More Network Flow

CSE 421 Winter 2023
Lecture 18

Announcements

We had to fix HW6 P3's statement a few times.

We think we've got it now (it says "version 3" in red on the pdf).

Holiday on Monday!

We'll have an announcement about office hours tonight.

You'll get exam solutions over the long weekend.

Closing The Loop

Feedback on homeworks

It's hard to interpret tone over text

- Especially when one person is in "technical statement mode" and the other isn't.
- And to you it's *your* work

I've asked TAs to be cognizant of that.

We're tweaking some things on our end to try to make it easier for TAs to give feedback

Late Policy

Hope was to be more granular; some it's working that way for, some it isn't.

Closing The Loop

We’re going to keep the Wednesday deadlines.

It’s chosen intentionally to work with sections

You wrap-up topic on Wednesday, start practicing new topic in the new homework and section on Thursday.

We’re very glad to hear sections are working!

If you haven’t been going, please consider starting.

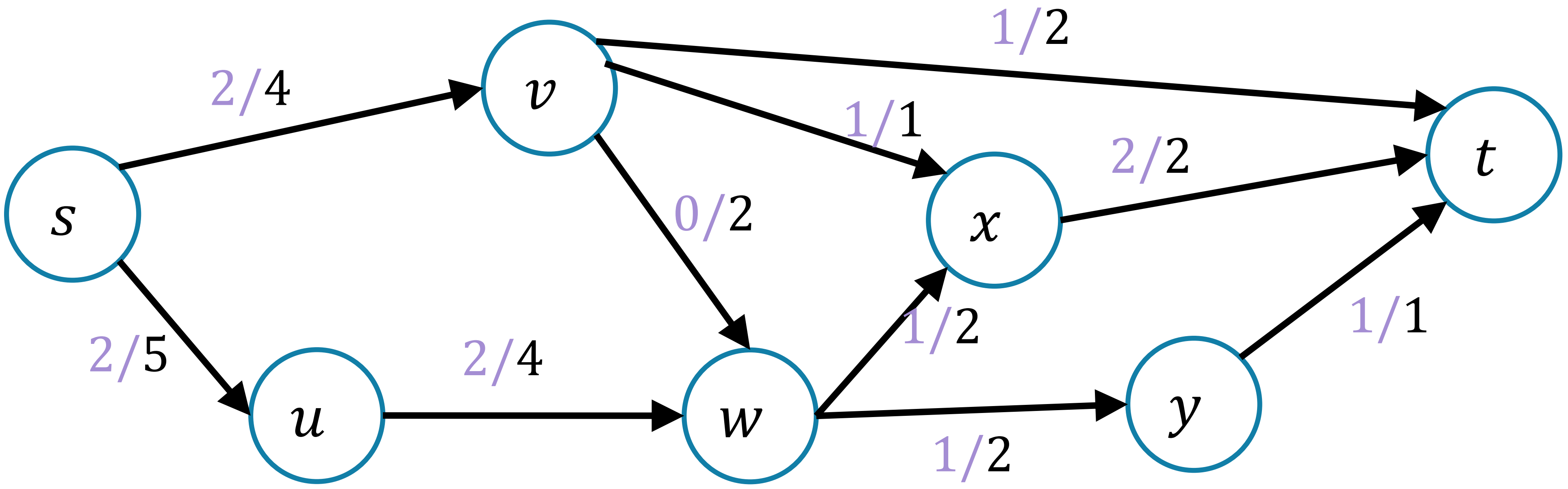
Especially if you are struggling with homework. It’s practice on similar problems!

Flows

A **flow** moves units of water from s to t.

Water can only be created at s and only disappear at t.

And you cannot move more water than the capacity on any edge.



What's a Cut?

For directed graphs (like we have here)

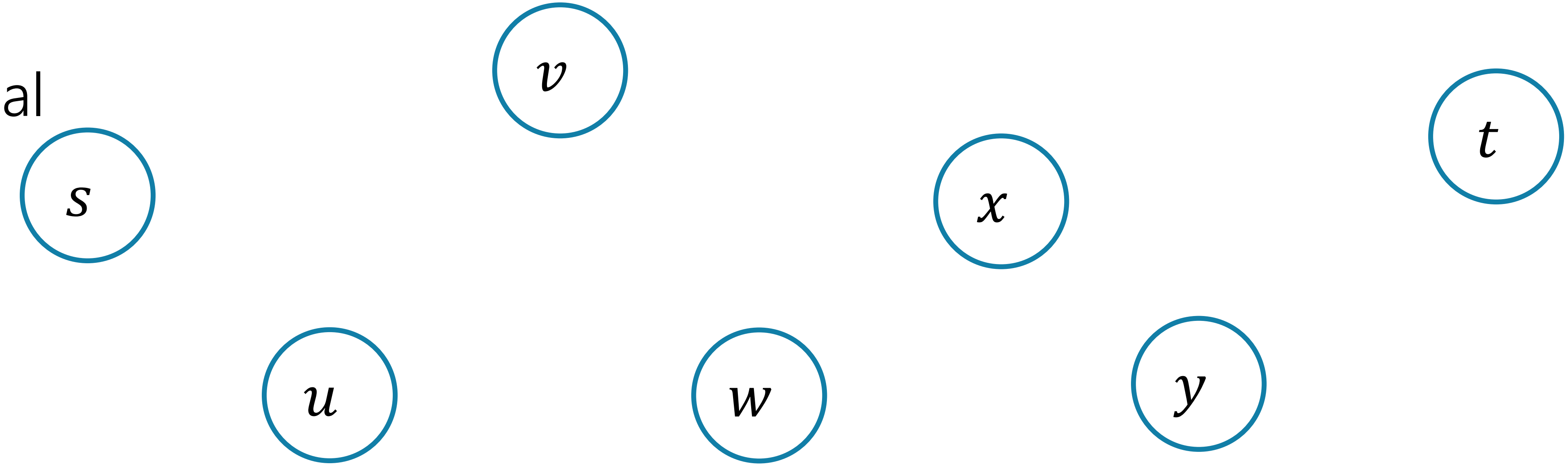
An (s, t)-cut, is a split of the vertices into two sets (A, B)

So that s is in A, t is in B,
and every other vertex is in exactly one of A and B.

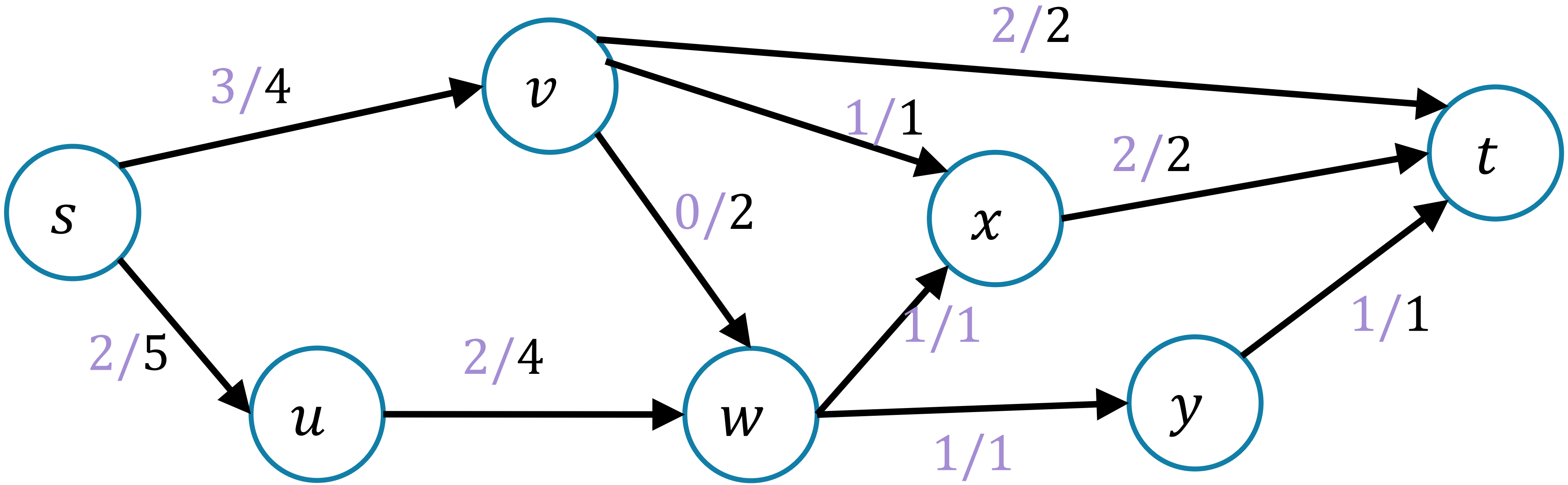
The capacity of a cut (or size of a cut) is the capacity of the edges going from A to B (don't count capacity from B to A).

Another Example

Residual



Flow



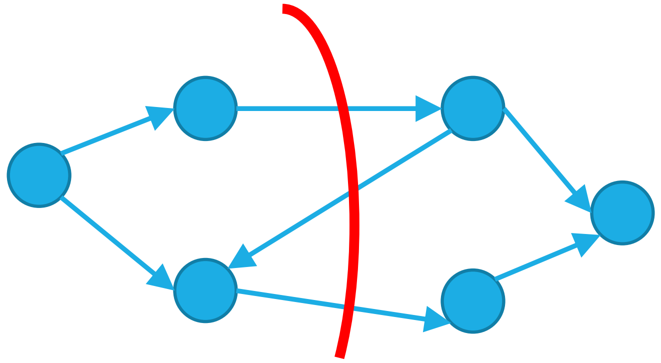
Max Flow-Min Cut Theorem

Max-Flow-Min-Cut Theorem

The value of the maximum flow from s to t is equal to the value of the minimum cut separating s and t.

The full proof is VERY notation heavy.
Focus on the words and intuition. The notation is there to support your intuition; the notation is not the main point.
We're going to skip a few steps for the sake of minimizing notation. See any textbook for all the details.

Some notation (more formally)



Let f be a flow.

For an edge e, $f(e)$ is the flow on e.

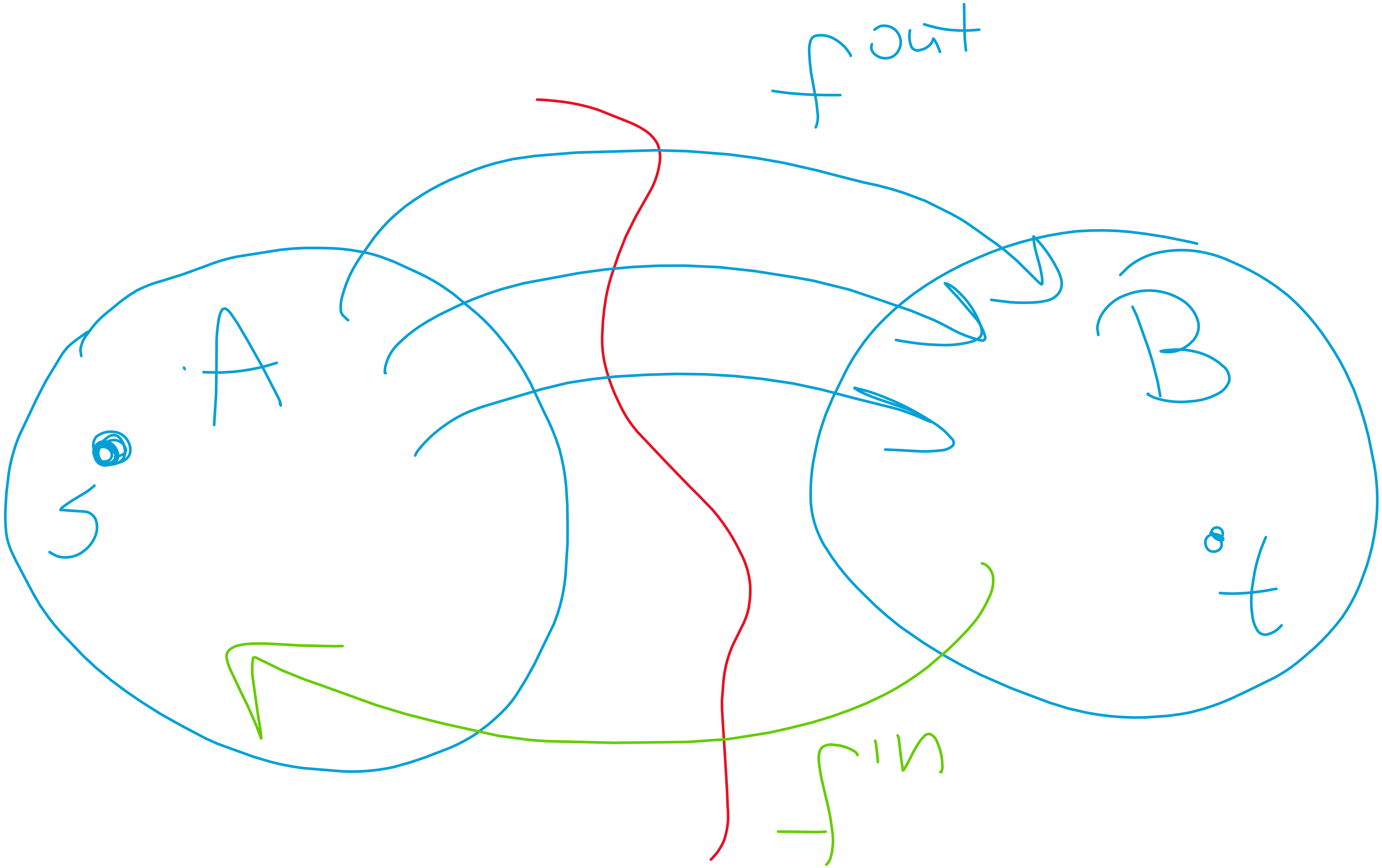
$\mathrm{val}(f)$ is the sum of flow leaving s (equivalently entering t).

