CS3000
6/12 - mon.

Admin

- Long HW5 due tom. 9pm
- Exam #2 thurs 6/15 during class
- optional recitation tues.
- Wrap-up lecture 6/21 → form on piazza
- mon 6/19 is holiday

Agenda

1. Network flow
2. Ford-Fulkerson algorithm
3. FF example

II. Network Flow

Graph today:

- directed
- ~~a~~ capacity on each edge
- sometimes we add new edges

Previously...

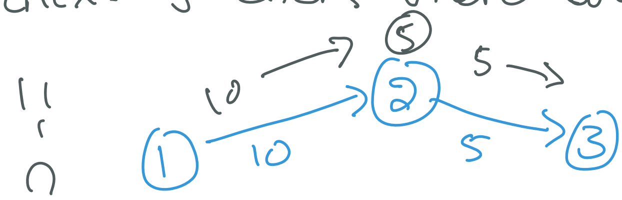
- Dijkstra's / Floyd Warshall
  - ↳ also on directed graphs
  - ↳ weights on edges

→ graphs as maps!
Best way to get from A to B

Now...

- Graphs as flow network
- material moves through system
  - ↳ starts at same place
  - ↳ has same destination
- capacity on each edge tells how much material can move along that edge
- edge: conduit carrying material

• vertex: junction where we change paths



not allowed

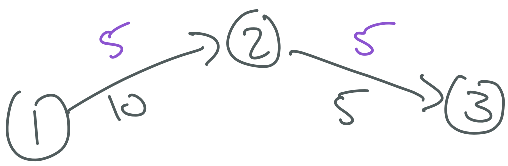
XXXX
vertices do not
store material

Problem to solve:

• How much material can we send through the network, without violating capacity constraints?

☆ optimization problem ☆

maximize flow in a network



→ f is flow

$|f| = 5$

Definitions/Refinements

• Flow network G is a graph with capacity $c(u,v)$ on every edge

• $c(u,v) \geq 0$

• Two special vertices

source → flow starts. no incoming edges

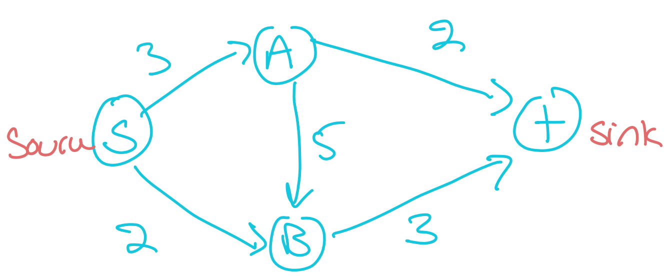
Sink → destination. no outgoing edges

- Assume: integer capacities
- Every edge is assigned a flow $(u,v).f \le c(u,v)$
- Flow f, we want to maximize

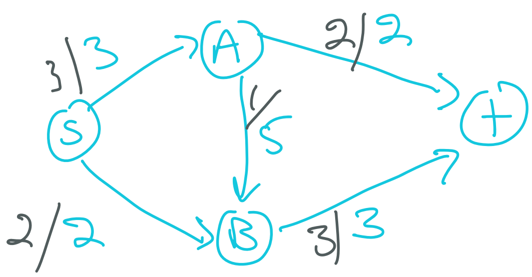
$$|f| = \sum_{v \in V} (s,v).f$$ → s is source

- no edge (u,v), then $c(u,v) = 0$
- start with:
    - ∃ a simple path from source s to sink t
    - every vertex is reachable from s, and can reach the sink t
- no cycles to start

what is actual flow?
what is $|f|$?



Draw the flow ?



