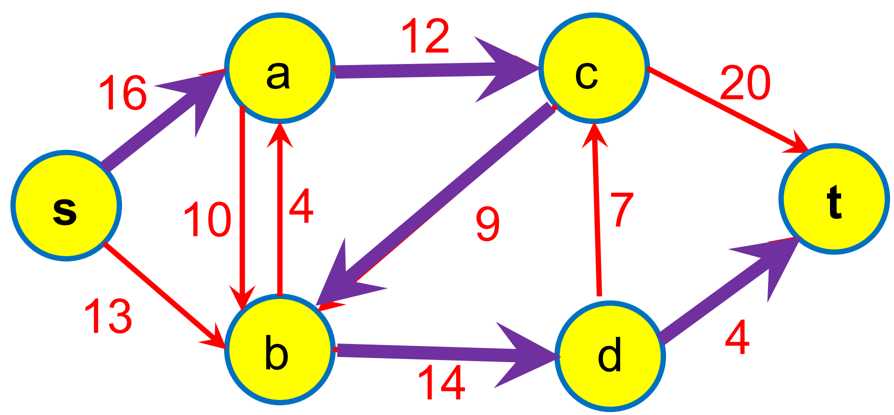
  choose any augmenting path P

  augment the flow equal to residual capacity of P in the flow network

  update residual network

return f





Flow Network: Green numbers indicate flow and red indicate capacity. Flow is not shown if 0

f = 4

Residual Network: Where possible blue edges indicate reversible flow and red indicate residual capacity

Ford-Fulkerson Method

Initialize flow f to zero

Create residual network

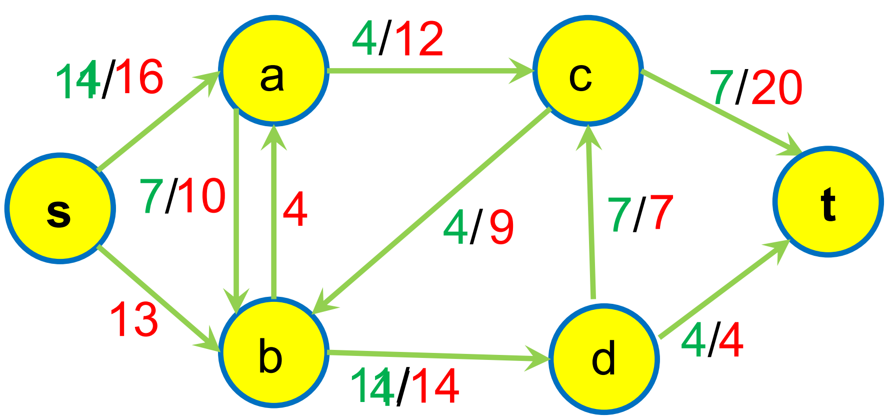
While there exists an augmenting path P in the residual network:

    choose any augmenting path P

    augment the flow equal to residual capacity of P in the flow network

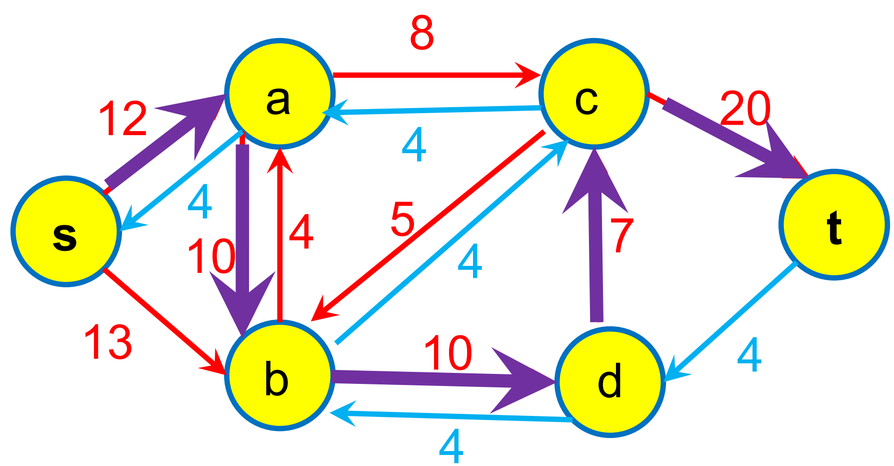
    update residual network

return f



Flow Network: Green numbers indicate flow and red indicate capacity. Flow is not shown if 0

f = 41



Residual Network: Where possible blue edges indicate reversible flow and red indicate residual capacity