For a cut (A, B), $\mathrm{cap}(A, B) = \sum_{e:e=(u,v),u\in A,v\in B} c(e)$

i.e., the sum of the capacities on edges going from A to B.

Direction matters!

Notice the capacity of a cut is independent of any particular flow. It's a property of the **original** graph, not the flow or the residual graph.



f out
A
s
B
t
f in

Step 1: The Flow Goes Somewhere

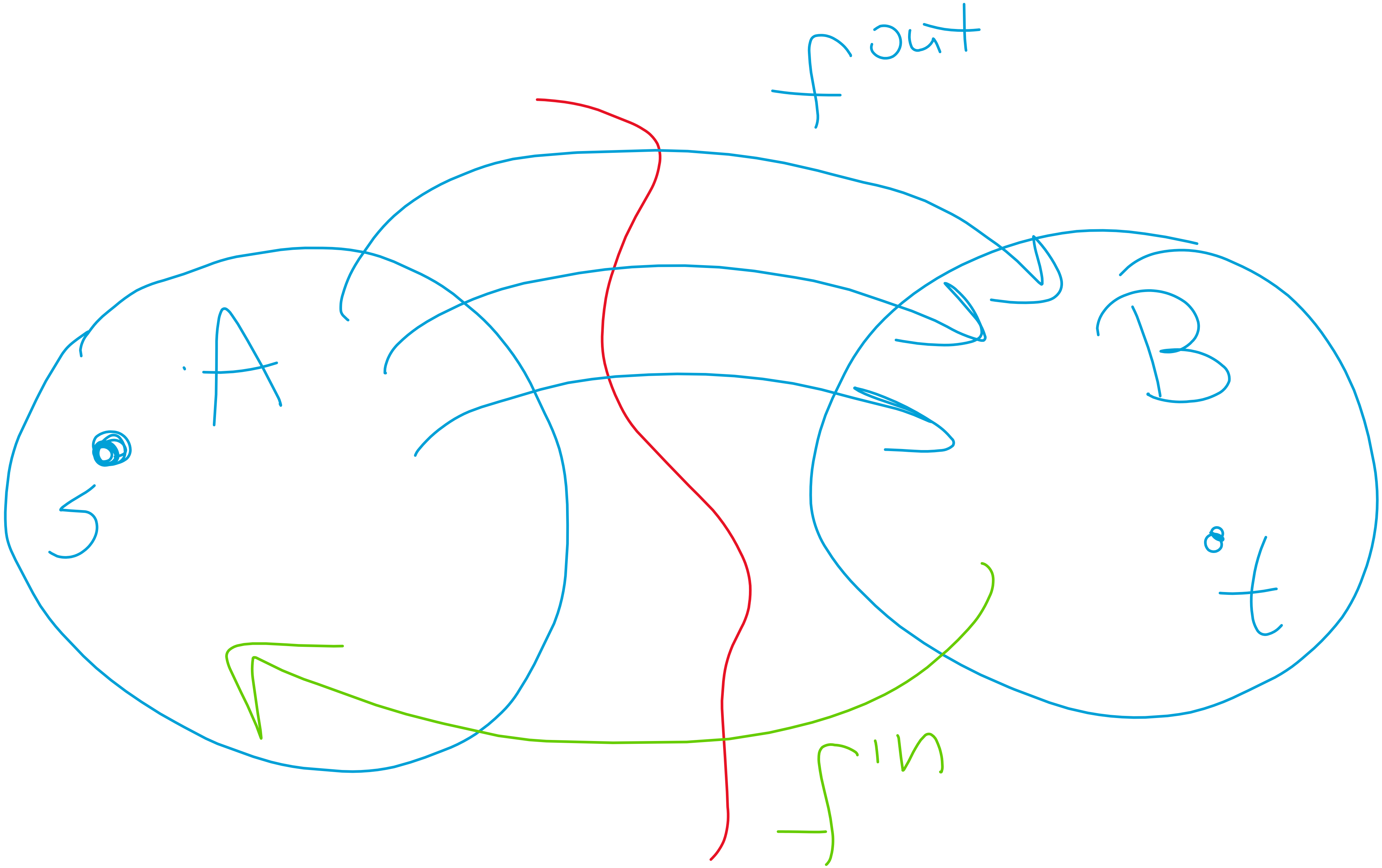
For every s-t cut, (A, B):

$$\mathrm{val}(f) = f^{out}(A) - f^{in}(A) = \sum_{e=(u,v):u\in A, v\in B} f(e) - \sum_{e=(v,u):u\in A, v\in B} f(e)$$

Intuitively, the net-flow for *every* cut is the same as the net flow for the cut $(s, V \setminus \{s\})$.

Why? Well the flow has to go somewhere! It can only disappear at t.

Why care? It's a technical observation we'll need later.



f out
A
s
B
t
f in

Step 2: Cuts limit flows ('weak duality')

Let f be any s-t flow, and (A, B) be any s-t cut.
Then $\text{val}(f) \leq \text{cap}(A, B)$

Cuts limit flows! Intuition: to get the flow to t it has to "all get through" every cut. So you can't have a flow of value more than any given cut.

Proof:

$$\text{val}(f) = f^{out}(A) - f^{in}(A) \leq f^{out}(A) = \sum_{e=(u,v):u\in A, v\in B} f(e)$$

$$\leq \sum_{e=(u,v):u\in A, v\in B} c(e) = \text{cap}(A, B)$$

Step 3: Cuts are the only things that limit flows

Let f^* be an s-t flow such that there is no s-t path in the residual graph.
Then there is a cut (A^*, B^*) such that $\mathrm{val}(f^*) = \mathrm{cap}(A^*, B^*)$

Intuition: going from $A^* \to B^*$, you're saturated; $B^* \to A^*$ is unused.

Sketch:

Let A^* be all the vertices reachable from s in the residual graph, and $B^* = V \setminus A^*$.

Observe that (A^*, B^*) is indeed an s-t cut. The only way to not be a cut is to have $t \in A^*$. But we assumed t was not reachable from s.

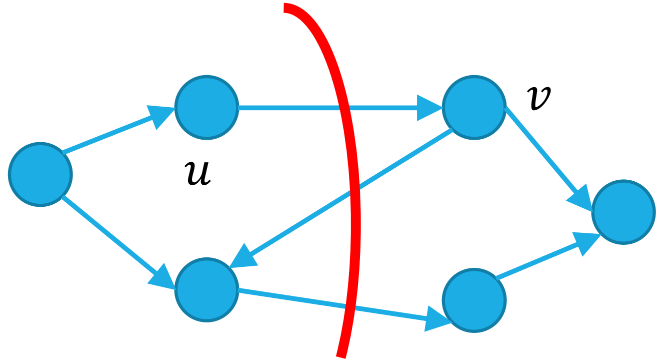
Step 3: Cuts are the only things that limit flows

Let f^* be an s-t flow such that there is no s-t path in the residual graph.
Then there is a cut (A^*, B^*) such that $\text{val}(f^*) = \text{cap}(A^*, B^*)$

(sub)-claim: If $e = (u, v)$ such that $u \in A^*$, $v \in B^*$, then $f(e) = c(e)$.

(i.e. edges going from A^* to B^* are saturated).

In the residual graph, we only don't have a copy of e if e is saturated. If we did have the edge e, we would be able to reach v from s, and it would be in A^*, not B^*. So e must be saturated.



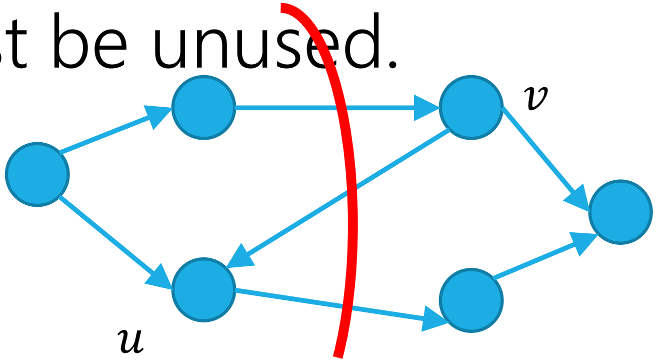
Step 3: Cuts are the only things that limit flows

Let f^* be an s-t flow such that there is no s-t path in the residual graph.
Then there is a cut (A^*, B^*) such that $\text{val}(f^*) = \text{cap}(A^*, B^*)$

(sub)-claim: If $e = (v, u)$ such that $u \in A^*$, $v \in B^*$, then $f(e) = 0$.

(i.e. edges going from B^* to A^* are unused).

In the residual graph, we add a copy of (u, v) when there is any flow on (v, u). If we did have the edge (u, v), we would be able to reach v from s, and it would be in A^*, not B^*. So $e = (v, u)$ must be unused.



Step 3: Cuts are the only things that limit flows

Let f^* be an s-t flow such that there is no s-t path in the residual graph.
Then there is a cut (A^*, B^*) such that $\text{val}(f^*) = \text{cap}(A^*, B^*)$

Put it together: What's the value of the flow?

$\text{val}(f^*) = f^{out}(A^*) - f^{in}(A^*)$	Step 1's lemma
$= \sum_{e=(u,v):u\in A,v\in B} f(e) - \sum_{e=(v,u):u\in A,v\in B} f(e)$	Net is flow out minus flow in.
$= \sum_{e=(u,v):u\in A,v\in B} c(e) - \sum_{e=(v,u):u\in A,v\in B} 0$	Last 2 slides
$= \text{cap}(A^*, B^*)$	Definition of capacity.

Concluding The Theorem

Max-Flow-Min-Cut Theorem

The value of the maximum flow from s to t is equal to the value of the minimum cut separating s and t.

Proof: Run Ford-Fulkerson, you'll get a flow of value f^* such that there is a cut of capacity f^*. There can be no larger flow and no smaller cut, as for all flows f and all cuts (A, B): $\mathrm{val}(f) \leq \mathrm{cap}(A, B)$.

Isn’t This Cool?

Another instance where we prove a big theorem using an algorithm.

The max-flow min-cut theorem doesn’t mention an algorithm, but it can be proved via analyzing Ford-Fulkerson!

So What?

Great quick check for if you've found the maximum flow (or min-cut).

Check the other and see if the value is the same!

We'll see examples of max-flow used for modeling. Sometimes the min-cut can be interpreted as a "barrier" to a good assignment.

It's also a nice example of **duality**

A maximization problem and a minimization problem that restrict each other.

Applications

Applications of Max-Flow-Min-Cut

Max-Flow and Min-Cut are useful if you work for the water company…

But they're also useful if you don't.

The most common application is assignment problems.

You have jobs and people who can do jobs – who is going to do which?

Big idea:

Let one unit of flow mean "assigning" one job to a person.

Hey Wait…

Isn’t this what stable matching is for?

Stable matching is very versatile, and it lets you encode preferences.

Max-flow assignment is even more versatile on the types of assignments.

But there’s not an easy way to encode preferences.

Example Problem

You and your housemates need to decide who is going to do each of the chores this week.

Some of your housemates are unable to do some chores.