$|f| = 5$

Goal

$c((S,B)) = 2$ $\quad c((S,A)) = 3$ $\quad c((A,t)) = 2$

$(S,B).f = 2$ $\quad (S,A).f = 3$ $\quad (A,t).f = 2$

$c((A,B)) = 5$ $\quad c((B,T)) = 3$

$(A,B).f = 1$ $\quad (B,T).f = (S,B).f + (A,B).f = 3$

this is max flow $|f| = f^*$

$\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad$ ↳ max

for this graph:

- nothing else can go into t
- nothing else can leave s

} Sanity check

2. Ford Fulkerson

- goal: network flow ~~thru~~ with $(u,v).f \leq c(u,v)$

$$\text{with } \underbrace{|f| = \sum_{v \in V} (s,v).f}_{\text{what we do}} = \underbrace{f^*}_{\text{ideal (max)}}$$

- can't violate capacity constraints
- can't leave anything in vertex

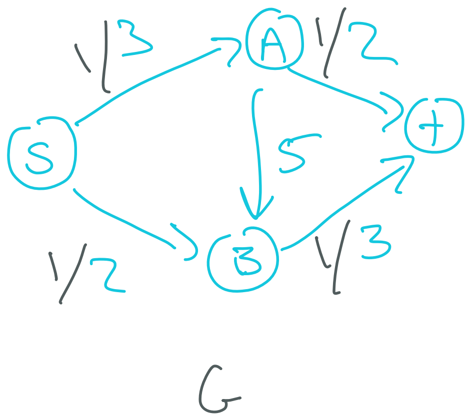
Ford-Fulkerson relies on ...

1. Residual networks
2. Augmenting paths
3. min cuts

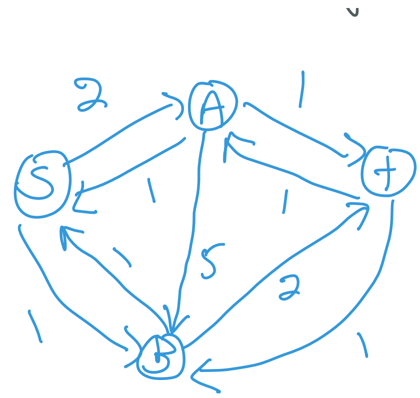
1 → Residual network

- graph G • flow so far f
- residual network G_f: edges with capacities that represent how we can change the flow
- add backwards edges to indicate where we can decrease flow

f so far... G_f on this graph



$|f| = 2$

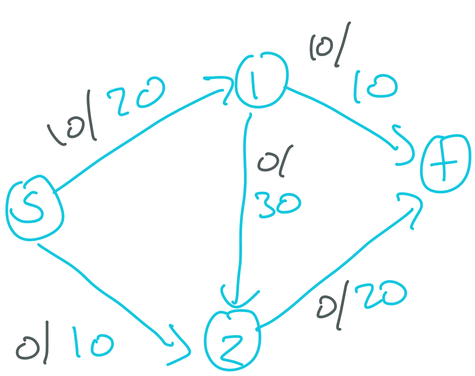


- left over capacities on existing edges
- back edges to decrease flow
- back edges not in G.E

2. Augmenting Paths

- augmenting path $s \rightsquigarrow t$ in residual network G_f that increases flow
- can I send more from s to t in leftover capacity?

(ex)



Flow so far...

	Are these aug?
⟨s, 1, t⟩	no :(
⟨s, 2, t⟩	yes → by 10
⟨s, 1, 2, t⟩	yes → by 10

3 ⟶ min cut

cut: split vertices into two sets $S \subseteq V$, $V - S$

min cut: $\boxed{\text{cut whose edges crossing the cut have min. total capacity over all cuts}}$

whichever is lower!

10:58

Ford-Fulkerson method

while augmenting path exists in residual network

- do min cut
- increase flow along path by min amount

Refinement... (still leaves a few things open)

c_f = capacity in residual network

```
FORD-FULKERSON(G, s, t)
for each edge (u,v) ∈ G.E                                 } Θ(E)
    (u,v).f = 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}                → Θ(E)
    for each edge (u,v) in p
        if (u,v) ∈ G.E
            (u,v).f = (u,v).f + c_f(p)                    → Θ(E)
        else
            (v,u).f = (v,u).f − c_f(p)                    → today's simplification:
```

we can't go into the edge!

