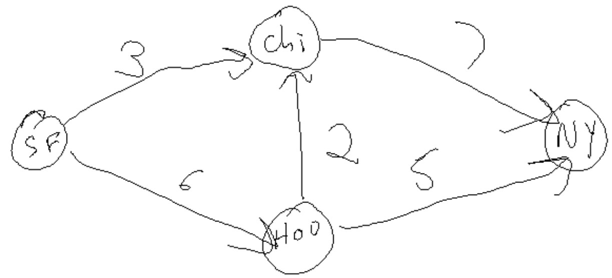
Internet Routing Example

Acme Routing Company wants to route traffic over the internet from San Fransisco to New York. It owns some wires that go between San Francisco, Houston, Chicago and New York. The table below describes how many kilobytes can be routed on each wire in a second. Figure out a set of routes that maximizes the amount of traffic that goes from San Francisco to New York.

Cities	Maximum number of kbytes per second
S.F. - Chicago	3
S.F. - Houston	6
Houston - Chicago	2
Chicago - New York	7
Houston - New York	5



One commodity, one source, one sink

Maximum Flows

- A **flow network** $G = (V, E)$ is a directed graph in which each edge $(u, v) \in E$ has a nonnegative **capacity** .
- If $(u, v) \notin E$, we assume that $c(u, v) = 0$.
- We distinguish two vertices in a flow network: a **source** s and a **sink** t .

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

Capacity constraint: For all $u, v \in V$, we require $0 \leq f(u, v) \leq c(u, v)$.

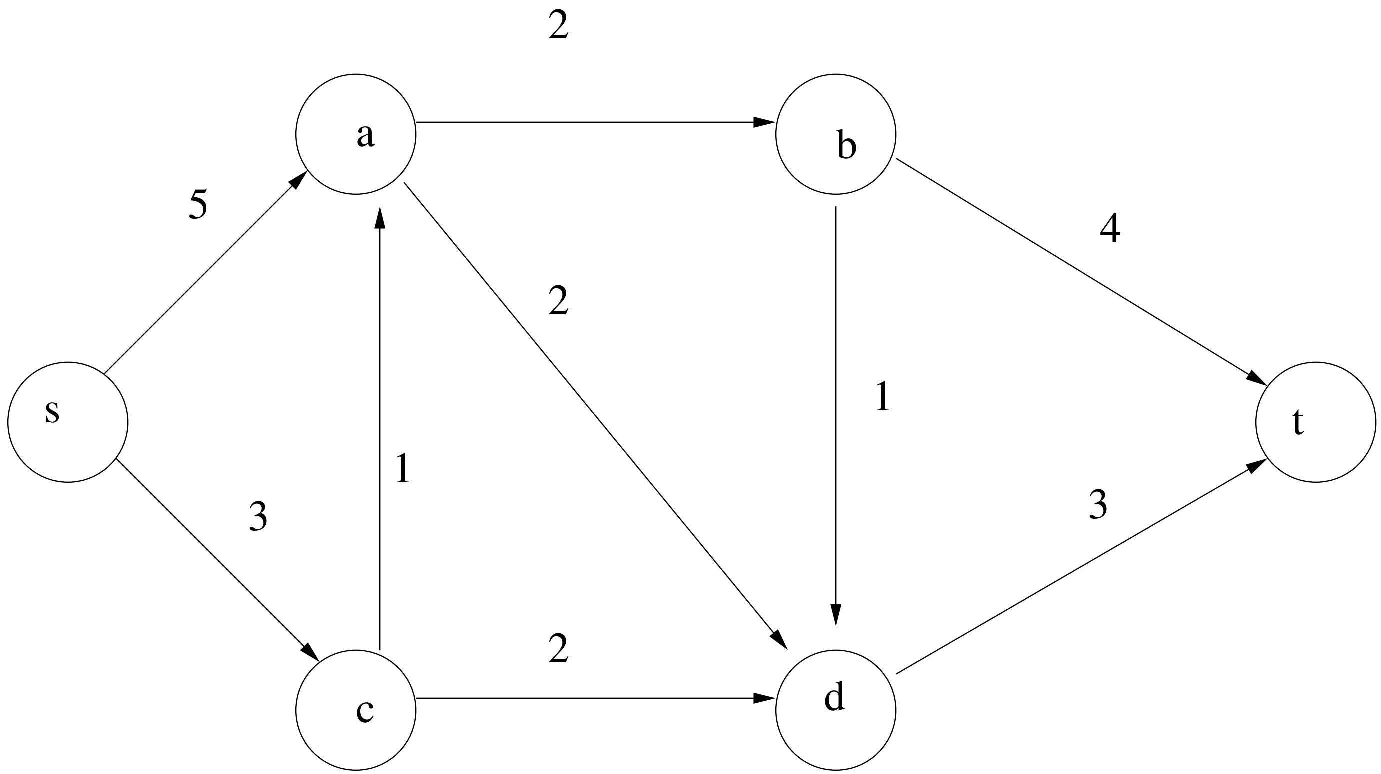
Flow conservation: For all $u \in V - \{s, t\}$, we require

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

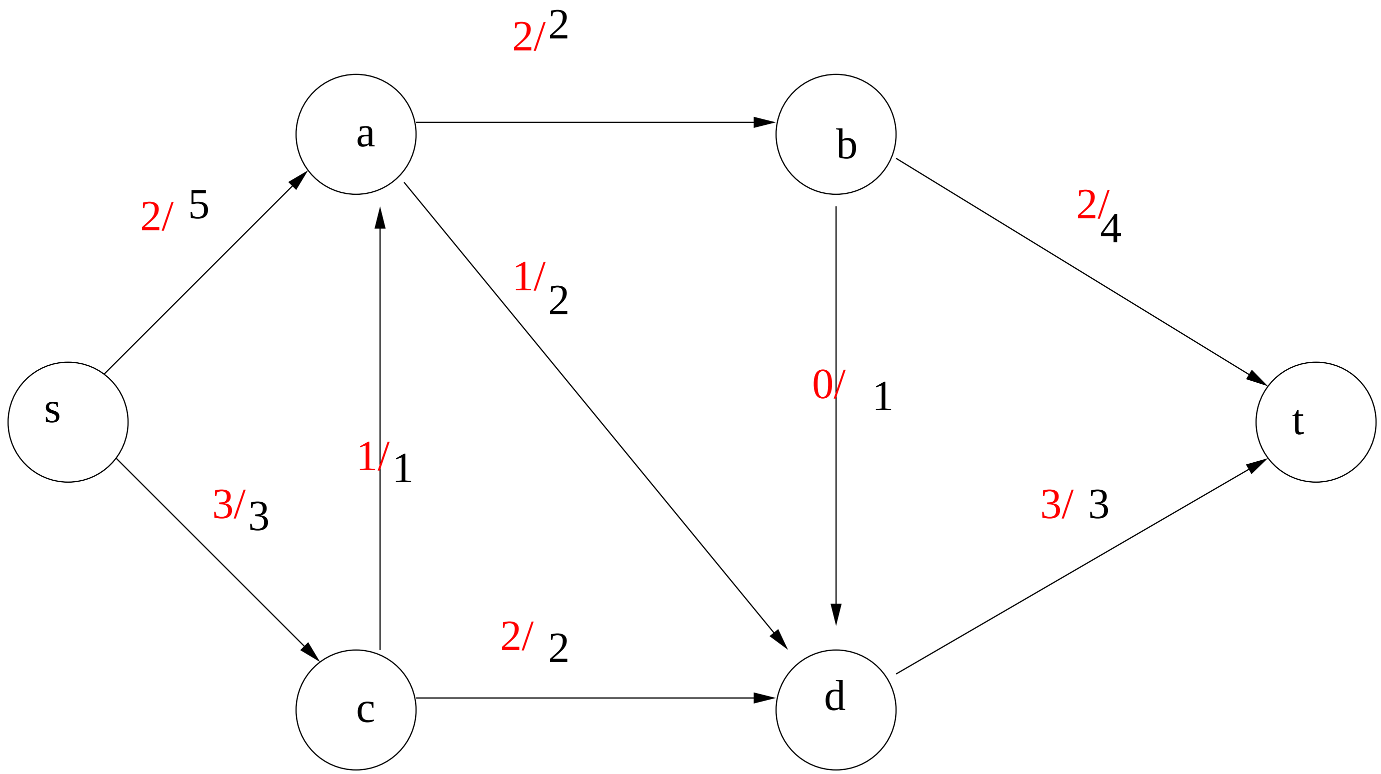
The **value** of a flow f is defined as

$$|f| = \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s) \ , \tag{1}$$

Example



Solutions



Internet Advertising

The New York Times
Monday, August 2, 2010

Sports

Search All NYTimes.com

WORLD | U.S. | N.Y. / REGION | BUSINESS | TECHNOLOGY | SCIENCE | HEALTH | SPORTS | OPINION | ARTS | STYLE | TRAVEL | JOBS | REAL ESTATE | AUTOS

BASEBALL | N.F.L. | COLLEGE FOOTBALL | N.B.A. | COLLEGE BASKETBALL | HOCKEY | SOCCER | GOLF | TENNIS | GLOBAL SPORTS | BUY TICKETS

Candidate's Platform: Jobs. Experience: N.B.A.

By THOMAS KAPLAN

Leah Nash for The New York Times

Chris Dudley campaigned in Beaverton, Ore., to become Oregon's next governor.