Ford-Fulkerson Method

Initialize flow f to zero

Create residual network

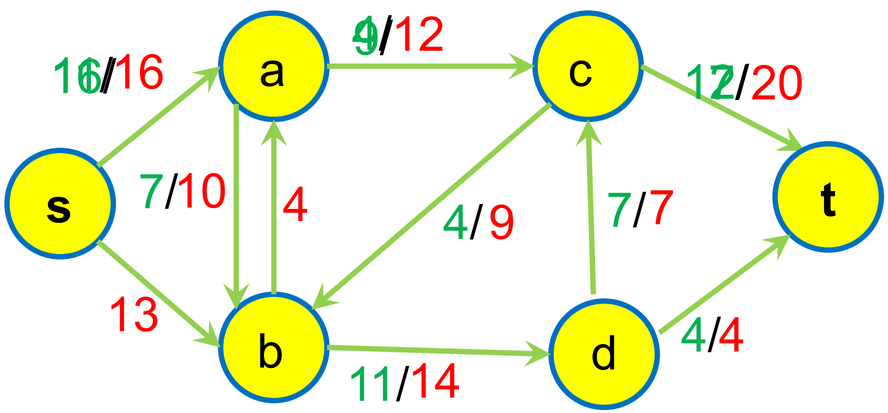
While there exists an augmenting path P in the residual network:

choose any augmenting path P

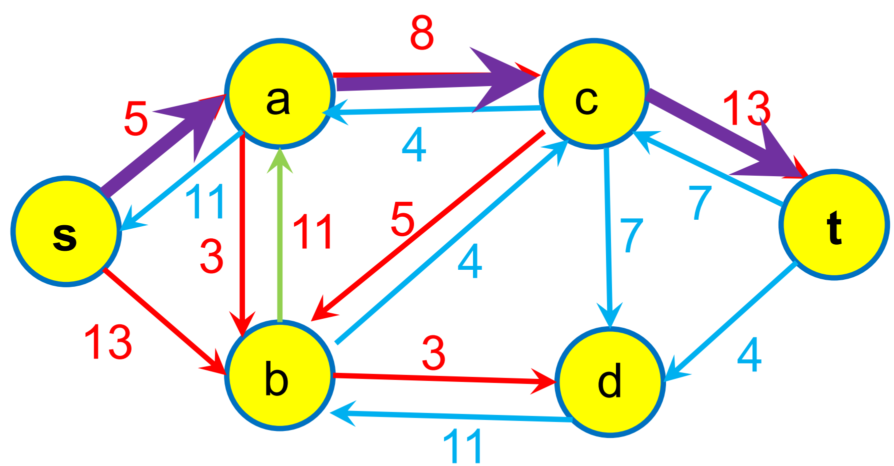
augment the flow equal to residual capacity of P in the flow network

update residual network

return f



Flow Network: Green numbers indicate flow and red indicate capacity. Flow is not shown if 0



f = 16

Residual Network: Where possible blue edges indicate reversible flow and red indicate residual capacity

Ford-Fulkerson Method

Initialize flow f to zero

Create residual network

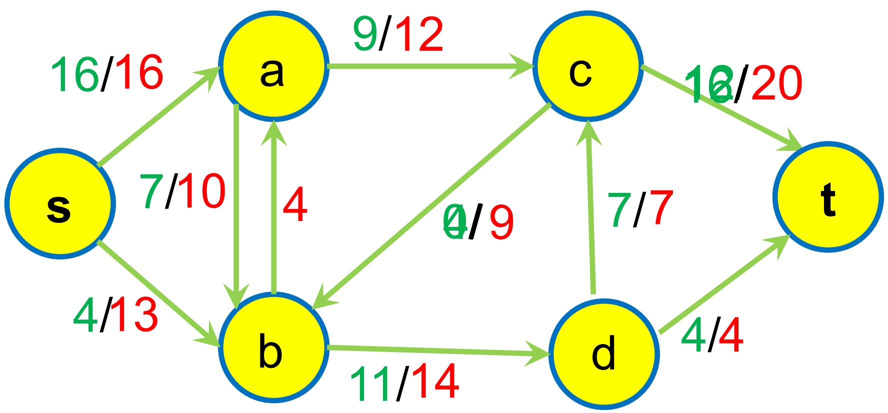
While there exists an augmenting path P in the residual network:

  choose any augmenting path P

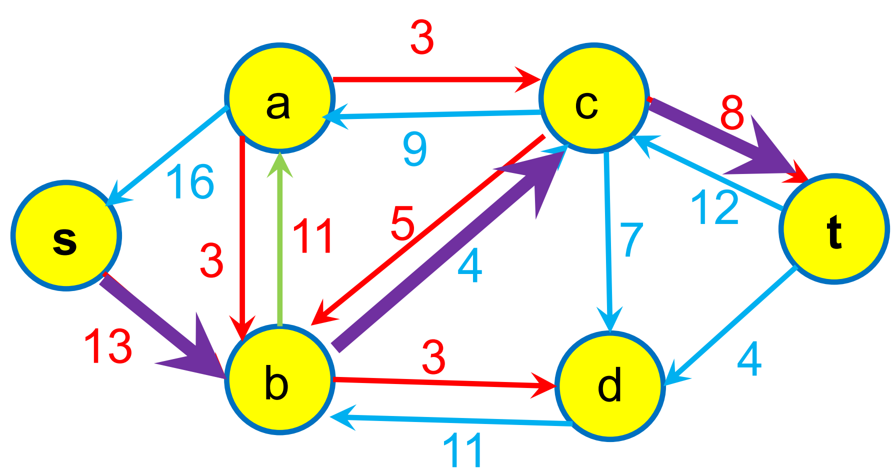
  augment the flow equal to residual capacity of P in the flow network

  update residual network

return f



Flow Network: Green numbers indicate flow and red indicate capacity. Flow is not shown if 0



f = 20

Residual Network: Where possible blue edges indicate reversible flow and red indicate residual capacity

Ford-Fulkerson Method

Initialize flow f to zero

Create residual network

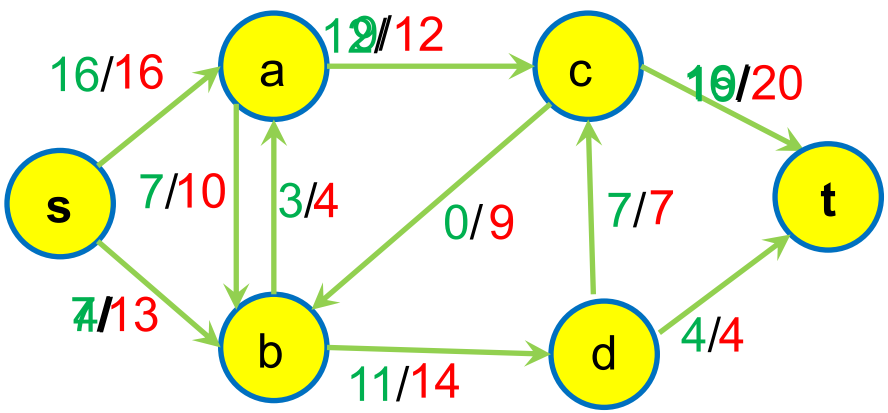
While there exists an augmenting path P in the residual network:

  choose any augmenting path P

  augment the flow equal to residual capacity of P in the flow network

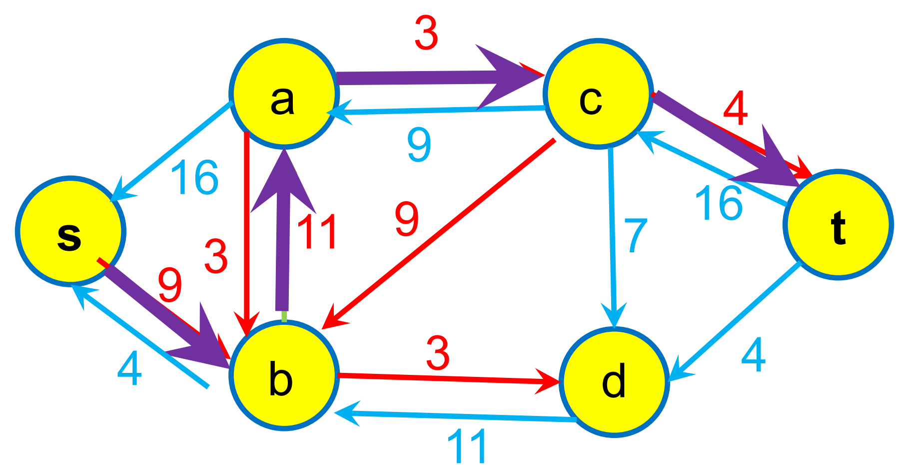
  update residual network

return f



Flow Network: Green numbers indicate flow and red indicate capacity. Flow is not shown if 0

f = 20



Residual Network: Where possible blue edges indicate reversible flow and red indicate residual capacity

