Lecture 04: More on Locality

Announcements

1. Assignment 01 due next Friday
    - draft posted
    - work in small groups — 1 submission per group
    - group assignment survey (see email)

2. Attendance & COVID
    - college masking policies strictly enforced
        - no food / drink in class
    - all sessions recorded, notes posted
    - do not come to class if sick/exposed
    - risk = shared risk

Last Time

Port #ing

Introduced Port Ordering (PO) Model

- network = graph w/ port ordering
  - no unique identifiers for nodes

LOCAL = PO + IDs

- execution in synchronous rounds
  - receive msgs
  - perform local computations (no restrictions!)
  - send msgs to neighbors

- Stable Matchings in PO model
  - SM instances ⟶ PO networks
  - additional input: role (student or internship)
  - GS algorithm has natural interpretation as protocol in PO model
  - ISSUE: termination detection

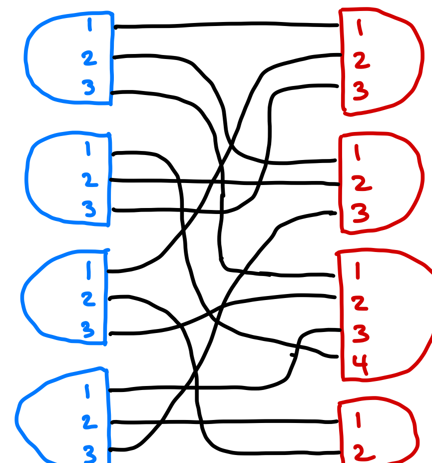
SM Instance

	1	2	3	4
Anna:	a	b	c	
Beck:	c	b	a	
Cameron:	a	d	c	
Daniel:	c	d	b	

	1	2	3	4
a:	A	C	B	
b:	A	B	D	
c:	A	C	D	B
d:	D	C		

$\Downarrow$

PO Network



Gale Shapley as a PO protocol

Procedure for students:

Initialize:

cur = 1 — cur match

each round i, do

- if $i = 1$, send "apply" to cur
- if received "reject" from cur in round $i-1$
  - if cur = my degree, return $\perp$ and halt
  - else, cur ← cur + 1, send "apply" to cur

return cur and halt

Procedure for internships

Initialize:

cur = ∞

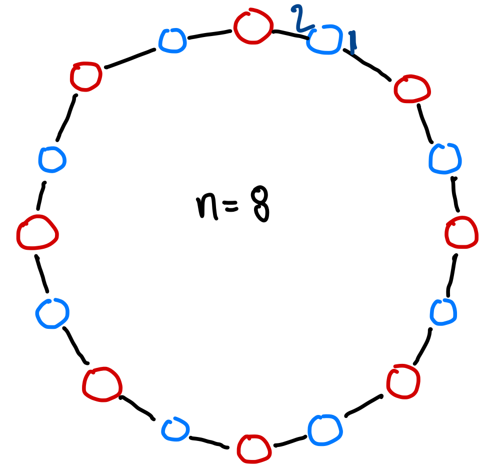
each round i, do

- set $\boxed{\text{reject}}$ ← $\emptyset$ — applicant to reject
- for each j from which received "apply" in round $i-1$
  - if $j <$ cur
    - add cur to reject
    - set cur ← j
  - else
    - add j to reject
- for each j in reject
  - send "reject" to j

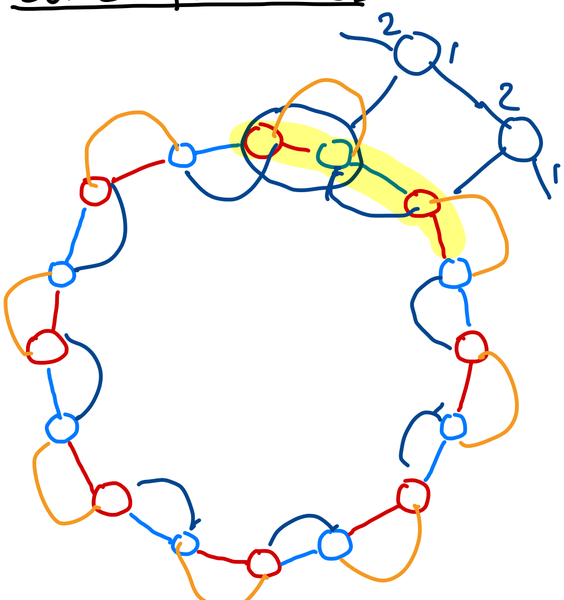
return cur

An Illustrative Family of SM Instances

1. n students, n internships
2. each agent has only 2 acceptable partners
3. the graph of acceptable partners forms a cycle



Some preferences



- students prefer blue neighbor to red neighbor
- internships prefer red neighbor to blue neighbor

find matching of blue edges

Questions.