Chris Dudley, who stands 6 feet 11 inches and played for the Nets and the Knicks during a 16-year career, stands out while campaigning to become Oregon's governor.

No Sign of Favre at Camp, but the Vikings Don't Seem to Mind

By PAT BORZI

The players, coaches and fans in Mankato, Minn., for the Vikings' training camp seem certain that Brett Favre will be joining the team soon.

Agency Role Could Limit Basketball Broker's Power

By PETE THAMEL

William Wesley's affiliation with Creative Artists Agency could curtail his basketball recruiting.

Assuming Leading Role, Jets' Sanchez Acts the Part

By GREG BISHOP

On the first day of his second season, Mark Sanchez moved with swagger in his step and spoke with urgency in his voice.

In Guillen's Strong Words, Mets See a Nugget of Truth

By DAVID WALDSTEIN 18 minutes ago

Comments by White Sox Manager Ozzie Guillen about a lack of Spanish-language translators in baseball hit home for some Mets.

SPORTS OF THE TIMES

Jets' Ryan Is Leaner and Says He's Learning

By WILLIAM C. RHODEN

It was one thing for Jets Coach Rex Ryan to change his lifestyle; now he wants the rest of the team to follow suit.

· Columnist Page

ON BASEBALL

Showalter Looks Up Again, From Far Below

By TYLER KEPNER 15 minutes ago

Starting Tuesday, Buck Showalter begins the job of resurrecting the Baltimore Orioles, whose last winning season was 1997.

SPORTS OF THE TIMES

The Cameras Are Rolling, and the Jets Expect to Be, Too

By WILLIAM C. RHODEN

With the presence of HBO's "Hard Knocks" crew at training camp, the Jets' mission this season hasn't changed.

Bats

Rodriguez's Waiting Game

August 2, 2010
6:00 PM ET

Showalter Leads the Charge

August 2, 2010 5:59 PM ET

Want Instant Replay in Baseball? Try the Little League World Series

August 2, 2010 4:42 PM ET

Go to the Bats Blog »

The Fifth Down

Jets' McKnight Quiets Fitness Concerns

MLB | P.G.A. | L.P.G.A. | M.L.S. | A.T.P. | W.T.A.

MLB Scoreboard »

AMERICAN LEAGUE

Toronto	8	Bot
NY Yankees	5	6th
Cleveland	6	Bot
Boston	2	7th
Minnesota	2	Bot
Tampa Bay	4	7th
Kansas City		10:05 ET
Oakland		

NATIONAL LEAGUE

Cincinnati	3	Bot

NEWS FROM AP & REUTERS »

Roddick Finding Form With U.S. Open on The Horizon
27 minutes ago

Sabres Agree to 2-Year Deal With D Morrisonn
40 minutes ago

Osborne: Move to Big Ten About Stability
40 minutes ago

Cowboys Hurt Rookie Bryant Still Catches Balls
41 minutes ago

NBA Surpasses 2 Million Followers on Twitter
50 minutes ago

Vandy Drops 'Interim' From Caldwell Title
8:08 p.m.

N.F.L. Roundup: The Jets' Owner Is Taking a Holdout by a Star Cornerback

Availability Queries

Given:

- **Forecasts on ad impressions for each publisher's slot**
- **A reservation request**

Compute: **The maximum number of reservations that can be satisfied from the reservation request.**

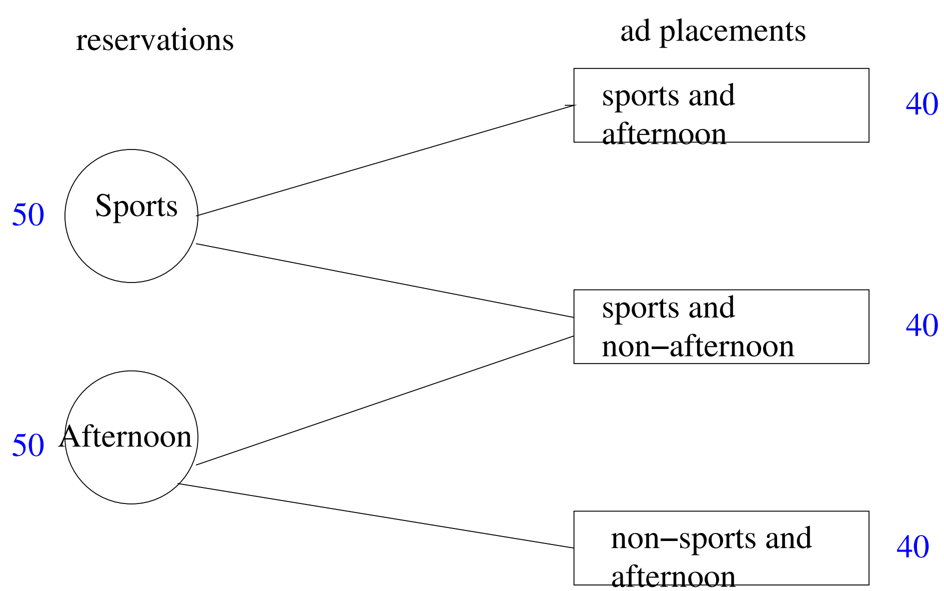
Example

An advertiser wants:

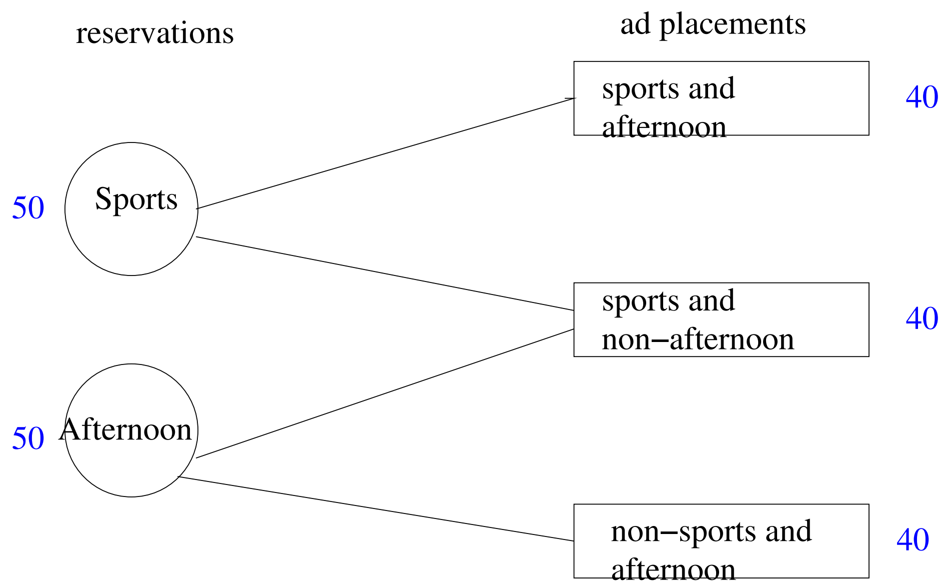
- **50 placements on sports slots**
- **50 placements on afternoon slots**

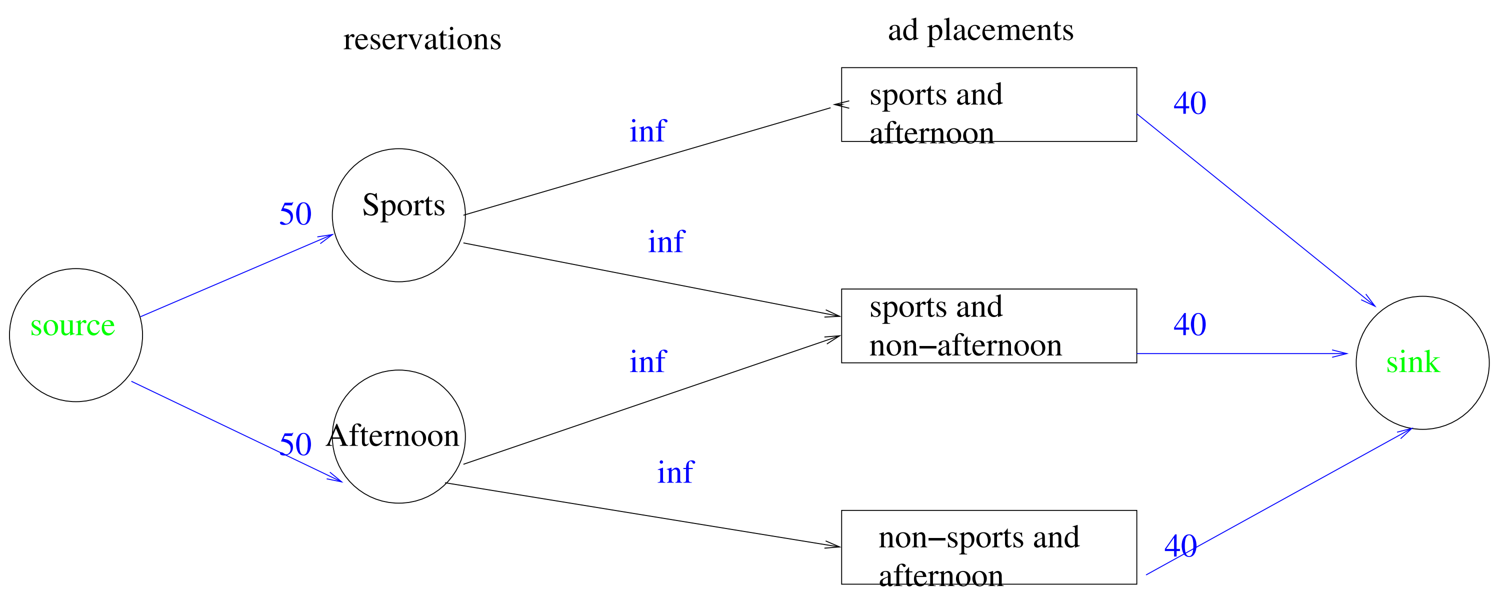
Publisher is offering

- **40 placements on slots that are sports and non-afternoon**
- **40 placements on slots that are sports and afternoon**
- **40 placements on slots that are non-sports and afternoon**



Example is a max flow problem





Algorithm: Ford Fulkerson

Greedily send flow from source to sink.