Ford-Fulkerson Method

Initialize flow f to zero

Create residual network

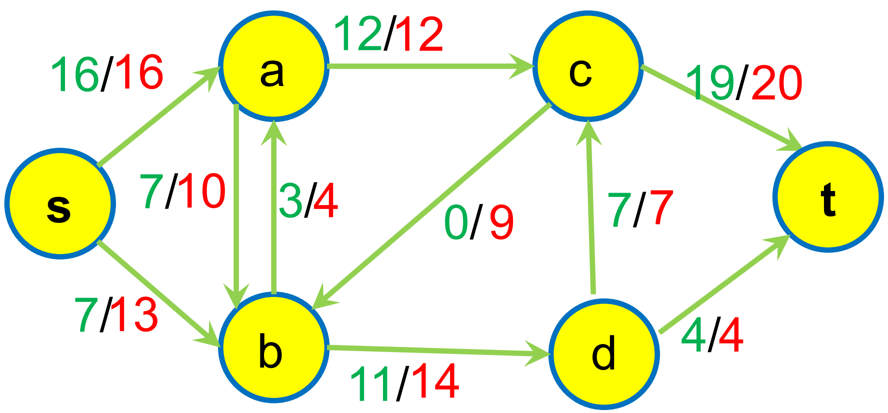
While there exists an augmenting path P in the residual network:

choose any augmenting path P

augment the flow equal to residual capacity of P in the flow network

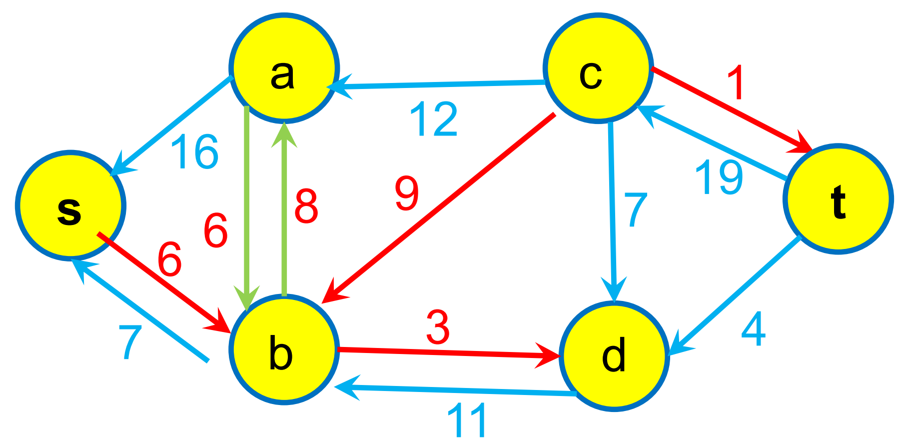
update residual network

return f



Flow Network: Green numbers indicate flow and red indicate capacity. Flow is not shown if 0

f = 23



Residual Network: Where possible blue edges indicate reversible flow and red indicate residual capacity

Complexity Analysis

Initialize flow f to zero
Create residual network
While there exists an augmenting path P in the residual network:
    choose any augmenting path P
    augment the flow equal to residual capacity of P in the flow network
    update residual network
return f

- Number of edges in residual network
  - At most 2E → O(E)
- Total cost for a single iteration of the while loop?
  - Cost of finding an augmenting path?
    - O(V+E) assuming we are using BFS to find the path (assuming all edges are unweighted)
  - Cost of augmenting the flow in network
    - O(V+E) – there are at most V-1 edges in a path; finding/updating these edges in adjacency lists takes at most O(V+E)
  - Cost of updating residual network
    - O(V+E) – same as above
  - Total Cost for a single iteration: O(V+E) or O(E) because the graph is connected and O(E) ≥ O(V)
- Let F be the maximum flow of the network. What is the maximum number of iterations assuming all edge weights are integers?
  - O(F).
- Total cost of the algorithm assuming integer capacities: O(EF)

NON EXAMINABLE

- The above time complexity is pseudo-polynomial because F is an integer which can be arbitrarily large.
- It can be proved that the complexity is $O(VE^2)$ when BFS is used for finding augmenting path. This complexity is polynomial.