Housemates: 1,2,3

Chores:

Arrange furniture, clean the **B**athroom, **C**ook dinner, do the **D**ishes

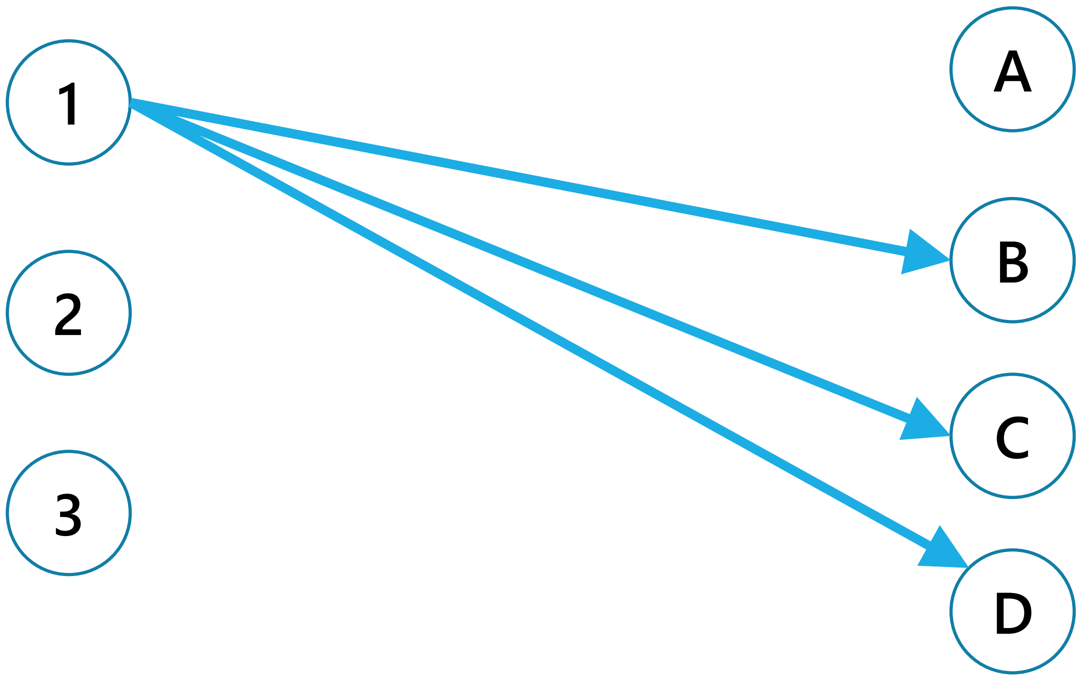
Housemate 1 is unable to arrange furniture, 2 is unable to cook.

Example Problem

Housemate 1 is unable to arrange furniture, 2 is unable to cook.

Vertex for each housemate and chore.

Edge if the housemate **could** do the chore

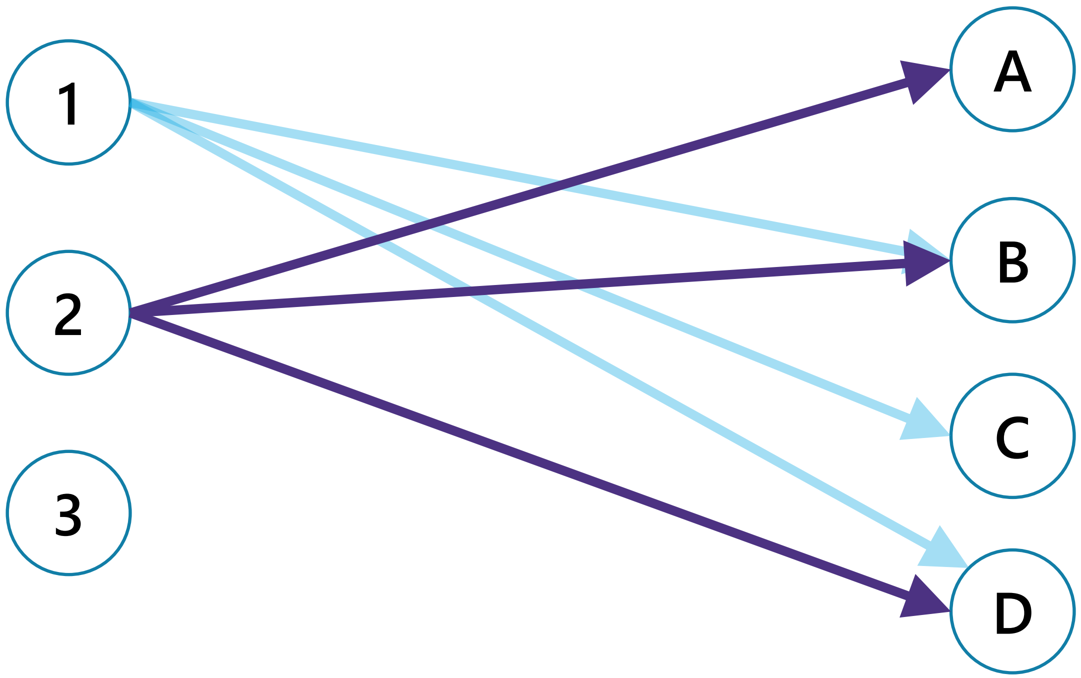


Example Problem

Housemate 1 is unable to arrange furniture, 2 is unable to cook.

Vertex for each housemate and chore.

Edge if the housemate **could** do the chore

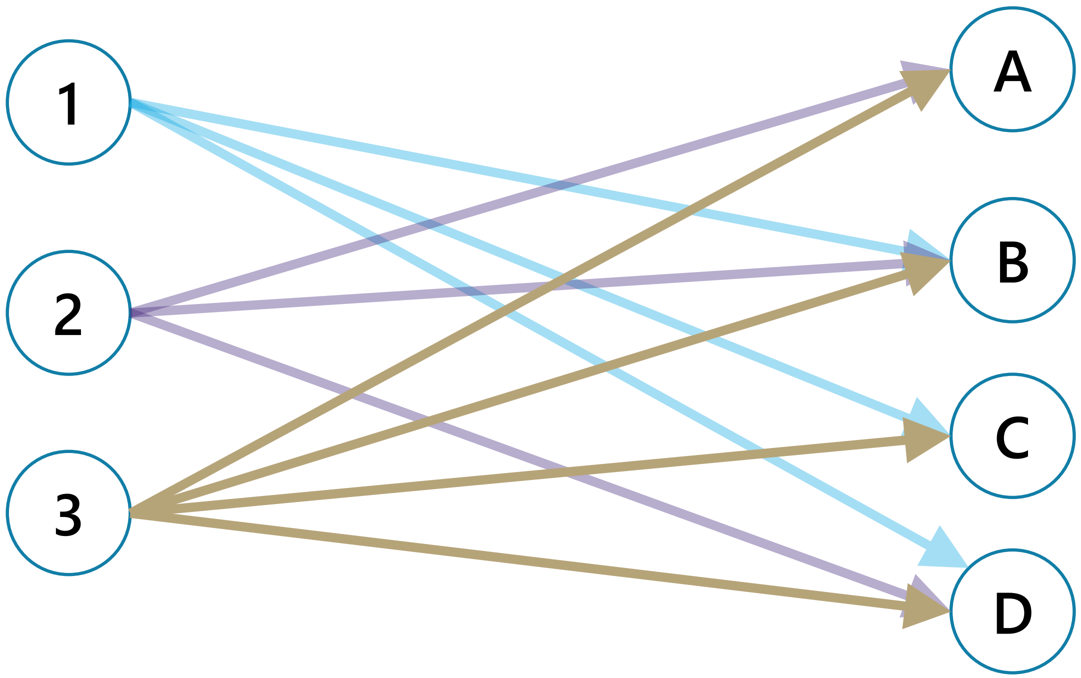


Example Problem

Housemate 1 is unable to arrange furniture, 2 is unable to cook.

Vertex for each housemate and chore.

Edge if the housemate **could** do the chore

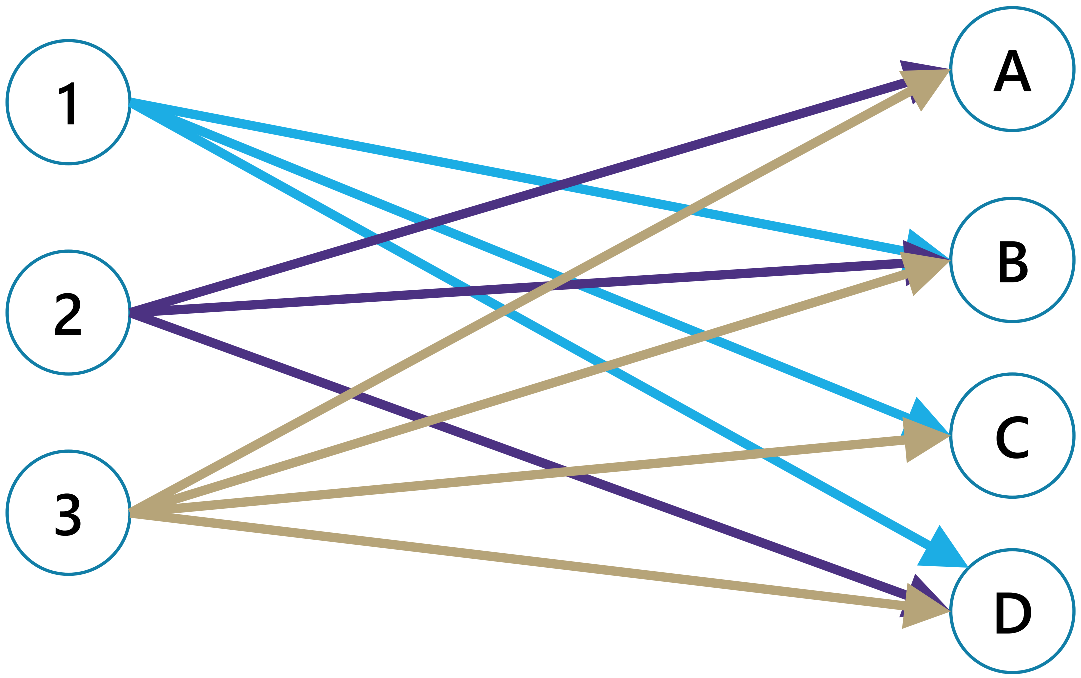


Example Problem

Housemate 1 is unable to arrange furniture, 2 is unable to cook.

Vertex for each housemate and chore.

Edge if the housemate **could** do the chore

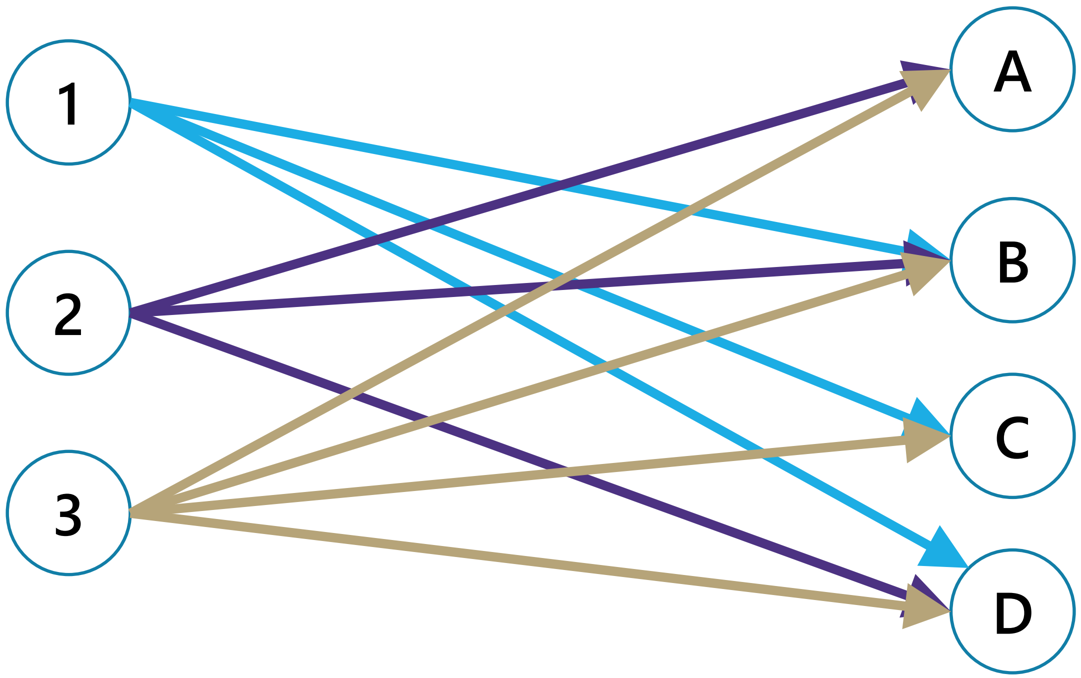


Example Problem

Idea: Flow from 1 to B means "make housemate 1 do chore B."

Every chore needs to be done (by one person).

Every person needs to do at most two chores.