Ford-Fulkerson-Method (G, s, t)

initialize flow f **to** 0
while there exists an augmenting path p
    augment flow f **along** p
return f

For this to work, we need a notion of a residual graph

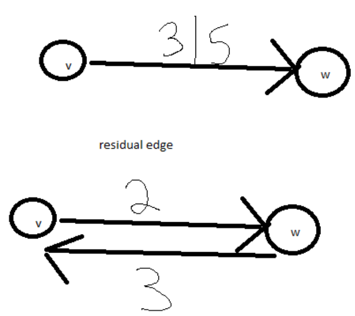
Residual Graph

The residual graph is the graph of edges on which it is possible to push flow from source to sink.

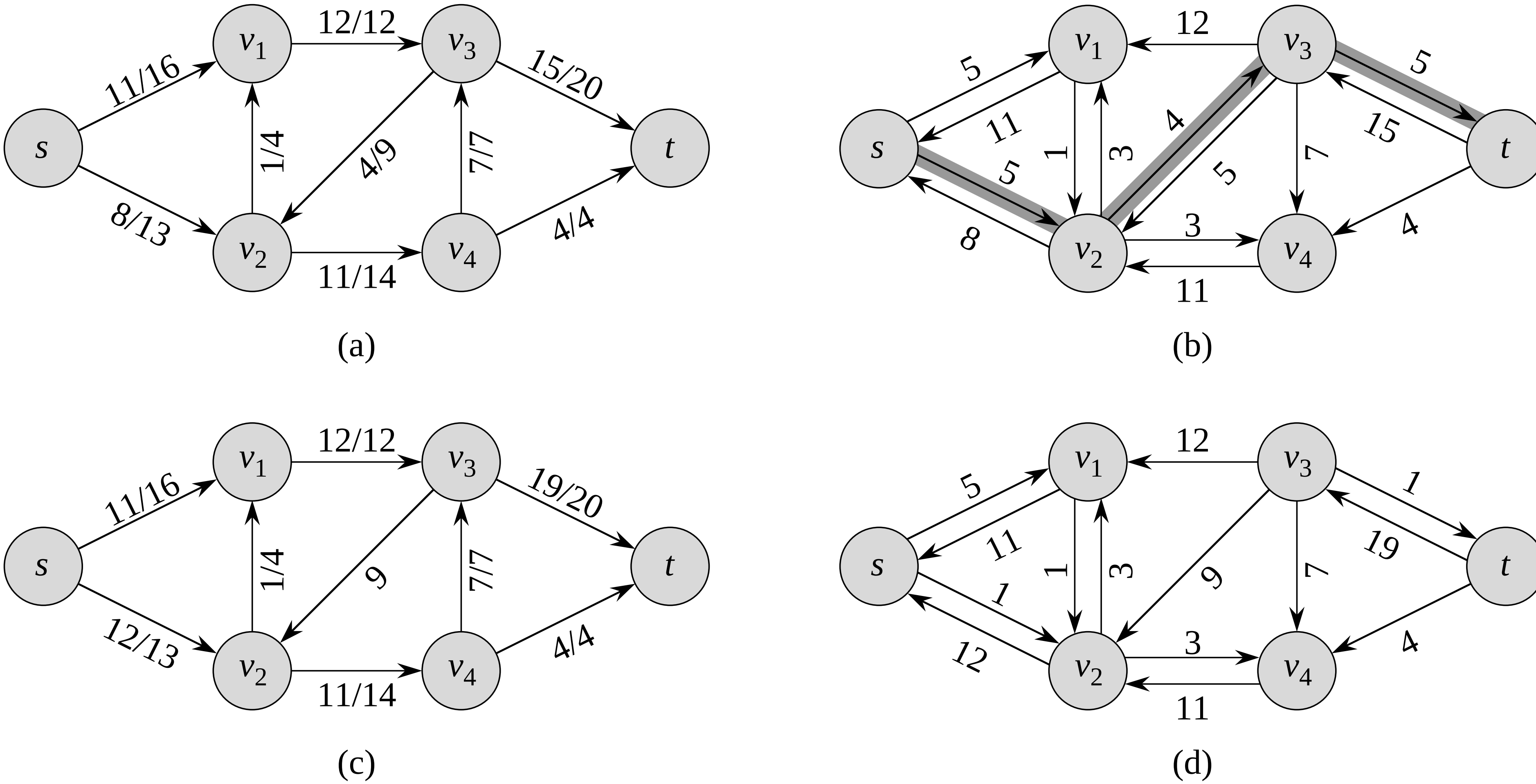
- **The residual capacity of (u, v), is**

$$c_f(u,v) = \begin{cases} c(u,v) - f(u,v) & \textbf{if } (u,v) \in E\ , \\ f(v,u) & \textbf{if } (v,u) \in E\ , \\ 0 & \textbf{otherwise}\ . \end{cases} \tag{2}$$

- **The residual graph G_f is the graph consisting of edges with positive residual capacity**

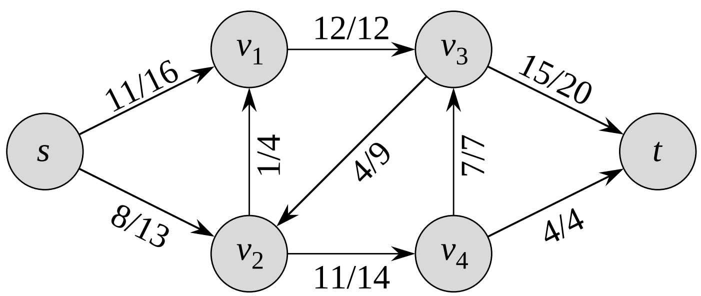


Residual Network

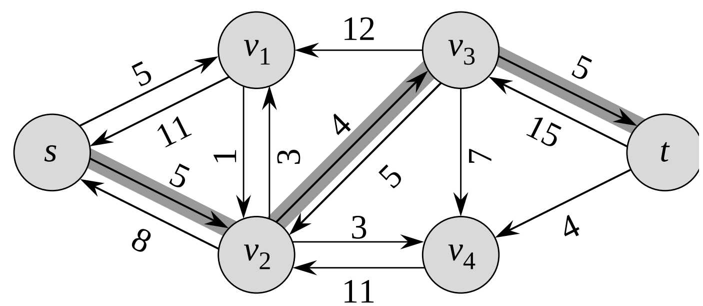


Updating a Flow

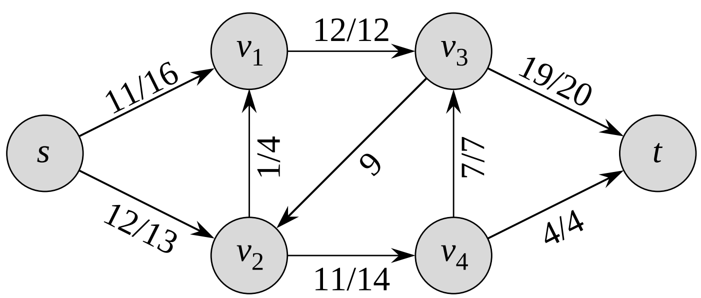
- Send flow along the path defined by the residual graph.
- Amount: minimum of capacity of all residual edges in the augmenting path.
- If a residual edge is a graph edge, then add the flow.
- If a residual edge is a reverse edge, then subtract the flow.