Proof of Correctness

- Does the algorithm terminate?
  - Yes (assuming all capacities are integers), because
  - the flow always increases by at least 1 and the algorithm terminates when flow is equal to the maximum flow
- When the algorithm terminates (i.e., there is no augmenting path in residual network), the flow of the network is the maximum flow.
  - We will need to understand “min-cut and max-flow” theorem

Outline

1. Maximum Flow Problem
2. Ford-Fulkerson Algorithm
3. Min-cut Max-flow Theorem

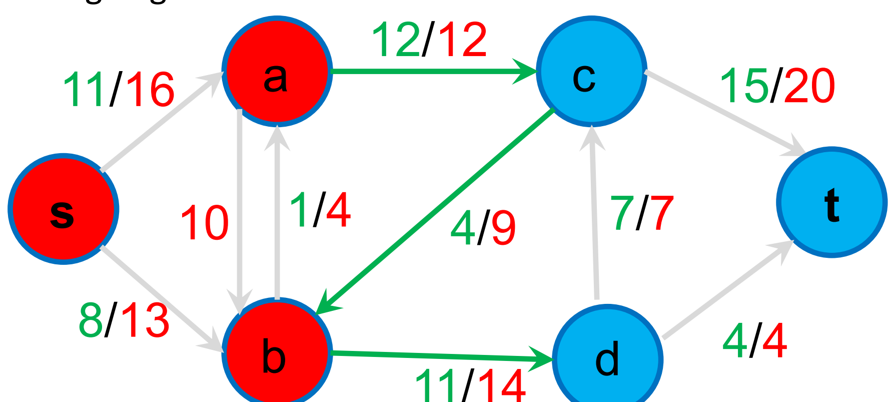
Flow and capacity of a cut

- A cut (S,T) of a flow network partitions the vertices of the network into two disjoint partitions S and T such that source s is in S and target t is in T.
  - E.g., S = {s,a,b} and T = {t, c, d}
- Cut-set of a cut (S,T) is the set of edges that "cross" the cut, i.e., each edge connects one vertex in S with another in T.
  - E.g., the cut-set for the example is a→c, b→d, c→b (green edges)
  - The edges that have direction from a vertex in S to a vertex in T are called outgoing edges of the cut.
    - E.g., a→c and b→d are the outgoing edges of the cut
  - The edges that have direction from a vertex in T to a vertex in S are called incoming edges of the cut.
    - E.g., c→b is an incoming edge of the cut.
- Capacity of a cut (S,T) is the total capacity of its outgoing edges
  - E.g., capacity of the cut in the example is 12 + 14 = 26
- Flow of a cut (S,T) is
  - Total flow of its outgoing edges – total flow of its incoming edges
  - E.g., flow in the example is 12 + 11 – 4 = 19

Is it true that flow of a cut is always less than or equal to the capacity of the cut?

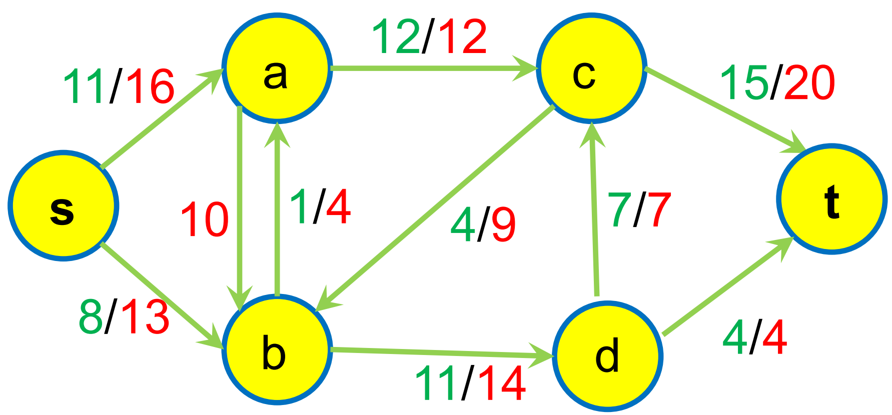
Yes, because

- Flow of an edge ≤ capacity of an edge
- Capacity of a cut does not subtract capacities for incoming edges