FF run-time

- implementation details matter!
- But, we can still bound the run time

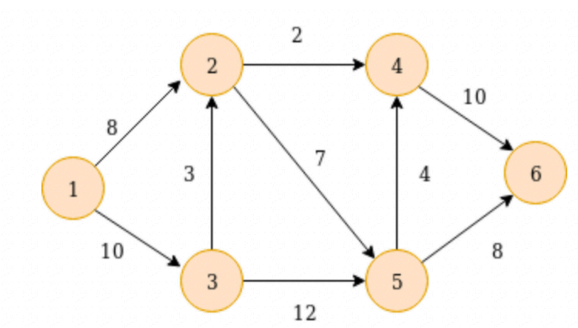
all flows start at 0

while loop runs until $|f| = f^*$

overall runtime $\Theta(f^* \cdot E)$ ⇒ b/c integer capacities

3. FF example

- find any path
- increase flow to min-capacity edge on path

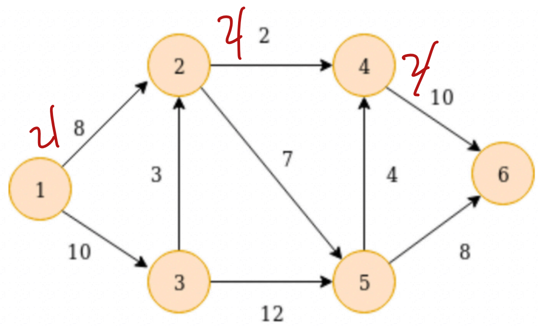


$|f| = 0$

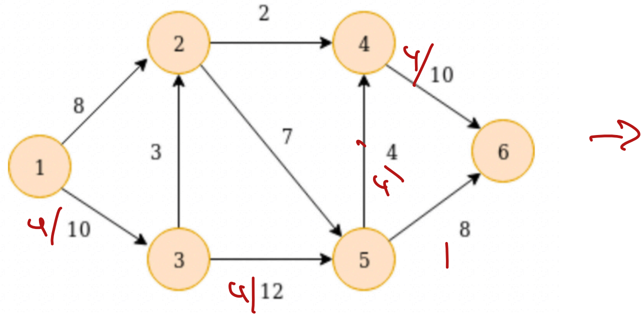
want:
|f|
and
f/c on each edge

(2) ⟨1, 3, 5, 4, 6⟩   (3) ⟨1, 3, 5, 6⟩

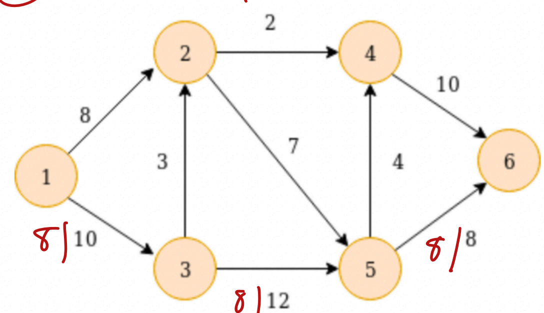
(1) ⟨1, 2, 4, 6⟩ → First Augmenting Path



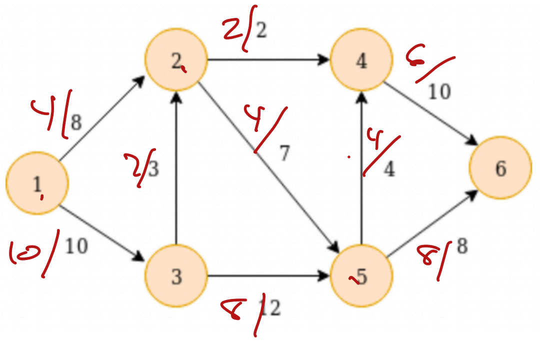
(2) <1, 3, 5, 4, 6> → First aug path



(3) <1, 3, 5, 6> → First aug path



one Final version → different aug. which aug paths we follow

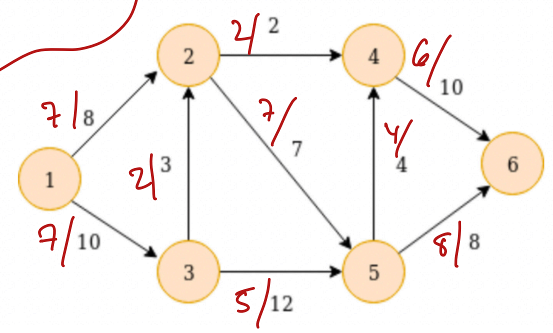


|f| = 14

Unusable leftover capacity: 4 + 1 + 3 + 4 + 4

= 16

Another final version!



unusable
leftover capacity:
$1+3+1+7+4 = 16$