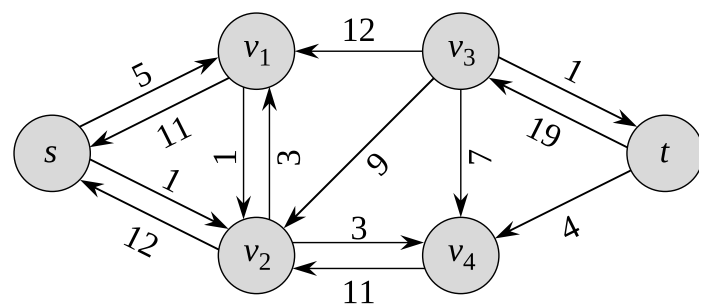
(a)



(b)



(c)

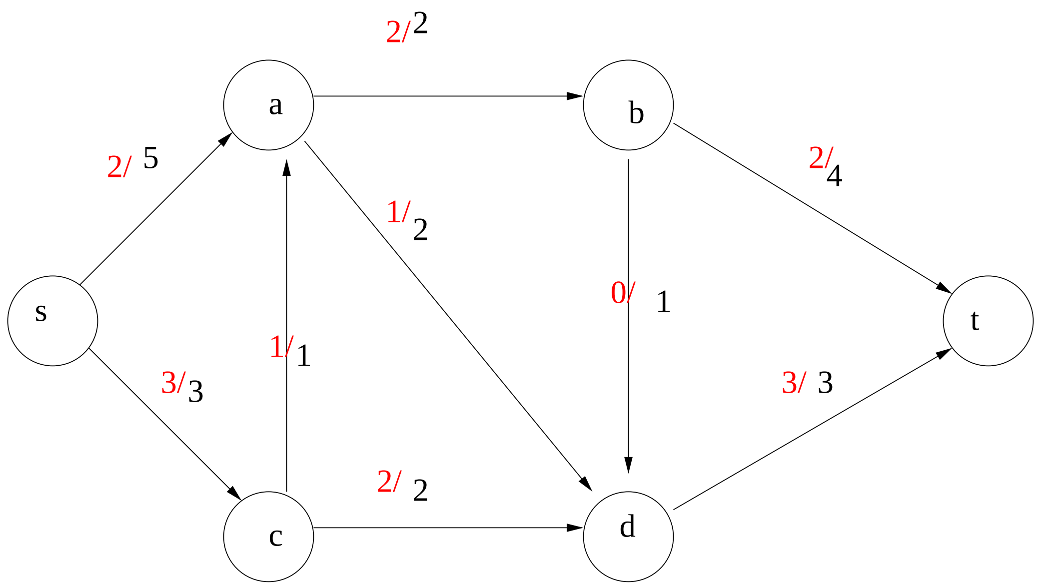


(d)

s - t Cuts

An s - t cut satsfies

- $s \in S$, $t \in T$
- $S \cup T = V$, $S \cap T = \emptyset$



Capacity of a cut **(only forward edges)**

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

Flow crossing a cut **(net flow)**

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

Properties of cuts and flows

Capacity of a cut (only forward edges)

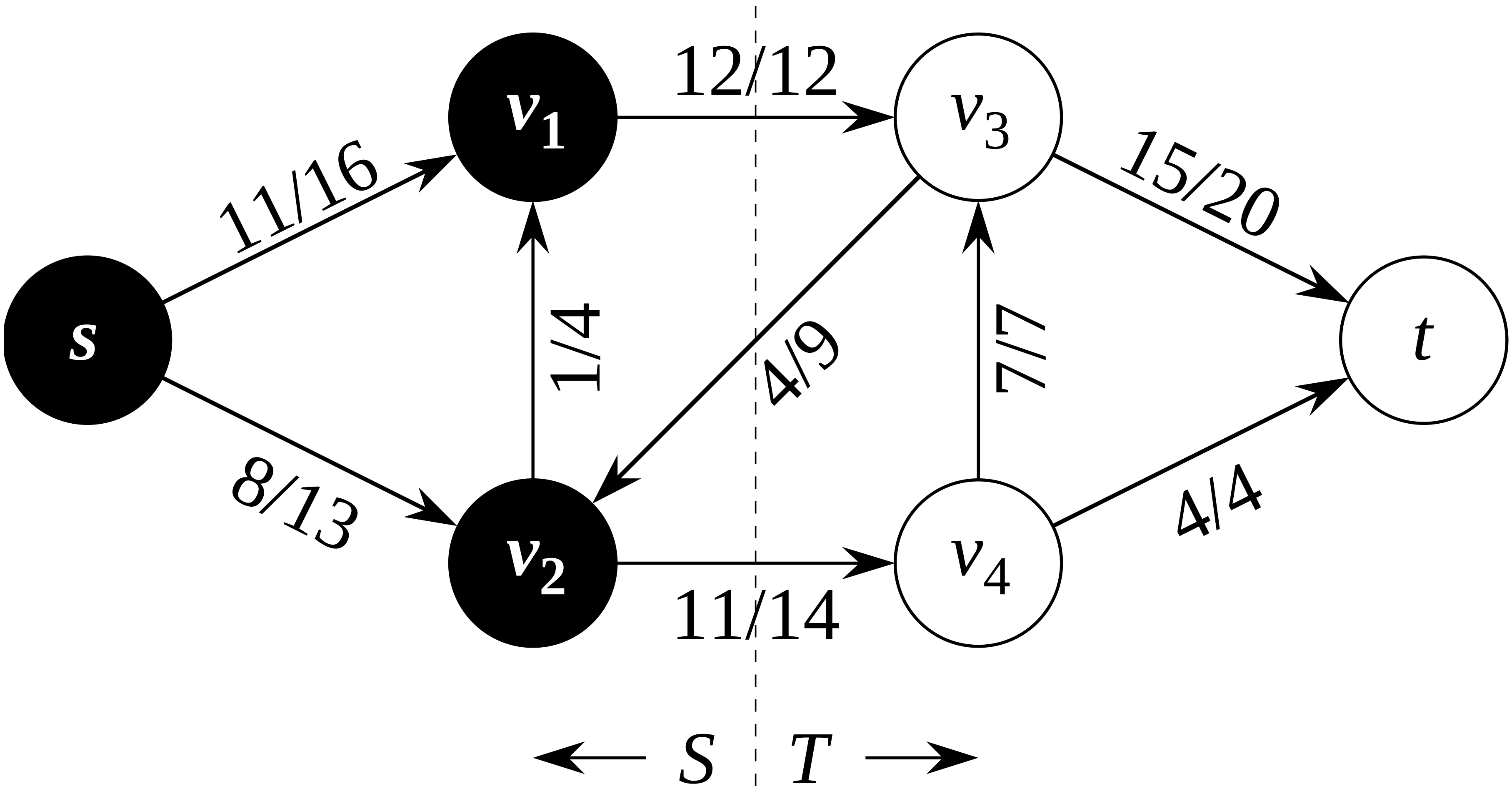
$$c(S,T) = \sum_{u \in S, v \in T} c(u,v)$$

Flow crossing a cut (net flow)

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

- **For all cuts (S,T) and all feasible flows f , $f(S,T) \leq c(S,T)$ (weak duality).**
- **For all pairs of cuts (S_1,T_1) and (S_2,T_2) , and all feasible flows f , $f(S_1,T_1) = f(S_2,T_2)$.**

Examples of cuts



Max-flow min-cut theorem

If f is a flow in a flow network $G = (V, E)$ with source s and sink t, then the following conditions are equivalent:

1. **f is a maximum flow in G.**
2. **The residual network G_f contains no augmenting paths.**
3. **$|f| = c(S, T)$ for some cut (S, T) of G.**

Proof

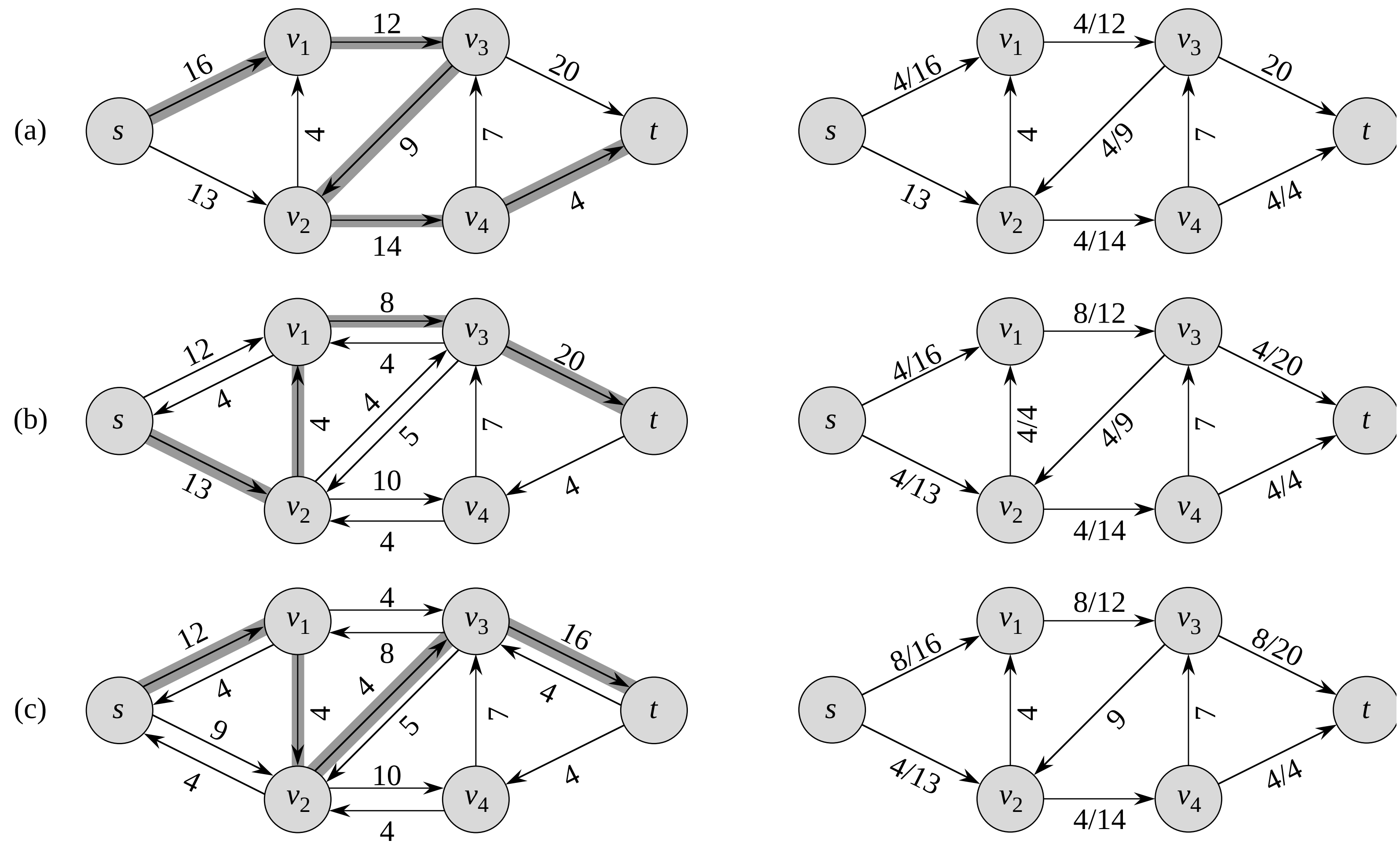
Ford Fulkerson expanded

$Ford - Fulkerson(G, s, t)$

for each edge $(u, v) \in E(G)$
    $f(u, v) = 0$
while there exists a path p **from** s **to** t **in the residual network** G_f
    $c_f(p) = \min\{c_f(u, v) : (u, v)$ **is in** $p\}$
    for each edge (u, v) **in** p
        if $(u, v) \in E$
            $f(u, v) = f(u, v) + c_f(p)$
        else $f(v, u) = f(v, u) - c_f(p)$