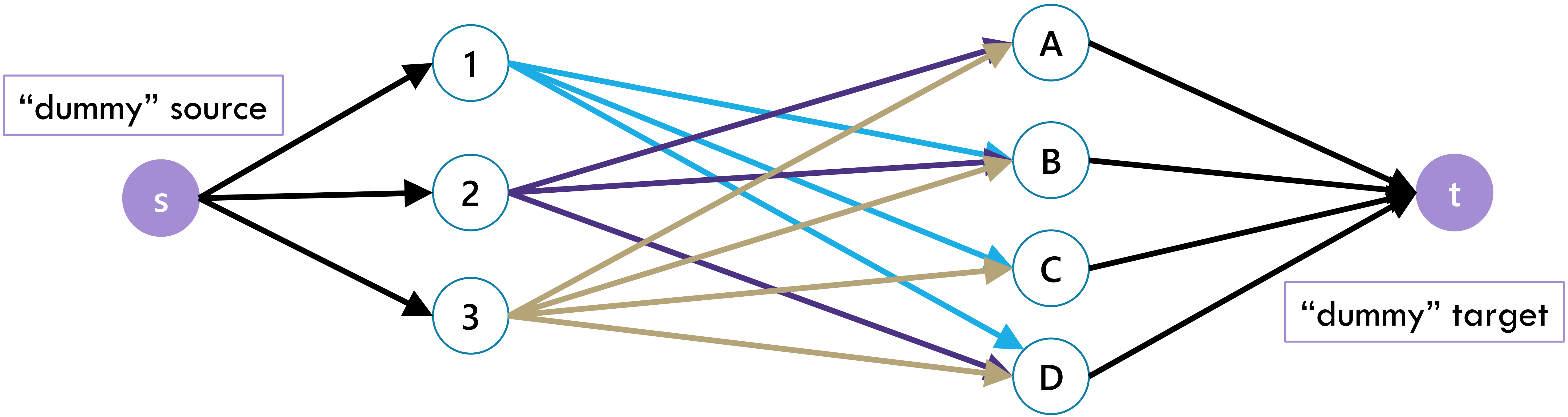


Example Problem

Idea: Flow from 1 to B means “make housemate 1 do chore B.”

Every chore needs to be done (by one person).

Every person needs to do at most two chores.

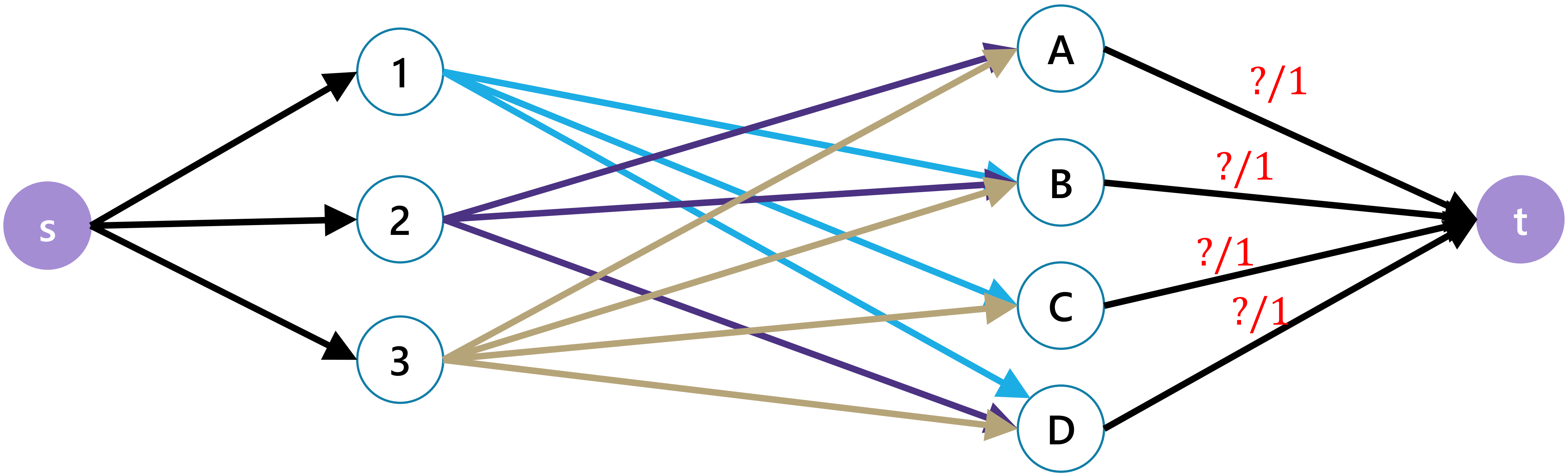


Example Problem

Idea: Flow from 1 to B means "make housemate 1 do chore B."

Every chore needs to be done (by one person).

Every person needs to do at most two chores.

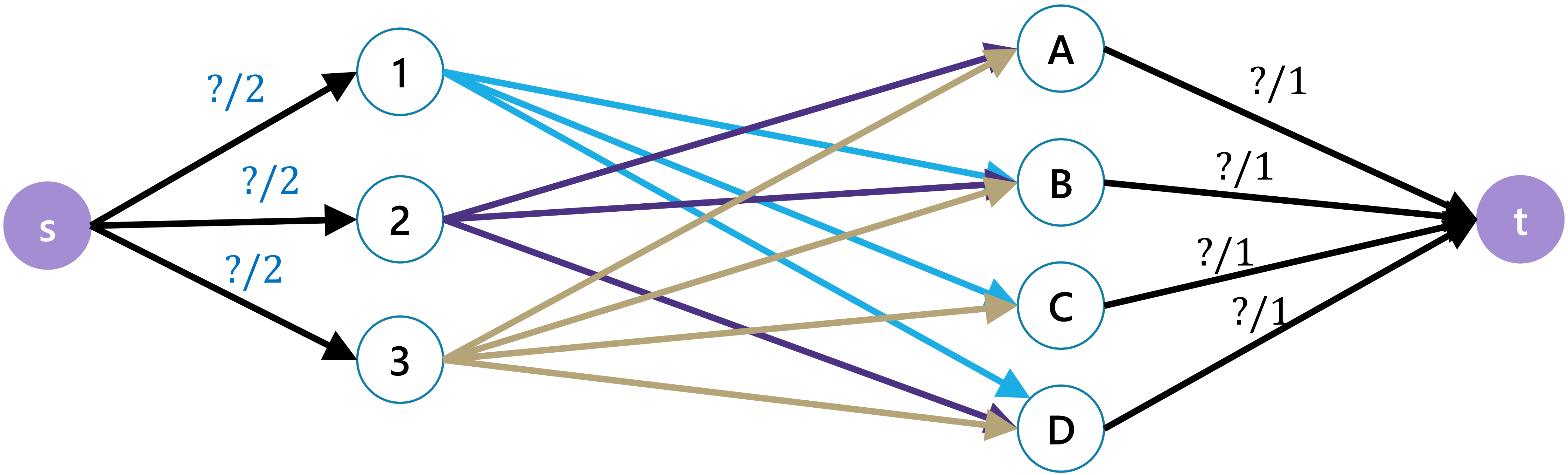


Example Problem

Idea: Flow from 1 to B means "make housemate 1 do chore B."

Every chore needs to be done (by one person).

Every person needs to do at most two chores.

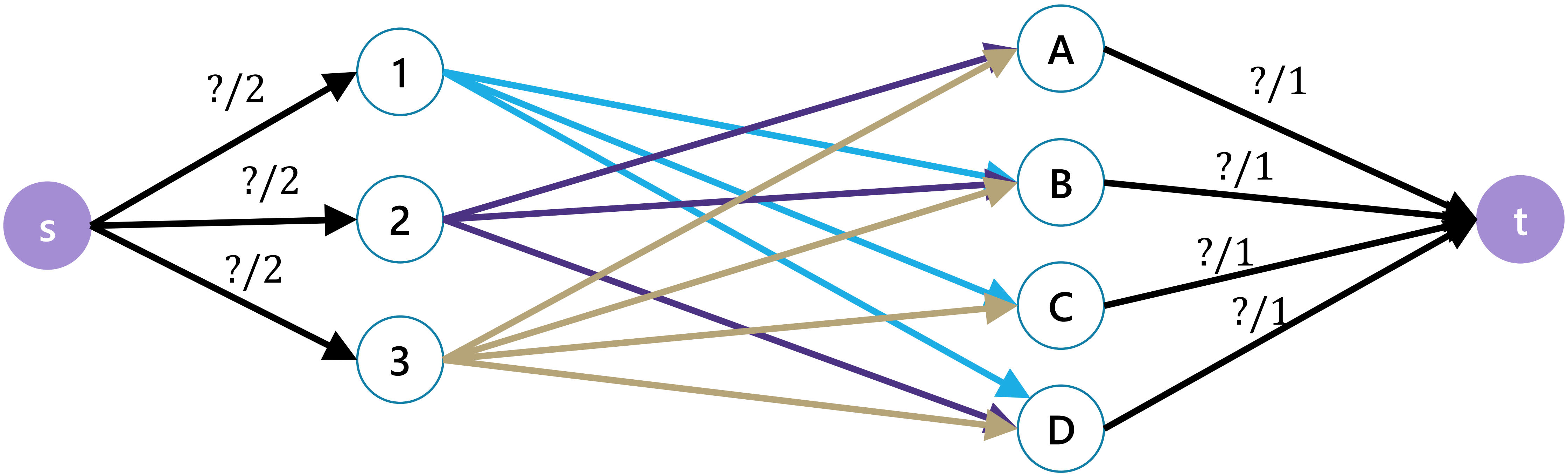


Example Problem

What are the capacities for the middle edges?

Could make them 1 (make sure you don't get "two units of cooking"

All our requirements are already (implicitly) encoded. So could make them ∞ instead.

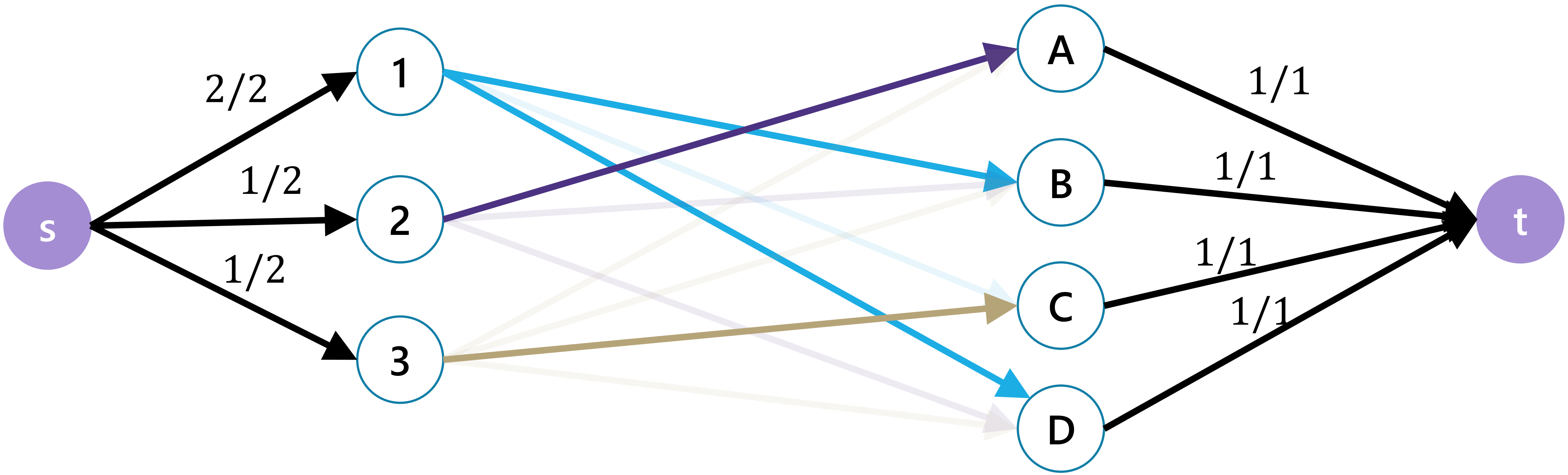


Example Problem

Find a max flow…And read off the assignment!

Full color: 1 unit of flow, faded: 0 units of flow

1 cleans the bathroom and does the dishes, 2 arranges furniture, 3 cooks.



Why are all of our constraints met?

Every chore gets done

No one does more than 2 chores

People only do chores they’re capable of

Why are all of our constraints met?

Every chore gets done

A flow of value 4 sends one unit of flow through each of A,B,C,D (because the edges to t are all capacity 1), so a max-flow ensures if possible we'll find an assignment.

No one does more than 2 chores

Only 2 units of flow can go through any person vertex (because edges from s to people are all capacity 2).

People only do chores they're capable of

There is only an edge from a person to a chore if they can do that chore.

One More Requirement…

There's another requirement we haven't mentioned:

People only get "whole units" of chores

i.e. you don't have two people each doing half of the cooking.

The max-flow approach guarantees this! As long as our requirements are integers (or ∞) as well.

Same logic as last lecture – Ford-Fulkerson will only add integers to the current flow.

Key Ideas

Use different vertices to represent different jobs (even if they look related).

You can (once per problem) use the value of the flow to evaluate whether you've met some minimum number (one "at least" or "exactly equal to" requirement)

Otherwise, use capacities to limit the options. One unit of flow usually represents one "assignment."