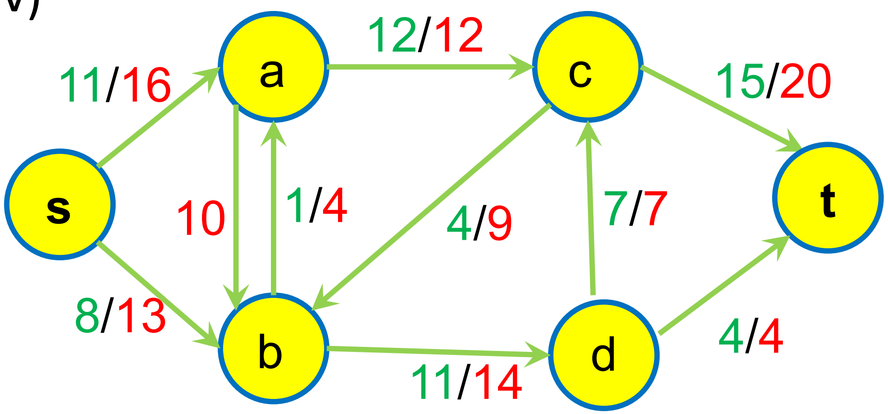
Flow and capacity of a cut

- Capacity of a cut (S,T): the total capacity of its outgoing edges
- Flow of a cut (S,T): Total flow of its outgoing edges – total flow of its incoming edges
- Assume S = {s, a,b,c,d} and T = {t}.
    - What is the capacity of this cut?
    - What is the flow of this cut?
- Assume S = {s, a, b,d} and T = {c,t}.
    - What is the capacity of this cut?
    - What is the flow of this cut?
- Assume S = {s, a} and T = {b,c,d,t}.
    - What is the capacity of this cut?
    - What is the flow of this cut?
- What is the flow value of this network?
- Note: flow of all of the above cuts is 19
    - which is the same as flow of the network.
- I.e., flow of **every** cut = flow of the network
- Let's prove this formally



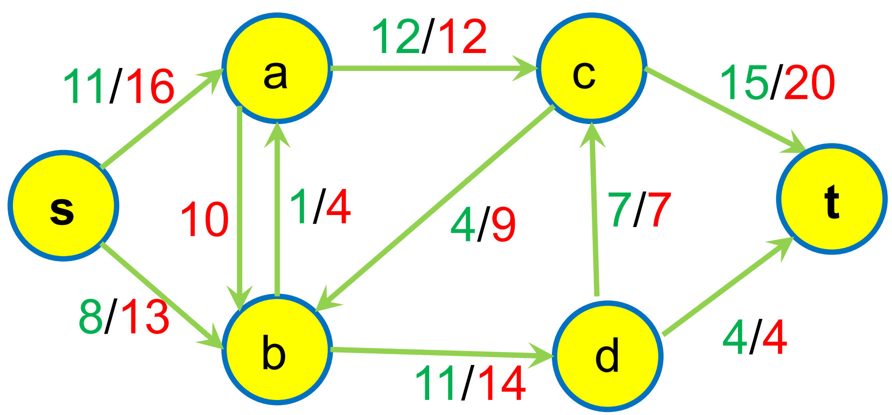
Flow of a cut = Flow of the network

- Let $F^{out}(v)$ be the total flow going out of a vertex and $F^{in}(v)$ be the total flow coming in the vertex
- Recall that flow of a network is the total flow going out from the source s.
  - Flow of the network = $F^{out}(s)$
- Flow conservation property: $F^{out}(v) - F^{in}(v) = 0$ for every vertex except s and t
- Flow of the network = $F^{out}(s)$
- Flow of the network = $F^{out}(s) + \sum_{v \in S \setminus s} F^{out}(v) - Fin(v)$
  - recall S is the cut containing s and excluding t
- Since $F^{in}(s) = 0$, we can rewrite the flow as.
- Flow of the network = $\sum_{v \in S} F^{out}(v) - Fin(v)$