Algorithm

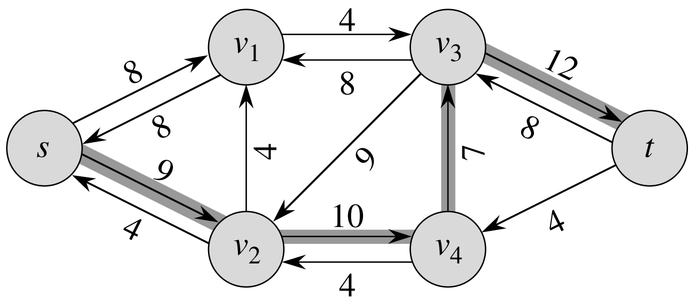
Residual graph (Capacities) | **Graph (Flow/Capacity)**

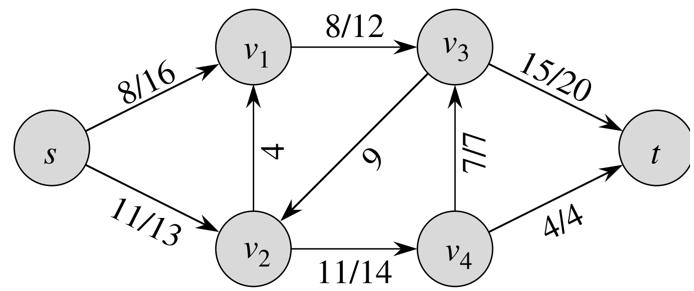


Algorithm continued

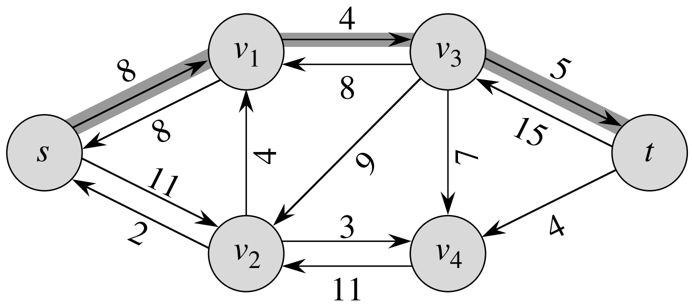
Residual graph (Capacities) **Graph (Flow/Capacity)**

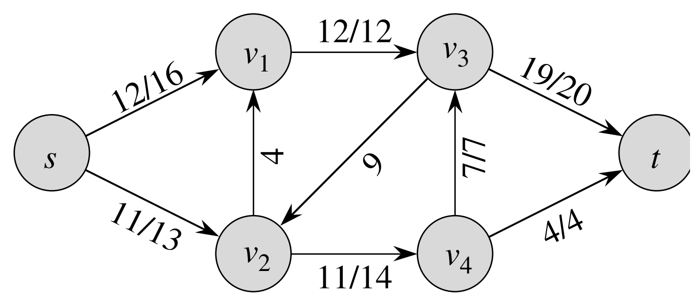
(d)



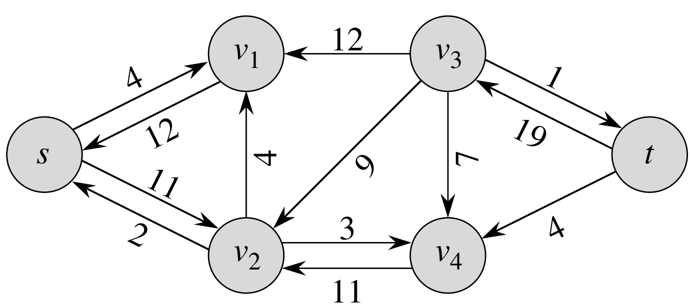


(e)



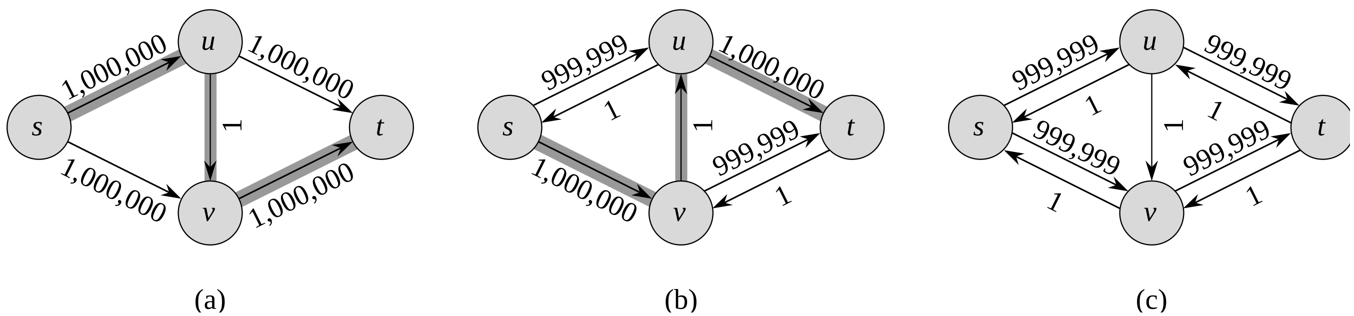


(f)



Analysis

- 1 iteration of FF takes $O(E+V)$ time (breadth-first search plus book-keeping).
- Each iteration sends at least one unit of flow.
- Total time $O(f^*E)$.
- This algorithm is only psuedo-polynomial.



A polynomial algorithm– Shortest Augmenting Path

Algorithm due to Edmonds and Karp, Dinic

Algorithm

- **Run Ford-Fulkerson, but always choose the shortest augmenting path in the residual graph.**
- **Breadth-first search implements this algorithm.**

Idea for Analysis

Intuition

- **Augment along shortest path in residual graph.**
- **Short paths get "saturated" and disappear.**
- **Future augmenting paths are along longer paths**
- **Eventually algorithm terminates because no more paths exist (remember that no augmenting path can have length greater than V.**

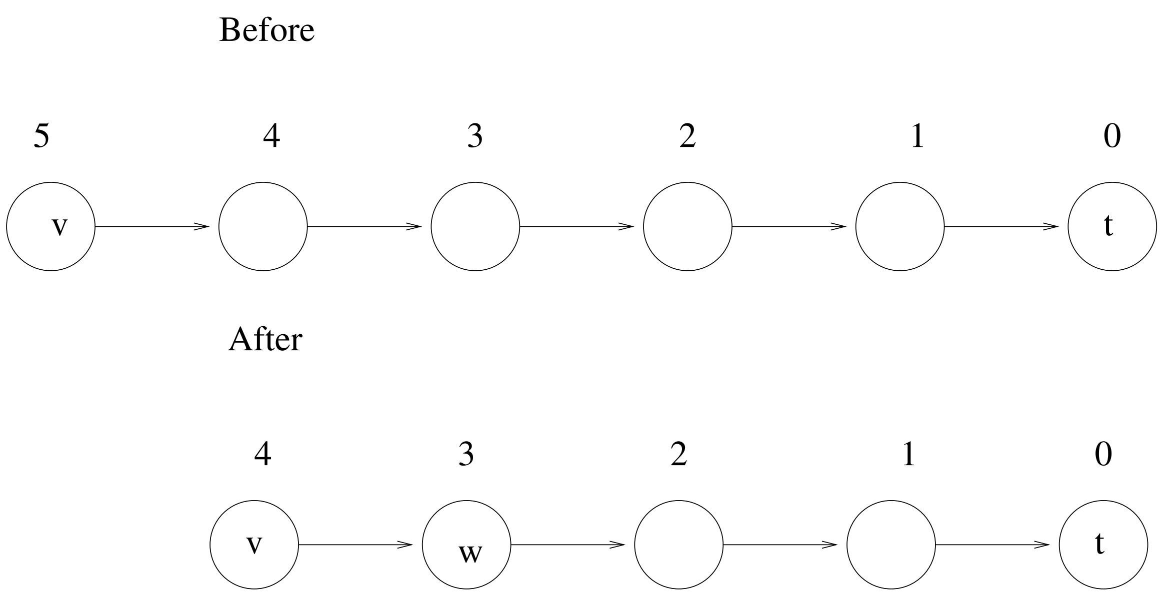
This doesn't quite work, but is close

Analysis

- **Let $\delta(v)$ be the shortest path distance from v to t in G_f.**
- **Lemma: $\delta(v)$ is monotonically increasing over time.**