More Practice

Another Problem

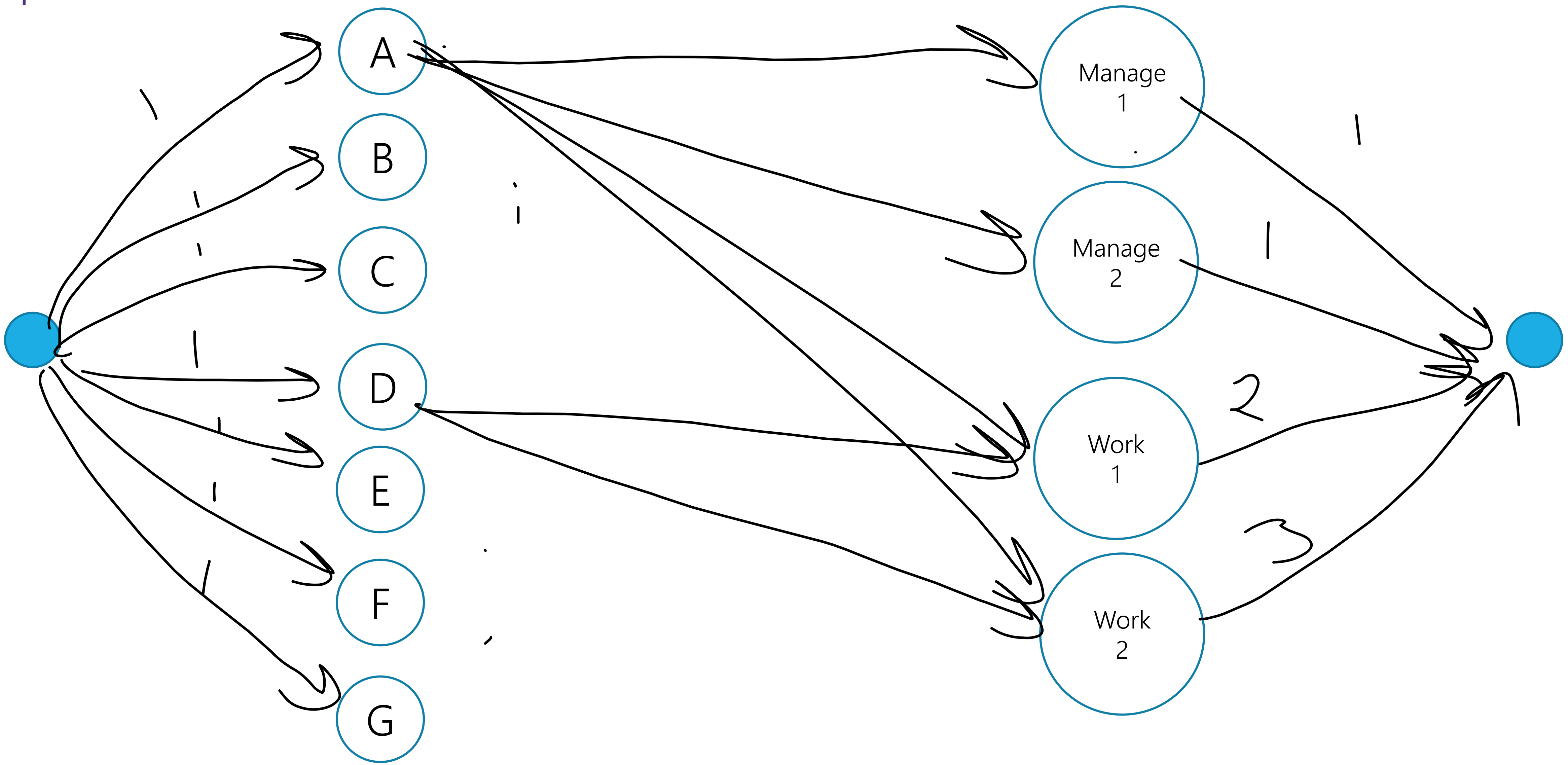
You run two coffee shops. You have to decide who will work at which of your shops today:

A, B, C are all capable of managing a shop.

D, E, F, G are all regular employees (can't be a manager)

You need at least one manager at each shop, at least 3 people (total) at shop 1 and at least 4 people (total) at shop 2.

Hint: think of assigning managers and non-managers as separate…


A
B
C
D
E
F
G
Manage 1
Manage 2
Work 1
Work 2

Key Ideas

Use different vertices to represent different jobs (even if they look related).

You can (once per problem) use the value of the flow to evaluate whether you've met some minimum number (one "at least" or "exactly equal to" requirement)

Otherwise, use capacities to limit the options. One unit of flow usually represents one "assignment."

Baseball Elimination

One More Example

A classic example

We’ll also be able to use the min-cut in addition to the flow!

Question: Can the Mariners still win* the division?

*or at least tie for first place.

And if they can’t, can you explain why.

Can The Mariners Win The Division?

It's late at night September 14, 1998.
You're working for the Seattle Times.
The Mariners won! But the Angels did too.
How do you frame the Mariners current situation in your postgame article?

Team	Wins (w)	Games Left
Angels	81	12
Rangers	80	12
Mariners	70	12
A's	69	12

MLB rules say all games will be played (if they end up mattering) so you can assume those will happen.

Can The Mariners Win The Division?

Team	Wins (w)	Games Left	Possible Wins (P)
Angels	81	12	93
Rangers	80	12	92
Mariners	70	12	82
A’s	69	12	81

$P_{MARINERS} \geq w_i$ for all i, so the Mariners can win the division, right?

Well…No

The teams will play each other, here are the number of games to be played against each other.

g_{ij}	Angels	Rangers	Mariners	A's
Angels	-	5	3	4
Rangers	5	-	4	3
Mariners	3	4	-	5
A's	4	3	5	-

Team	Wins (w)	Games Left	Possible Wins (P)
Angels	81	12	93
Rangers	80	12	92
Mariners	70	12	82
A's	69	12	81

Well…No

At least one of the Angels and Rangers is going to win at least 83 games

someone wins at least three of the five they play against each other.

The Mariners can only win 82 games.

Lessons

Comparing P_i to w_j is insufficient to tell if a team is eliminated.

The teams are interconnected by the games they play against each other.

Let's find a way to do this calculation…not by hand.

What do we need to assign?

Assignment

We need to assign who wins each of the remaining games.

Safe to assume the Mariners will win them all.

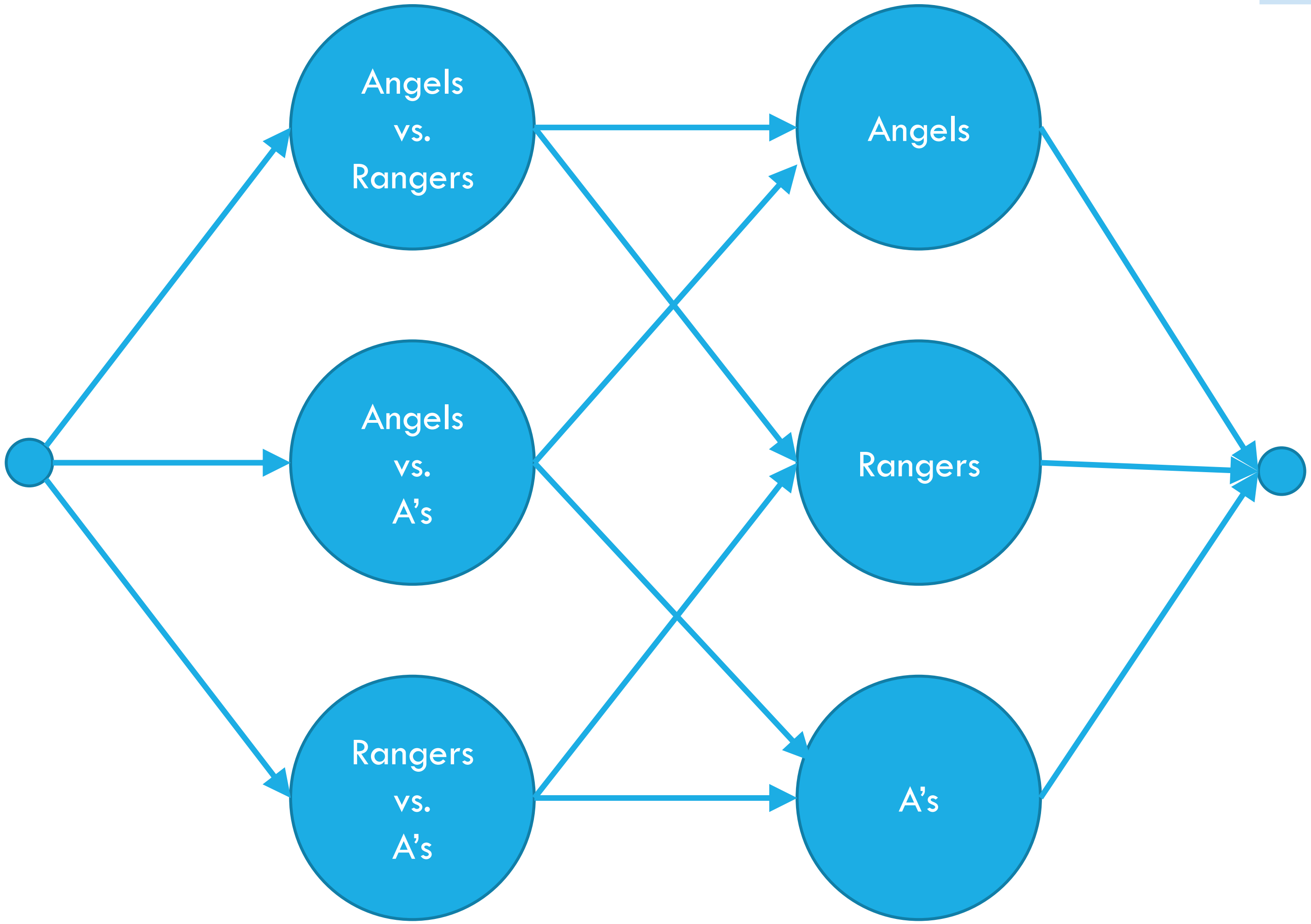
Just need to figure out the others.

One unit of flow represents one win.

Making a Network

	Angels	Rangers	Mariners	A's
Angels	-	5	3	4
Rangers	5	-	4	3
Mariners	3	4	-	5
A's	4	3	5	-

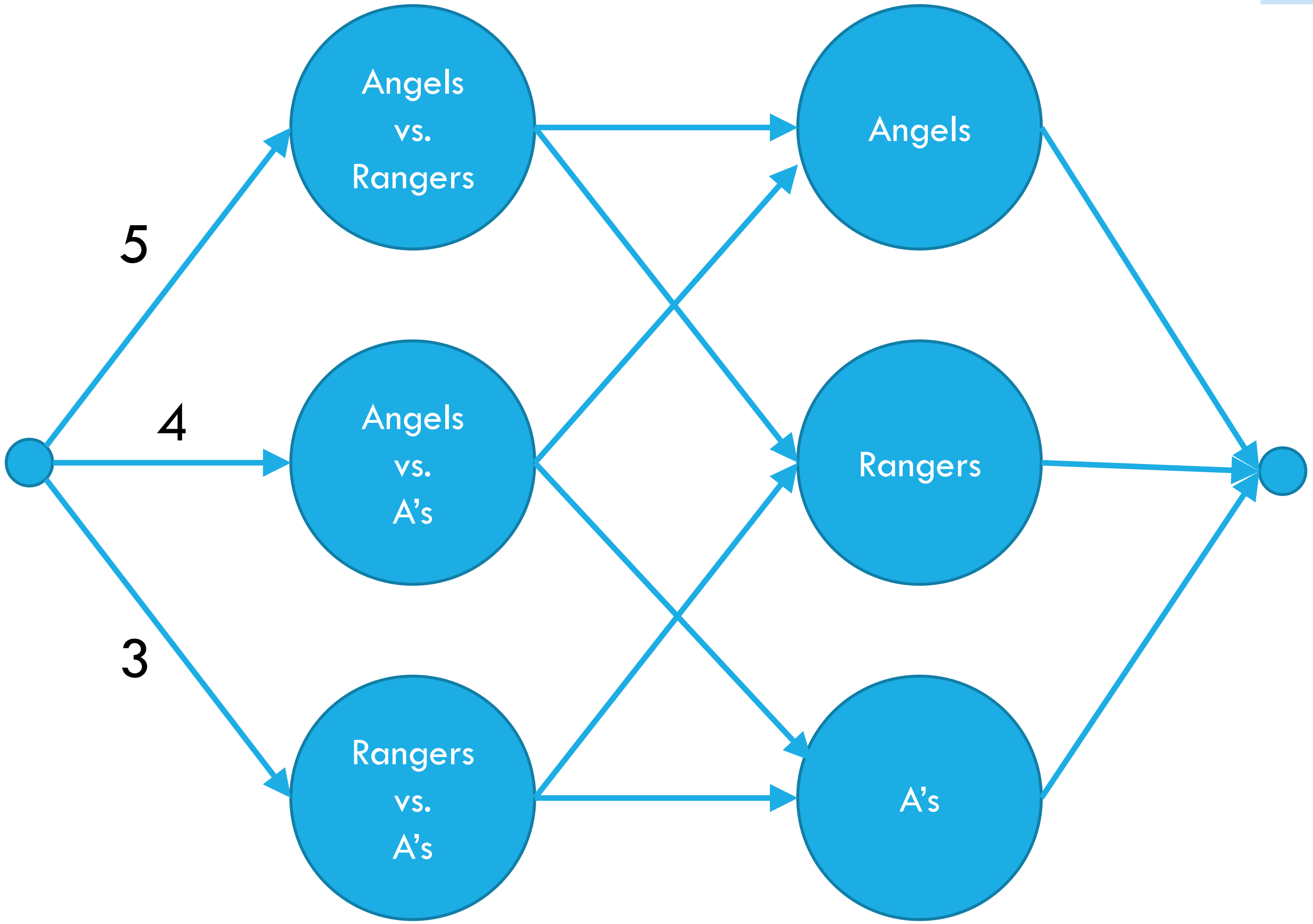
Team	Wins (w)	Possible Wins (P)
Angels	81	93
Rangers	80	92
Mariners	70	82
A's	69	81



s, t on the ends
First layer is pairs of opponents (i.e. what game is being played)
Second layer is individual teams.