Flow of a cut = Flow of the network

- Flow of the network = $\sum_{v \in S} F^{out}(v) - Fin(v)$
- Each vertex v in S (red vertices) has two types of edges
  - Grey edges (the edges that connect the vertex to another vertex in S)
  - Green edges (the edges that connect the vertex to a vertex in T)
  - Let $F^{out-grey}(v)$ be the total flow out from v via grey edges. Similarly, $F^{in-grey}(v)$ be the total flow coming to v via grey edges.
  - Let $F^{out-green}(v)$ be the total flow out from v via green edges. Similarly, $F^{in-green}(v)$ be the total flow coming to v via green edges.
  - We have $F^{out-green}(v) + F^{out-grey}(v) = F^{out}(v)$ and $F^{in-green}(v) + F^{in-grey}(v) = F^{in}(v)$
- Flow of the network = $\sum_{v \in S} F^{out-green}(v) + F^{out-grey}(v) - (F^{in-green}(v) + F^{in-grey}(v))$
- Flow of the network = $\sum_{v \in S} F^{out-green}(v) - Fin^{-green}(v) + F^{out-grey}(v) - F^{in-grey}(v)$
- Note that $\sum_{v \in S} F^{out-grey}(v) - F^{in-grey}(v) = 0$ because each grey edge appears once as an incoming edge for one vertex and once as an outgoing edge for another vertex.
- Flow of the network = $\sum_{v \in S} F^{out-green}(v) - Fin^{-green}(v)$
- Flow of the network = $Flow\ of the\ cut$



Min-cut Max-Flow Theorem

- Min-cut of a flow network is the cut with the minimum capacity

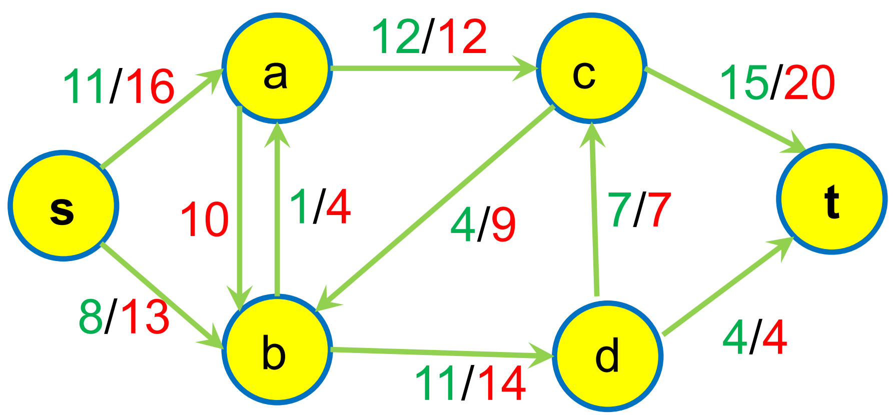
We know that

1. Flow of a cut ≤ capacity of the cut
2. Flow of **<u>every</u>** cut = Flow of the network

- Therefore, Maximum possible flow of the network ≤ capacity of every cut
- Or, Maximum possible flow of the network ≤ capacity of min-cut
- What if we can find a cut such that the flow of the network = capacity of the cut
  - This would mean flow of the network is the maximum possible (we have found maximum possible flow)
  - The cut is the min-cut of the flow network