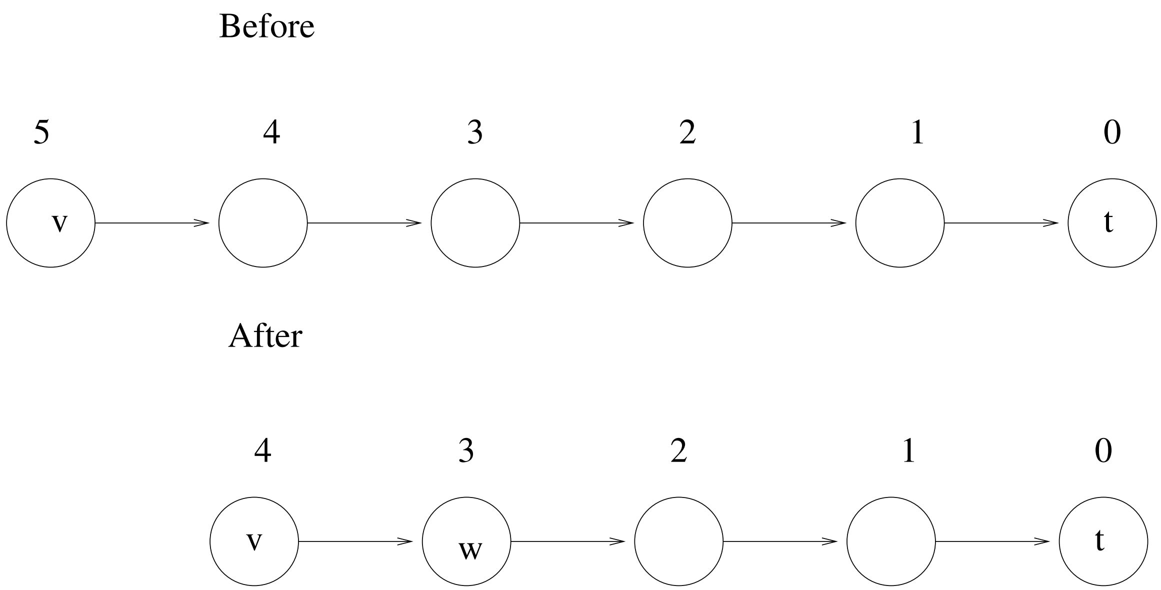
Proof:

- **Let δ be shortest path distances before an augmentation.**
- **Let δ' be shortest path distances after an augmentation.**
- **Suppose, f.p.o.c. that for some v, $\delta'(v) < \delta(v)$, and v is the minimum vertex with this property (w.r.t δ')**



How could this happen?

How could this happen?



- $(v, w) \in G'_f$ **but not in** G_f **. (why?)**
- **We must have sent flow on** (w, v) **in the augmenting path**
- $\delta(w) = 6$
- $\delta'(w) < \delta(w)$ **, which contradicts** v **being the minimum vertex whose label decreased.**

Conclusion: $\delta(v)$ **never decreases.**

Analysis Continued

- **Each $\delta(v)$ increases at most V times.**
- **Total number of times any $\delta(v)$ increases is at most V^2 .**
- **We need to tie this to an augmenting path.**
- **Problem: A particular augmenting path may not increase any $\delta(v)$. (How can this happen?)**

Solution We need to find some other event that happens on every augmenting path and then relate this event to increasing distances.

Conclusion: $\delta(v)$ never decreases.

Analysis Continued

- $0 \le \delta(v) \le V$
- $\delta(v)$ never decreases.
- Each $\delta(v)$ increases at most V times.
- Total number of times some $\delta(v)$ increases is at most V^2 .
- We need to tie this to an augmenting path.
- **Problem:** A particular augmenting path may not increase any $\delta(v)$. (How can this happen?)

Solution We need to find some other event that happens on every augmenting path and then relate this event to increasing distances.

Edge Saturation

- If an augmeting path send $c_f(v, w)$ units of flow on edge (v, w) then we call the push saturating.
- Every augmenting path saturates at least one edge. (Why?)

Analysis Continued

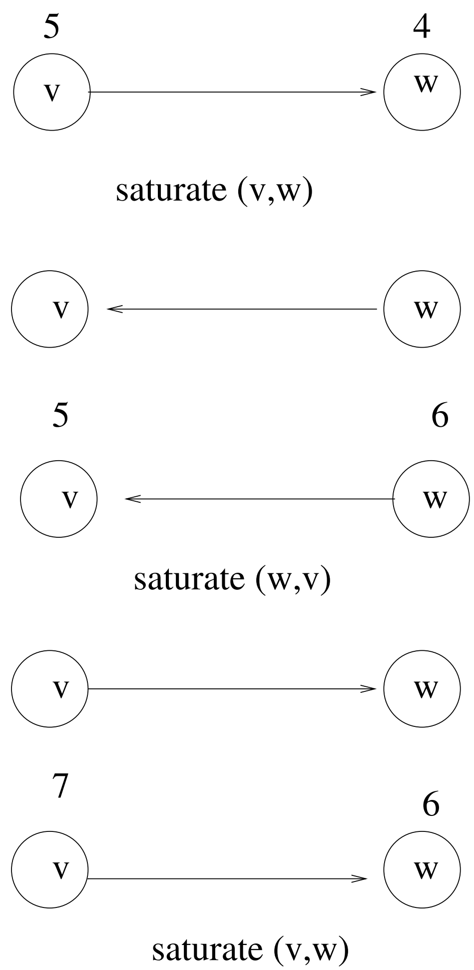
Edge Saturation

- If an augmeting path send $c_f(v, w)$ units of flow on edge (v, w) then we call the push saturating.
- Every augmenting path saturates at least one edge.

Focus on a particular pair of vertices (v, w)

- Suppose that $(v, w) \in G_f$
- What has to happen between two consecutive saturations of (v, w)

Events Between 2 edge saturations



- Both v and w have to be relabeled.
- So each edge can be saturated at most $V/2$ times.

Putting it together

- Each edge (or its reverse) can be saturated at most $V/2$ times.
- There are most EV edge saturations.
- Each augmenting causes at least one edge saturation
- Therefore, there are at most EV augmenting paths.
- Recall $O(E)$ time per augmenting path.

A max flow can be found in $O(E^2V)$ time .

Faster augmenting path based algorithms exist, most based on finding the augmenting paths more efficiently.

- $O(VE\log V)$ (Sleator and Tarjan)
- $O(\min E^{3/2}, V^{2/3}E\log(V^2/E)\log C)$ (Goldberg and Rao)

Push relabel algorithms

Algorithm due to Goldberg and Tarjan

Ideas

- Maintain variables $d(v)$ which are "distances", estimates (lower bounds) on the distance to the sink (or source) in the residual graph
- Push flow over one edge at a time, pushing from a higher distance vertex to a lower distance vertex.
- Allow excess flow to accumulate at a vertex.
- Maintain a preflow rather than a flow.

Capacity constraint: For all $u, v \in V$, we require $0 \leq f(u,v) \leq c(u,v)$.

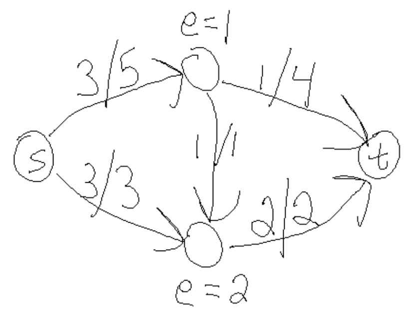
Relaxed Flow conservation: For all $u \in V - \{s,t\}$, we require

$$\sum_{v \in V} f(v,u) \geq \sum_{v \in V} f(u,v) \ .$$

Define excess

$$e(u) = \sum_{v \in V} f(v,u) - \sum_{v \in V} f(u,v) \tag{3}$$

Example


e=1
3/5
1/4
s
t
1/1
3/3
2/2
e=2

Comparison with augmenting paths

- **Augmenting paths**
  - **Always maintains a flow (feasibility)**
  - **Works towards optimality (maximum flow)**
- **Push/relabel**
  - **Always maintains a preflow and works towards a flow**
  - **Maintains a superoptimal preflow and moves towards feasibility.**

Basic Operations

- **If** $e(v) > 0$ **then v is active.**
- **If** $c_f(v, w) > 0$ **and** $d(v) = d(w) + 1$ **then** (v, w) **is admissible.**

Push **from active vertices over admissible edges.**

$Push(u, v)$

// **Applies when:** u **is active,** $c_f(u, v) > 0$**, and** $d(u) = d(v) + 1$**.**
// **Action: Push** $f(u, v) = \min(e(u), c_f(u, v))$ **units of flow from** u **to** v**.**
$\Delta(u, v) = \min(e(u), c_f(u, v))$
if $(u, v) \in E$
    $f(u, v) = f(u, v) + \Delta(u, v)$
else $f(v, u) = f(v, u) - \Delta(u, v)$
$e(u) = e(u) - \Delta(u, v)$
$e(v) = e(v) + \Delta(u, v)$

Relabel **When a vertex has all its outgoing residual edges pointing uphill, we relabel it.**