Making a Network

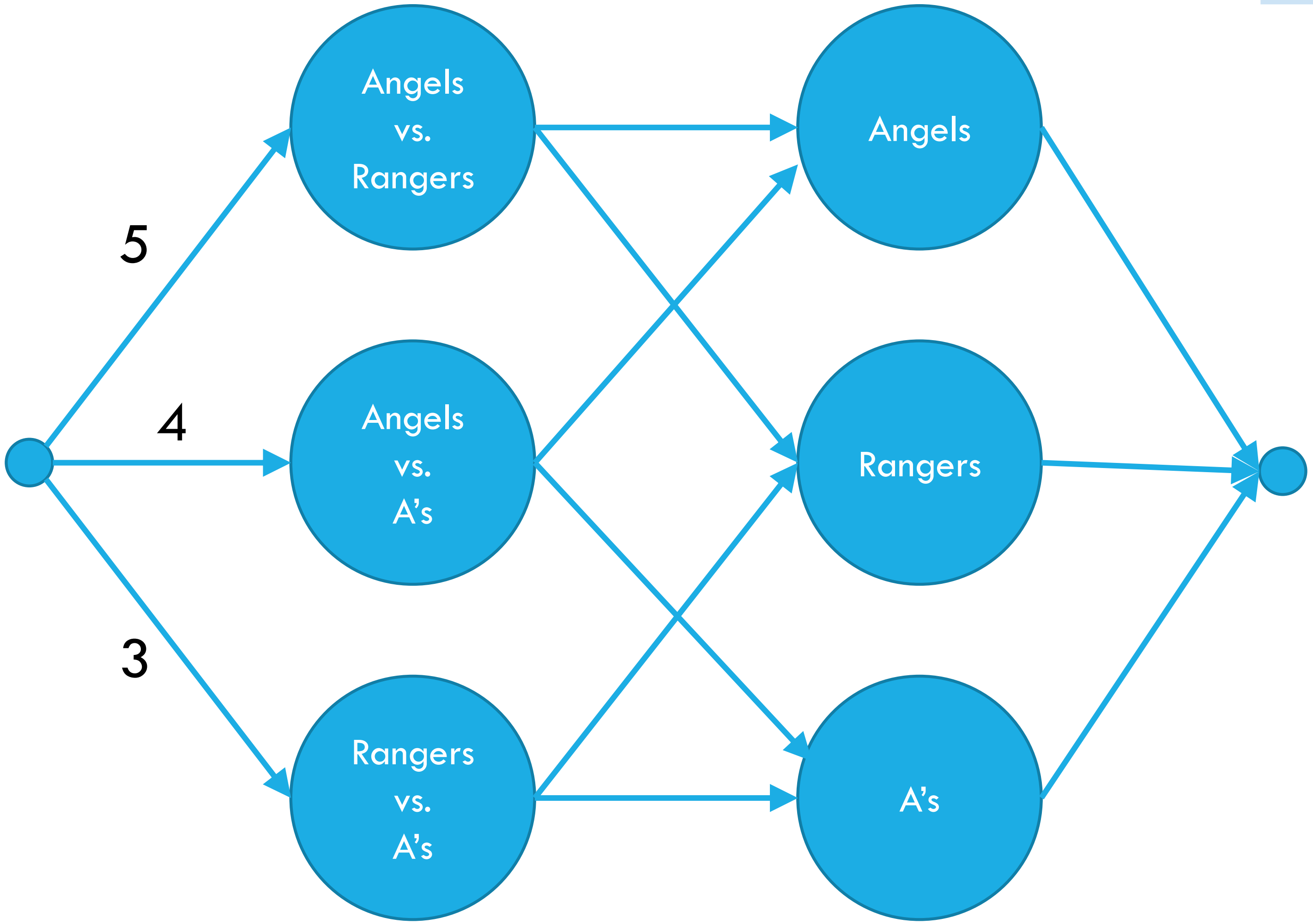
	Angels	Rangers	Mariners	A's
Angels	-	5	3	4
Rangers	5	-	4	3
Mariners	3	4	-	5
A's	4	3	5	-



Team	Wins (w)	Possible Wins (P)
Angels	81	93
Rangers	80	92
Mariners	70	82
A's	69	81

Put number of games to be played from s to pairs

Making a Network



	Angels	Rangers	Mariners	A's
Angels	-	5	3	4
Rangers	5	-	4	3
Mariners	3	4	-	5
A's	4	3	5	-

Team	Wins (w)	Possible Wins (P)
Angels	81	93
Rangers	80	92
Mariners	70	82
A's	69	81

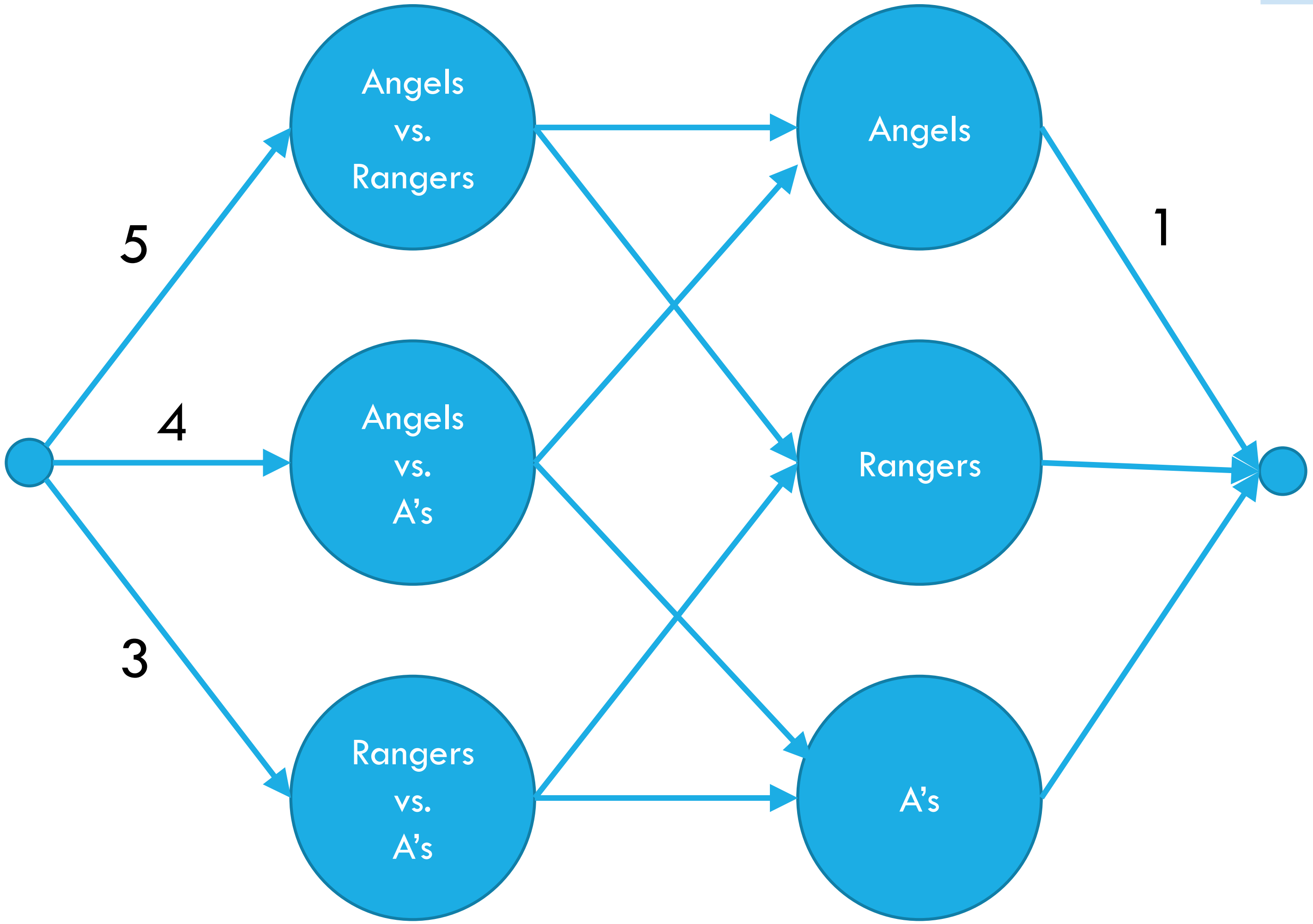
How do we make sure Mariners win? They'll end the season with 82 wins (current + games left).

How many more can each team win?

Mariners poss total – team current

Making a Network

Angels have 81 wins, 1 more is ok (total matches Mariners possible) 2 is not. Capacity is 1.



	Angels	Rangers	Mariners	A's
Angels	-	5	3	4
Rangers	5	-	4	3
Mariners	3	4	-	5
A's	4	3	5	-

Team	Wins (w)	Possible Wins (P)
Angels	81	93
Rangers	80	92
Mariners	70	82
A's	69	81

How do we make sure Mariners win? They'll end the season with 82 wins (current + games left).

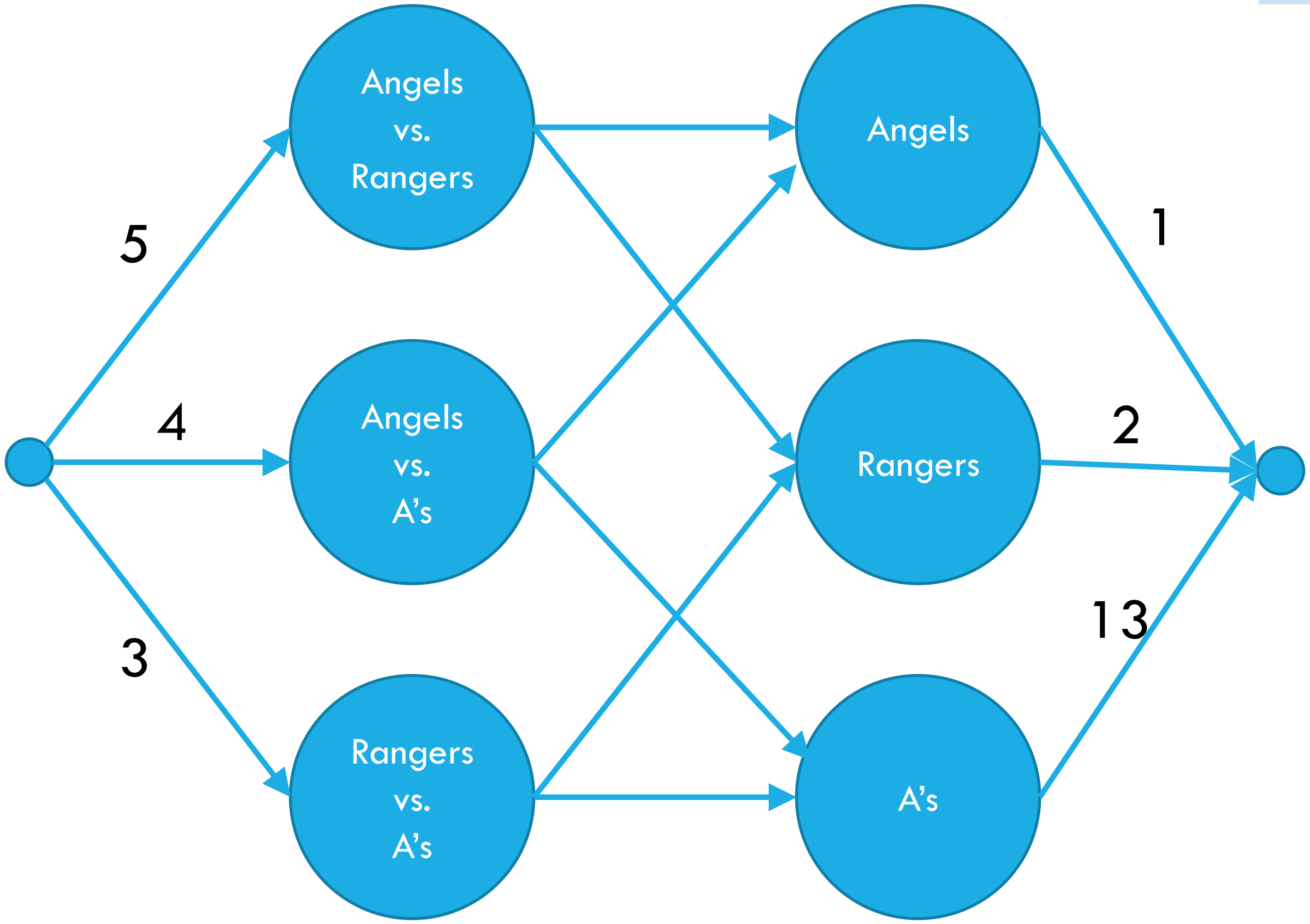
How many more can each team win?