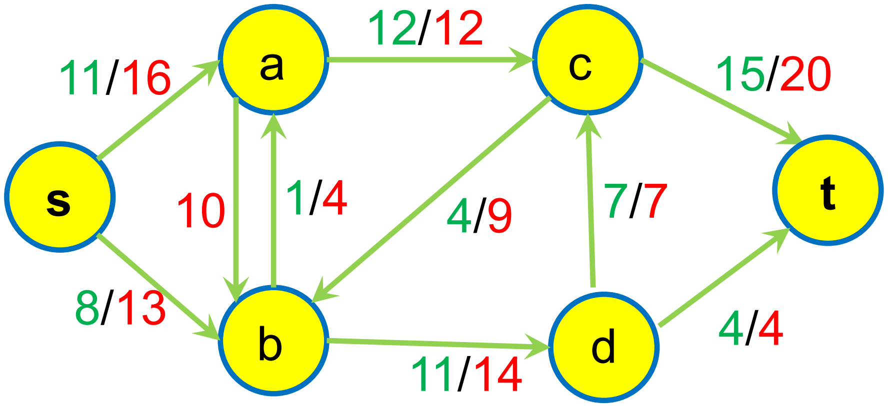
Min-cut Max-Flow Theorem

- Maximum possible flow of a network = capacity of the min-cut



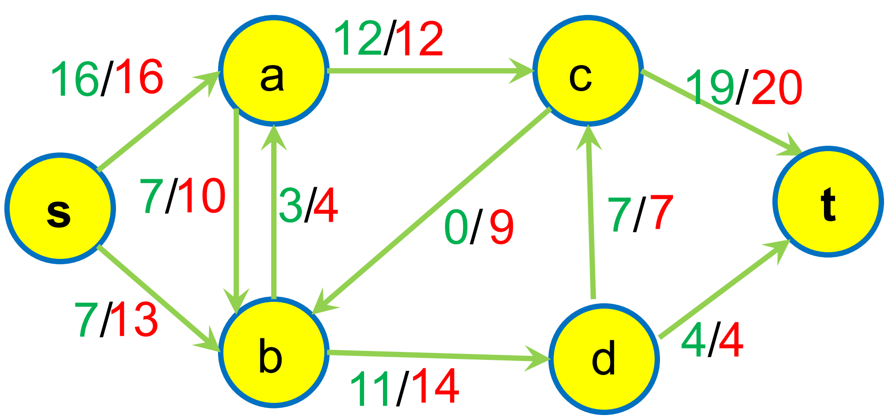
Min-cut Max-Flow Theorem

- Capacity of a cut (S,T) is the total capacity of its **outgoing** edges
- Flow of a cut (S,T) = Total flow of its **outgoing** edges – total flow of its **incoming** edges
- Is it true that flow of a cut = capacity of the cut when:
  1. Flow on **each** outgoing edge of the cut is equal to the capacity of the edge; AND
  2. Flow on **each** incoming edge of the cut is zero
- We show that when the algorithm terminates, there exists a cut for which both of the above two conditions hold which imply that the flow of the cut is equal to its capacity.
  - This guarantees that the flow is maximum



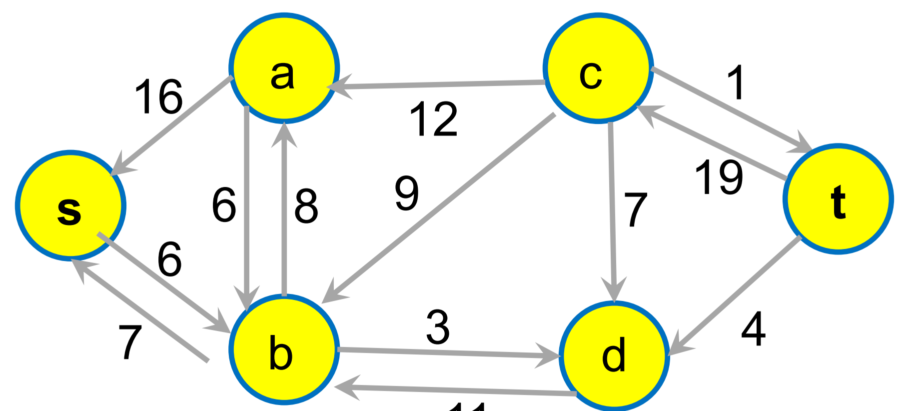
Proof of correctness

- Suppose the algorithm has terminated (there does not exist any augmenting path in the residual network).
- We define a cut (**S**,T) such that
  - **S** contains every vertex v that is reachable from s in **the residual network**.
  - **T** contains every other vertex. Note t cannot be in S because it is not reachable from S (no augmenting path)
- Flow of this cut = Capacity of this cut because
  - For each outgoing edge a→c, its flow is equal to the capacity of the edge
    - Otherwise, we would have an edge a→c in the residual network which would mean c is reachable from s but we know this is not the case as c is not in S.
  - For each incoming edge c → b, its flow is zero.
    - Otherwise, there would be an edge b→c in the residual network implying c is reachable from s but we know this is not the case as c is not in S.
- Therefore, the flow is maximum when the algorithm terminates



Flow Network: Green numbers indicate flow and red indicate capacity. Flow is not shown if 0

f = 23



Residual Network:

Start preparing for the final exam

- Highlights, week 12 lecture
  - Topological sorting
  - Some important information about final exam and more …

Do not procrastinate!

Summary

Take home message

- Maximum flow of a network is equal to its min-cut and can be found using Ford-Fulkerson

Things to do (this list is not exhaustive)

- Make sure you understand
  - the two algorithms
  - understand why Ford-Fulkerson is correct
- Start preparing for the final exam

Coming Up Next

- Topological sorting and final exam