$Relabel(u)$

// **Applies when:** u **is active and for all** $v \in V$ **such that** $(u, v) \in E_f$**,**
    we have $d(u) \leq d(v)$**.**
// **Action: Increase the label of** u**.**
$d(u) = 1 + \min\{d(v) : (u, v) \in E_f\}$

Generic Push Relabel Algorithm

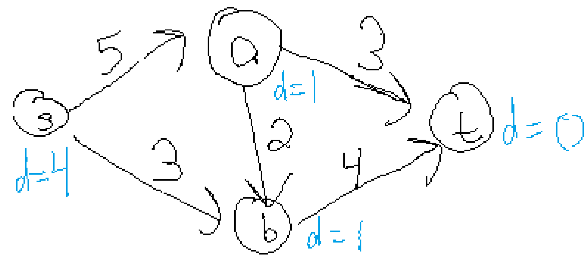
$Initialize - Preflow(G, s)$

for each vertex $v \in V(G)$
    $d(v) = 0$
    $e(v) = 0$
for each edge $(u, v) \in E(G)$
    $f(u, v) = 0$
$d(s) = |V(G)|$
for each vertex $v \in Adj(s)$
    $f(s, v) = c(s, v)$
    $e(v) = c(s, v)$
    $e(s) = e(s) - c(s, v)$

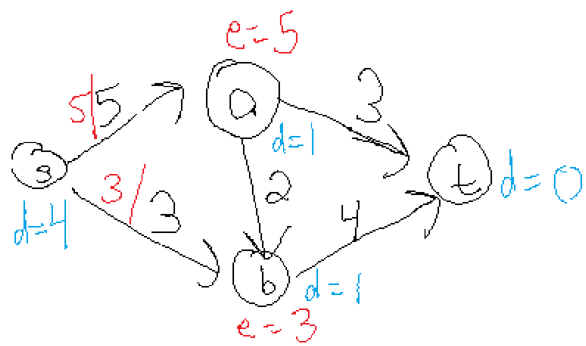
$Generic - Push - Relabel(G)$

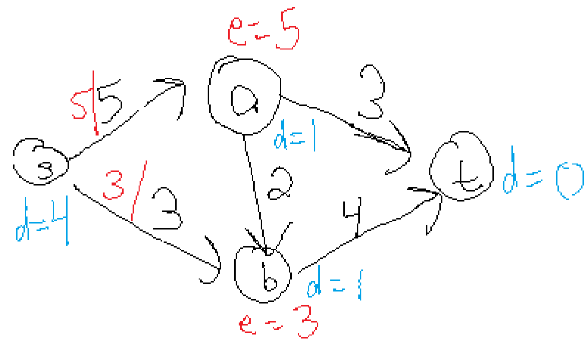
INITIALIZE-PREFLOW(G, s)
while there exists an applicable push or relabel operation
    select an applicable push or relabel operation and perform it

Run Through Example

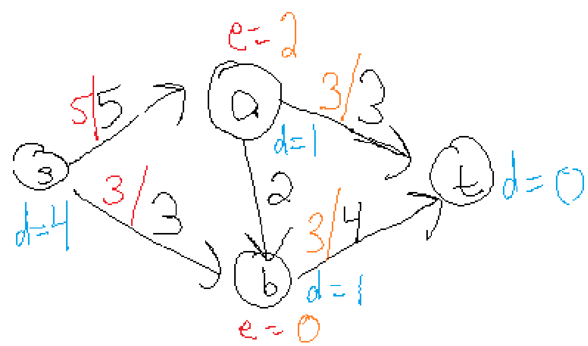


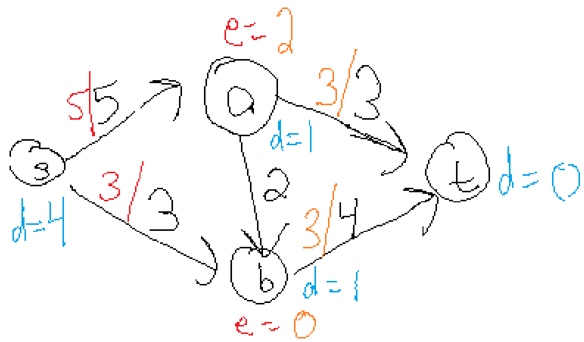
init



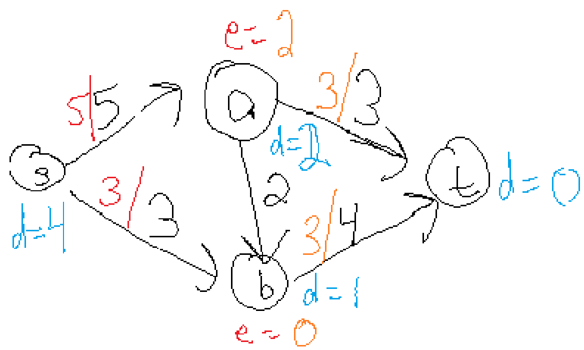


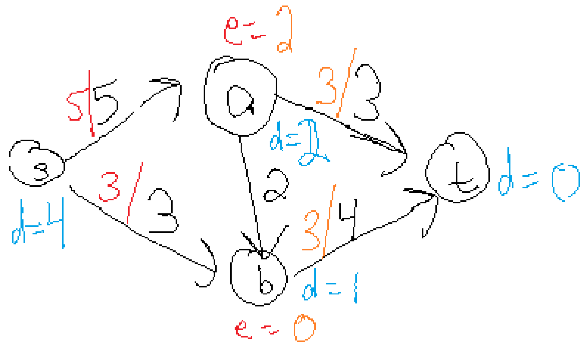
push(a,t)
push(b,t)



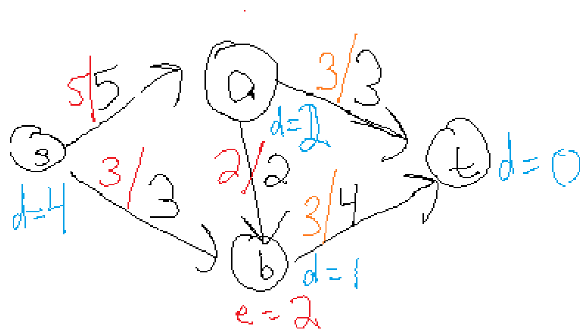

e=2
5/5
3/3
a
d=1
2
3/4
s
d=4
3/3
t
d=0
b
d=1
e=0

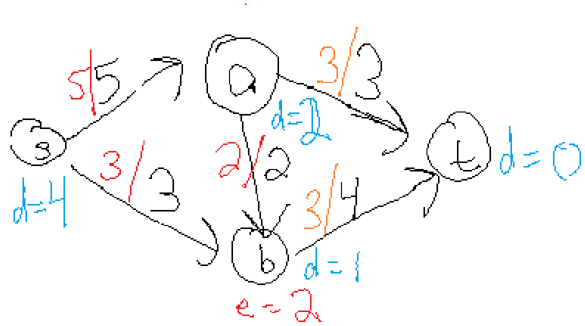
relabel a


e=2
5/5
3/3
a
d=2
2
3/4
s
d=4
3/3
t
d=0
b
d=1
e=0

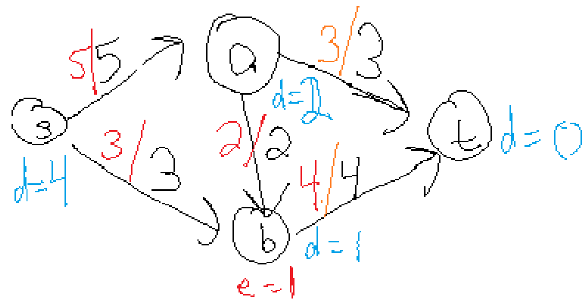


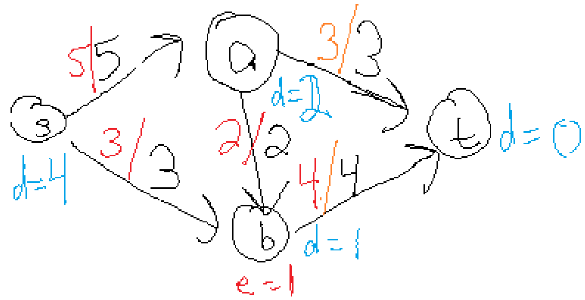
push(a,b)



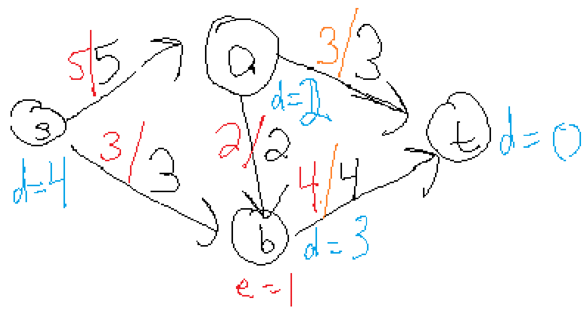


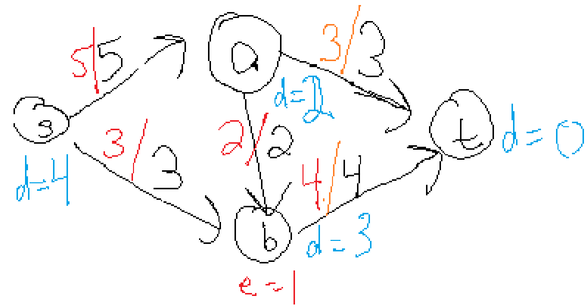
push(b,t)