Mariners poss total – team current

Making a Network

	Angels	Rangers	Mariners	A's
Angels	-	5	3	4
Rangers	5	-	4	3
Mariners	3	4	-	5
A's	4	3	5	-

Team	Wins (w)	Possible Wins (P)
Angels	81	93
Rangers	80	92
Mariners	70	82
A's	69	81



How do we make sure Mariners win? They'll end the season with 82 wins (current + games left).

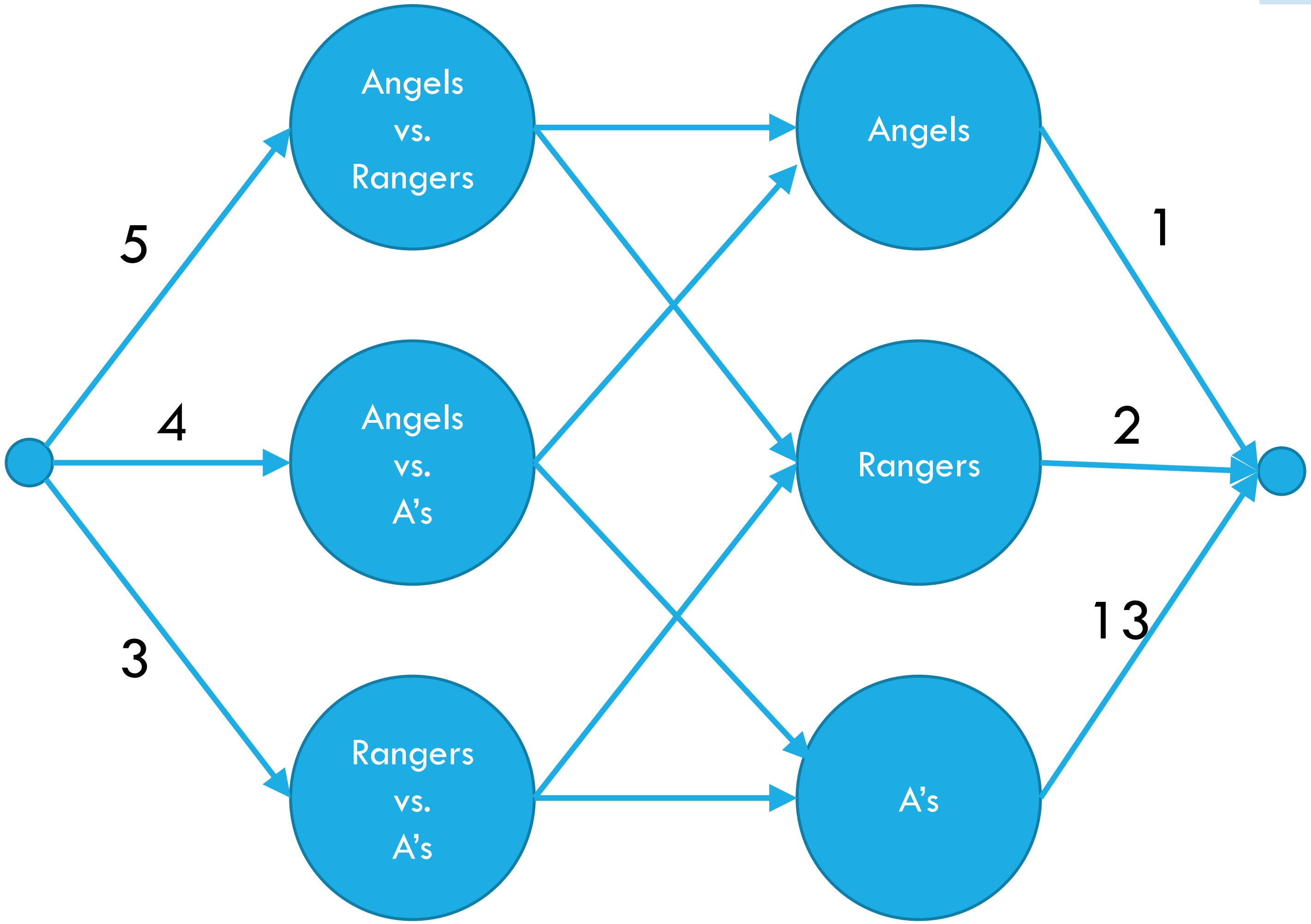
How many more can each team win?

Mariners poss total – team current

Making a Network

	Angels	Rangers	Mariners	A's
Angels	-	5	3	4
Rangers	5	-	4	3
Mariners	3	4	-	5
A's	4	3	5	-

Team	Wins (w)	Possible Wins (P)
Angels	81	93
Rangers	80	92
Mariners	70	82
A's	69	81



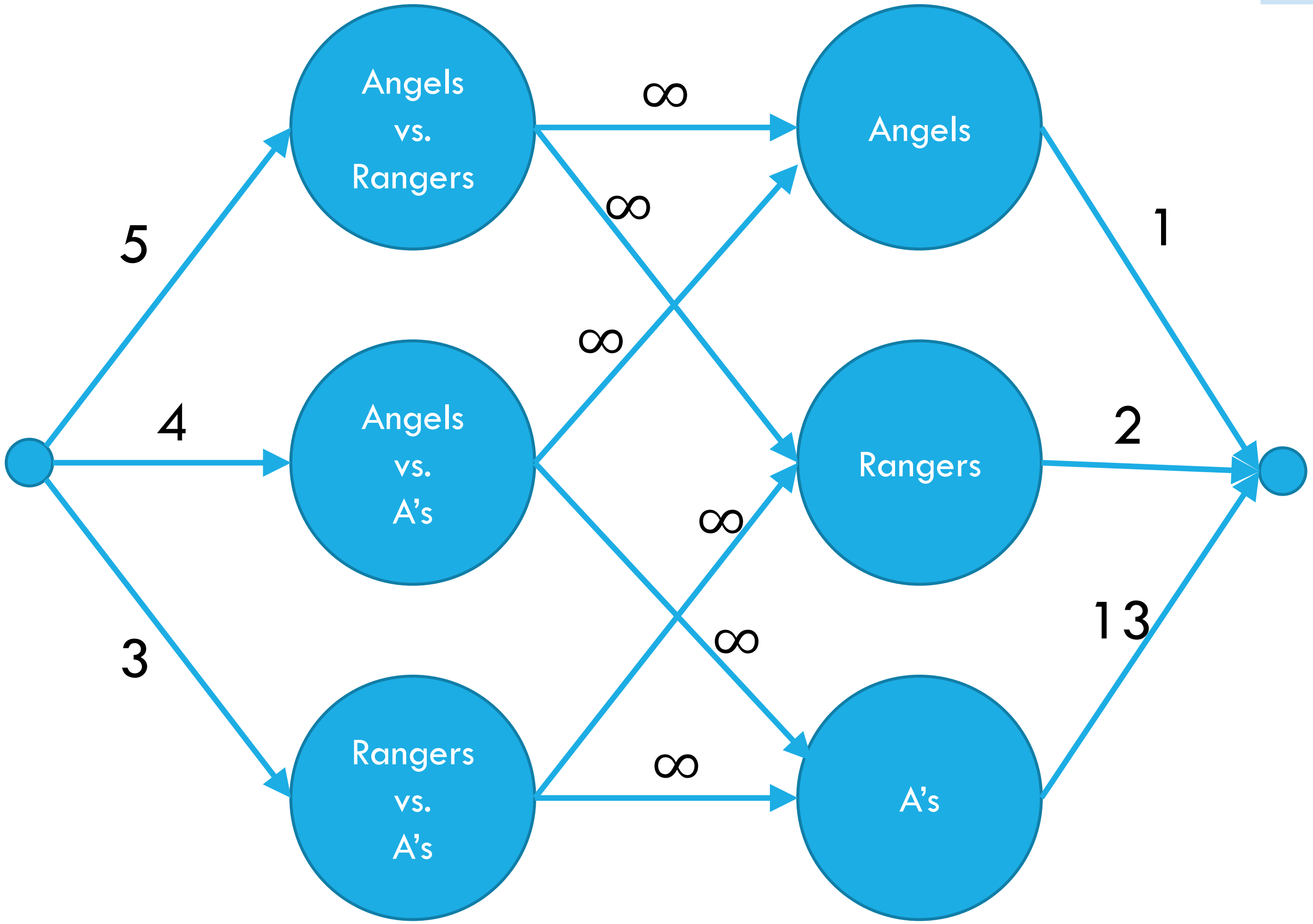
Edges in the middle?
Only to the two teams playing.

We've handled are constraints, can leave capacities at ∞.

Making a Network

	Angels	Rangers	Mariners	A's
Angels	-	5	3	4
Rangers	5	-	4	3
Mariners	3	4	-	5
A's	4	3	5	-

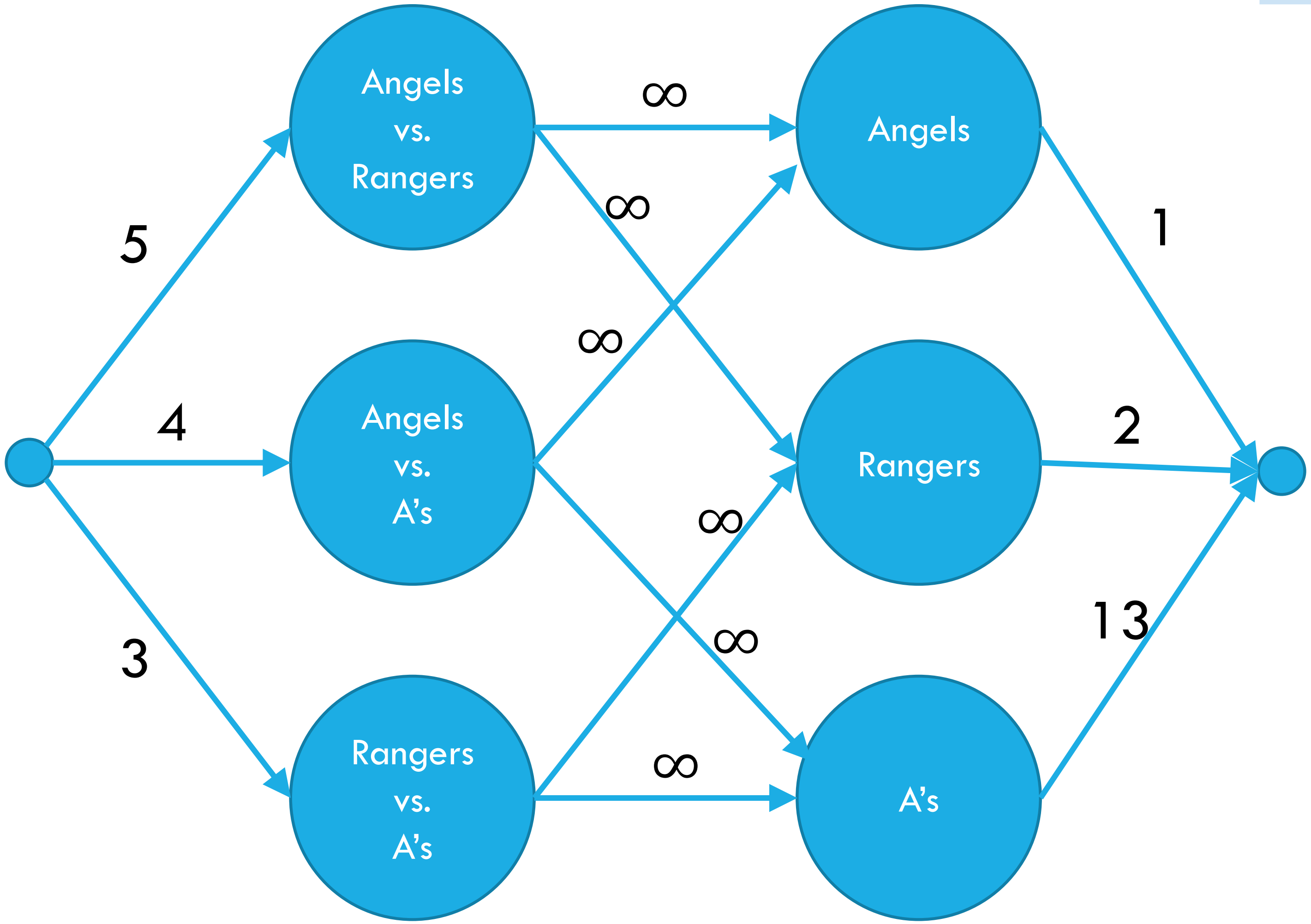
Team	Wins (w)	Possible Wins (P)
Angels	81	93
Rangers	80	92
Mariners	70	82
A's	69	81



Edges in the middle?
Only to the two teams playing.