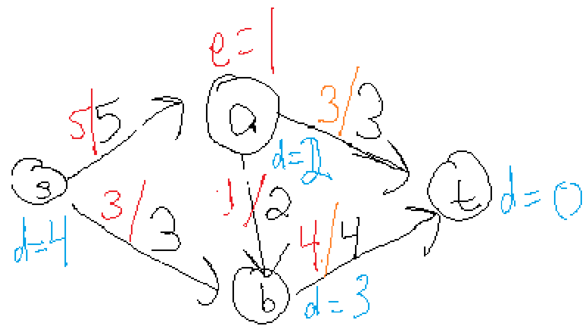


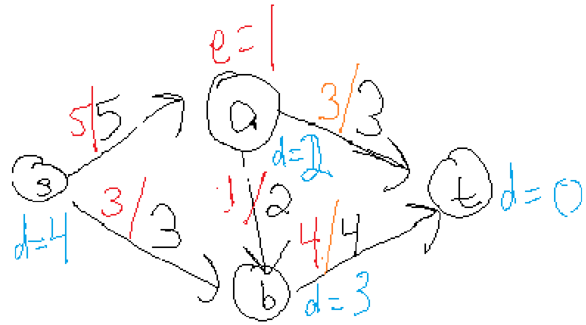
relabel b



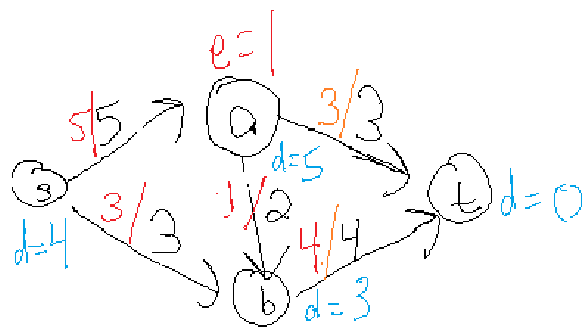


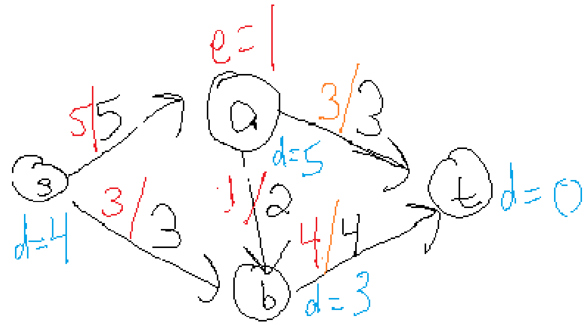
push(b,a)



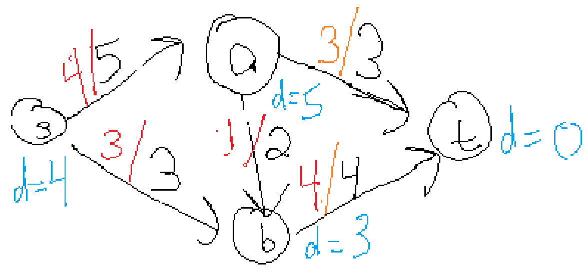


relabel a





push(a,s)



Flow!

Properties of Algorithm (without proofs)

- **At any point, any active vertex either has a path to the source or the sink in the residual graph.**
- **There is never an $s-t$ path in G_f.**
- **Let $\delta(v)$ be the minimum of distance to sink and V plus distance to soure in G_f.**
- **If v is active then $d(v) \le \delta(v)$. (inductive proof, similar to shortest augmenting path, new short paths are not created).**
- $d(v) \le d(w) + 1$ **for any $(v, w) \in G_f$.**
- **If $e(v) > 0$ then either**
  - v **has an outgoing admissible edge**
  - v **can be relabeled (and the label will increase)**
- **If $e(v) = 0$ for all $v \in V - \{s, t\}$, then f is a maximum flow.**

Proof of last statement

- f **is a flow by the definition of preflow.**
- f **is maximum because there is no $s-t$ path in G_f.**

Running Time

We will need to bound

1. **Number of relabels**
2. **Time per relabel**
3. **Number of pushes**
4. **Time per push**
5. **Bookkeeping time**

Easy ones:

Relabel(u)

// **Applies when:** u **is active and for all** $v \in V$ **such that** $(u, v) \in E_f$**,**
we have $d(u) \leq d(v)$**.**
// **Action: Increase the label of** u**.**
$d(u) = 1 + \min\{d(v) : (u, v) \in E_f\}$

Time per relabel

- **Easy bound:** $O(V)$
- **Time to relabel each vertex once :** $O(V^2)$

Easy ones:

Relabel(u)

// **Applies when:** u **is active and for all** $v \in V$ **such that** $(u, v) \in E_f$,
we have $d(u) \le d(v)$.
// **Action: Increase the label of** u.
$d(u) = 1 + \min\{d(v) : (u, v) \in E_f\}$

Time per relabel

- **Easy bound:** $O(V)$
- **Time to relabel each vertex once :** $O(V^2)$
- **Better bound:** $O(degree(v))$
- **Time to relabel each vertex once:** $\sum_v degree(v) = O(E)$

(This is an example of amortized analysis)

Number of relabels

- **Vertex labels are at most** $2V$ **, so each vertex is relabelled at most** $2V$ **times.**
- **Total time spent relabelling =** $V \sum_v degree(v) = O(EV)$

Pushes

Push(u, v)

// **Applies when:** u **is active,** $c_f(u, v) > 0$**, and** $d(u) = d(v) + 1$**.**
// **Action: Push** $\Delta(u, v) = \min(e(u), c_f(u, v))$ **units of flow from** u **to** v**.**
$\Delta(u, v) = \min(e(u), c_f(u, v))$
if $(u, v) \in E$
  $f(u, v) = f(u, v) + \Delta(u, v)$
else $f(v, u) = f(v, u) - \Delta(u, v)$
$e(u) = e(u) - \Delta(u, v)$
$e(v) = e(v) + \Delta(u, v)$

Easy part

- **Time per push =** $O(1)$ **.**
- **Bookkeeping (can be amortized against other operations with reasonable data structures and rules for choosing push/relabel operations).**

Two types of Pushes

- **Saturating push:** Sends $c_f(v, w))$ flow on (v, w)
- **Non-saturating push:** Sends less flow. ($e(v)$).

Bounding Saturating Pushes

Consider a saturating push on (v, w). What has to happen before the next saturating push on (v, w)?

Bounding Saturating Pushes

Consider a saturating push on (v, w) . What has to happen before the next saturating push on (v, w) ?

- w has to be relabelled
- A push on (w, v) must occur
- v has to be relabelled.

Conclusion

- Between any two saturating pushes on (v, w) , v must be relabelled.
- v can be relabelled at most $2V$ times.
- There are at most $2V$ saturating pushes on (v, w) .
- The total number of saturating pushes is $O(VE)$.

Non-saturating pushes

Issue After a non-saturating push, the graph does not change, nothing need be relabelled and another non-saturating push can occur on the edge before anything significant happnes.

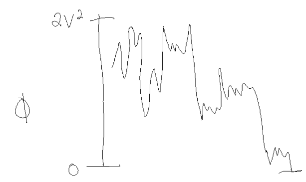
Solution As with shortest-augmeting-path, we won't count directly, but will bound other operations and related to non-saturating pushes. We will do so via means of a potential function.

$$\Phi = \sum_{v:e(v)>0} d(v)$$

Analysis

$$\Phi = \sum_{v:e(v)>0} d(v)$$

- **After initialiazation** $\Phi \le 2V^2.$
- **At termination** $\Phi = 0$ **.**
- **Let** R **be total increase in** Φ **due to relabellings.**
- **Let** S **be total increase in** Φ **due to saturating pushes.**
- **Each non-saturating push decreases** Φ **by at least 1. (Why?).**



2V²
ϕ
0

Putting these facts together

- **Total decrease in Φ associated with non-saturating pushes is at most $2V^2 - 0 + R + S$.**
- **Each non-saturating push decreases Φ by at least 1. (Why?).**
- **Number of non-saturating pushes is at most $2V^2 - 0 + R + S$.**

Bounding R and S

$$\Phi = \sum_{v:e(v)>0} d(v)$$

Relabellings

- **Each relabelling must increase Φ by at least 1.**
- **Total increase in Φ associated with relabelling v is at most $2V$.**
- **Total increase associated with all relabellings is at most $2V^2$.**

Saturating Pushes

- **A saturating push leaves excess at v . It adds excess to w .**
  - **If w already had excess, then Φ is unchanged.**
  - **If w did not have excess, then Φ increases by $d(w)$, which is at most $2V$.**
- **There are at most $O(EV)$ saturating pushes, therefore total increase due to saturating pushes is $O(EV^2)$.**

Putting it Together

Number of non-saturating pushes is at most

$$2V^2 - 0 + R + S = O(V^2 + EV + EV^2) = O(EV^2)$$

.

Conclusion: Running time is $O(EV^2)$.

Note: Can be improved by

- Choosing operations more carefully.
- Better data structures to represent flow
- Best running times are close to VE.
- Winner in practice, assuming that one uses two additional heuristic ideas
  - gap heuristic
  - global relabellings