We've handled are constraints, can leave capacities at ∞.

Making a Network



	Angels	Rangers	Mariners	A's
Angels	-	5	3	4
Rangers	5	-	4	3
Mariners	3	4	-	5
A's	4	3	5	-

Team	Wins (w)	Possible Wins (P)
Angels	81	93
Rangers	80	92
Mariners	70	82
A's	69	81

We're done!

Why are all the constraints met?

How many games are there to play? Equal to the capacities leaving s.

So if we have a flow of at least that value, we'll assign winners to all the games.

Why will the Mariners win with this assignment?

The capacity from team A to t ensures A will not end with more wins.

No "half-wins" or anything weird?

All capacities are integers, so we'll get an integer solution!

Interpreting the answer

If the max flow has value equal to number of games, we know how the Mariners can still win the division.

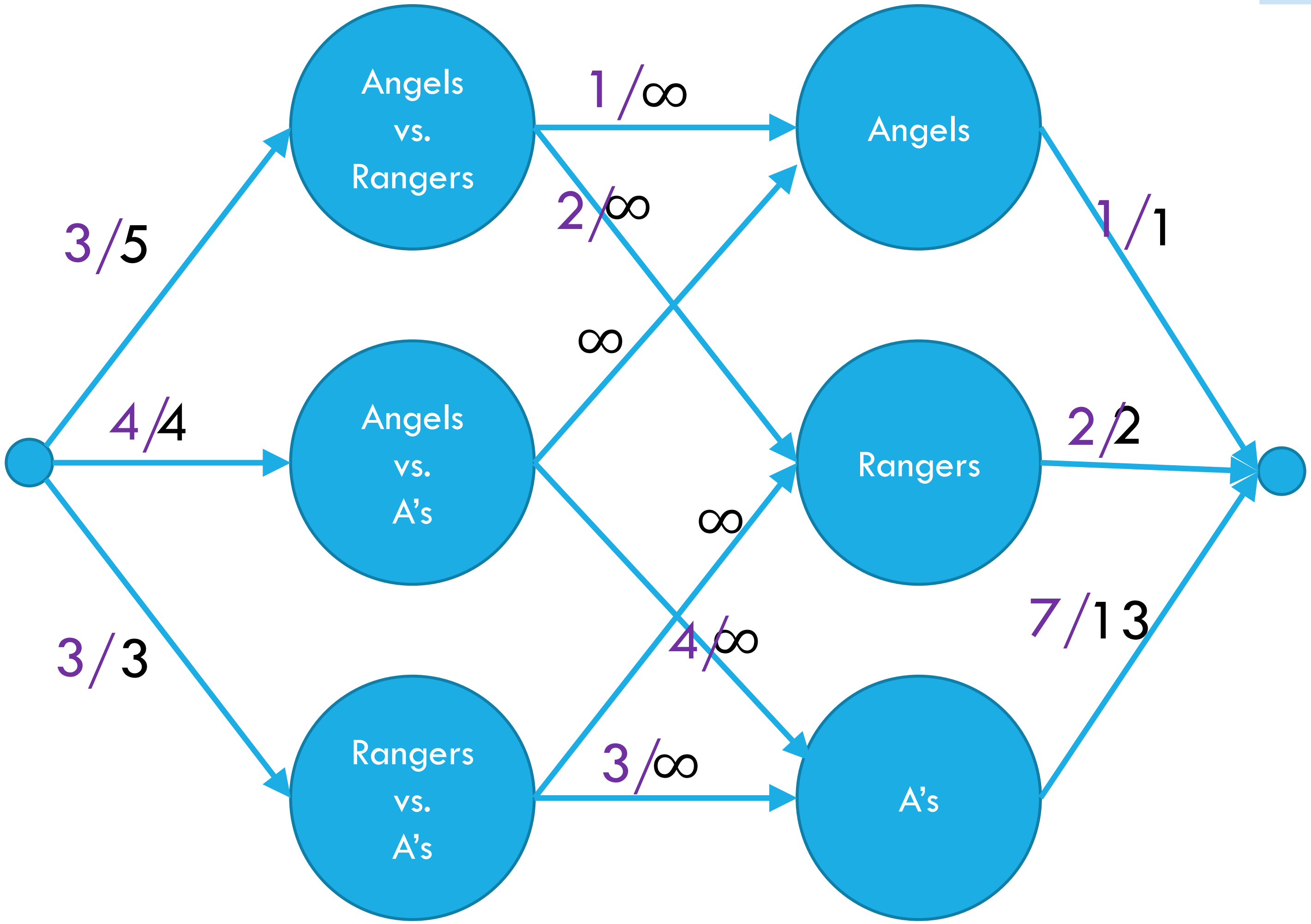
If the max flow is less than that, the Mariners can't win the division!

(if they could win the division, then there is a way that the remaining games could play out with the mariners having as many wins as anyone else, but then we could make a feasible flow by assigning a unit of flow for each winner).

Max Flow

	Angels	Rangers	Mariners	A's
Angels	-	5	3	4
Rangers	5	-	4	3
Mariners	3	4	-	5
A's	4	3	5	-

Team	Wins (w)	Possible Wins (P)
Angels	81	93
Rangers	80	92
Mariners	70	82
A's	69	81



This is the maximum flow. What's the min-cut?

{s, Angels vs. Rangers, Angels, Rangers} is one side of the cut.

The Angels and Rangers were enough to prove that the Mariners couldn't win!

Generating Proof that you're eliminated

How do you describe to the general public that the Mariners are eliminated.

People are going to say "the Mariners can still win 82 games, no one has one 82, it's not over yet!"

Of the Angels and Rangers, they will win (combined) at least

$81 + 80 + 5$ games (Angels wins, Rangers wins, games to be played among these teams)

On average they win $\frac{166}{2} = 83$ games. That's more than 82. Someone is beating that average, and whoever that is the Mariners won't catch them.

In General

Find the max flow. If its value is the number of games remaining, great!

Mariners can still win.

If its value is less than that, find the min cut. The set of all teams reachable from s in the residual graph will show you **why** the Mariners are eliminated.

Takeaways

If you want to “assign” things, max-flow might be a good option.

If you say “at most” you can probably just make a capacity constraint

Once you can do an “exactly equal” or “at most” by checking the value of the max-flow.

Sometimes you want an extra layer or two if you have a multiple types of assignments.

Sometimes you can convert an “at least” in one group into an “at most” on another group.

Optional – Why is there always an explanation?

An Explanation Always Exists

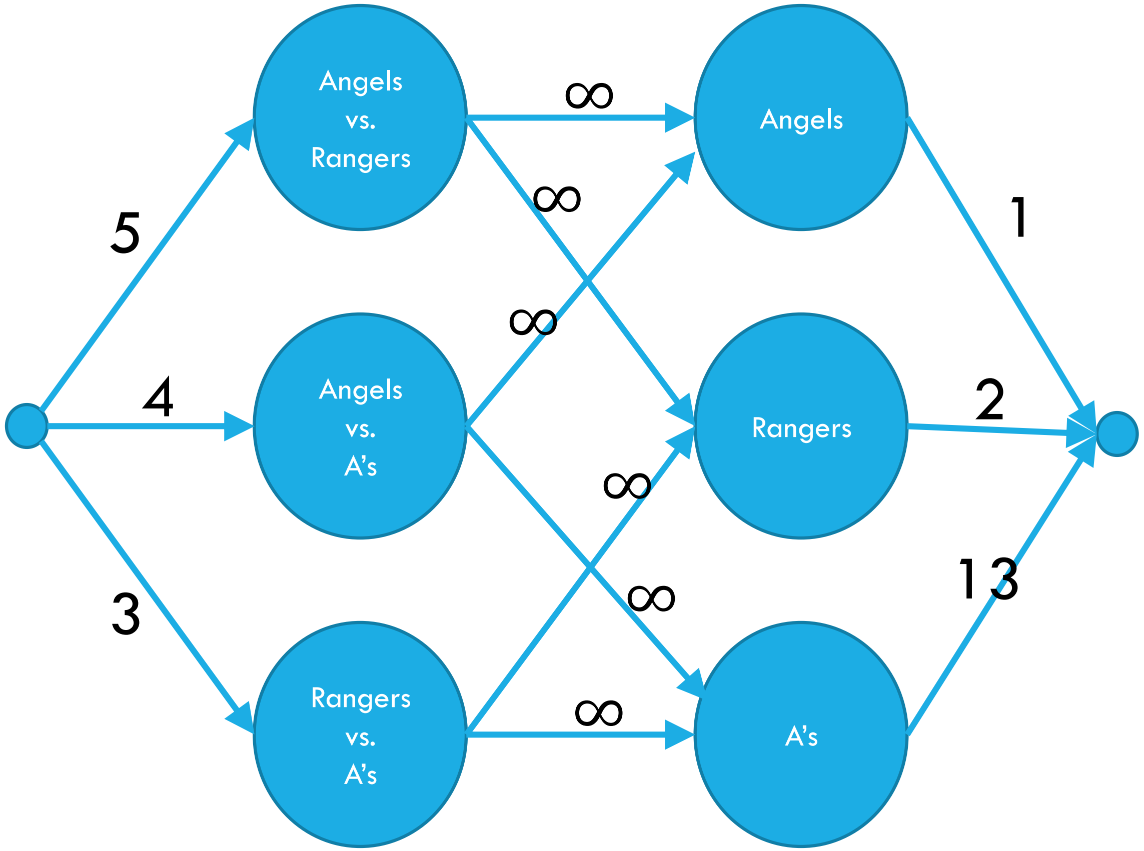
g_{ij} is games to be played between i and j
P is number of wins possible for Mariners
w_i is current number of wins for team i.

Let $(S, \bar{S})$ be a min-cut.
There's a lot of structure in the min-cut.
Let R be the set of teams whose vertices are reachable from s after the edges have been cut.

The capacity of the cut is
$\sum_{i \notin R \text{ or } j \notin R} g_{ij} + \sum_{i \in R} P - w_i$

And the capacity of the cut is less than $\sum_{i,j} g_{ij}$ (because that is a cut, and we can't have a flow of that value).

If R is a set of teams, let $a(R) = \frac{\sum_{i \in R} w_i + \sum_{i,j \in R} g_{i,j}}{|R|}$ the average number of games won be a team in R.



An Explanation Always Exists

g_{ij} is games to be played between i and j
P is number of wins possible for Mariners
w_i is current number of wins for team i.

$$\sum_{i \notin R \text{ or } j \notin R} g_{ij} + \sum_{i \in R} P - w_i < \sum_{i,j} g_{ij}$$

$$\sum_{i \in R} P - w_i < \sum_{i \in R, j \in R} g_{ij}$$

After subtracting pairs where at least one of i, j are not in R all that remains are pairs where both i, j are in R.

$$|R|P < \sum_{i \in R, j \in R} g_{ij} + \sum_{i \in R} w_i$$

Move w_i to the other side. P is a constant, so we just add $|R|$ copies of P.

$$P < \frac{\sum_{i \in R, j \in R} g_{ij} + \sum_{i \in R} w_i}{|R|}$$

That is, the average number of wins for a team in R (after all games are played) is strictly more than the possible number of wins for the Mariners.

Summary

To tell whether your favorite team is eliminated, you can run a max-flow computation on a graph with $O(n^2)$ vertices and $O(n^2)$ edges.

If your team is eliminated, there is a witness set of teams that must average more wins than is possible for your team.