1. Which SM does GS algorithm find?
2. What other SMs are there?

Internships apply to students

red matching

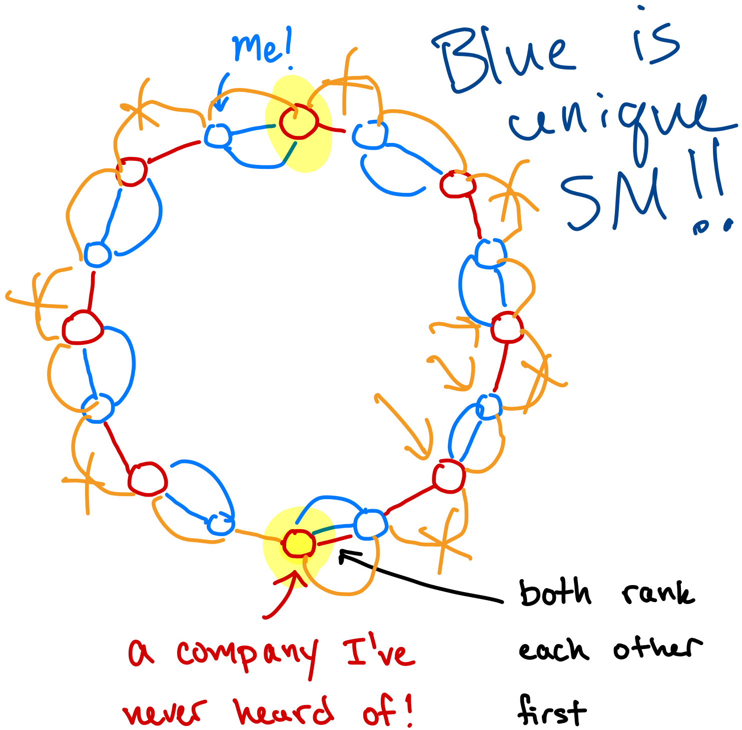
Recall

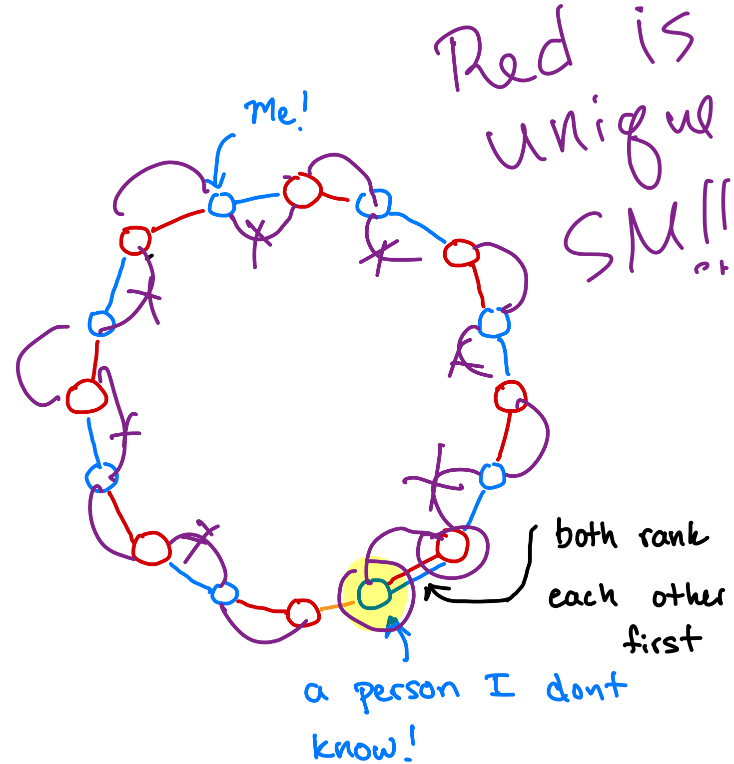
- can sometimes have multiple possible stable matchings
- here's how to find 2 (if possible):
  - run GS alg w/ students applying to internships
  - run GS alg. w/ internships applying to students

Fact. If the SM instance has multiple stable matchings then procedures above will give <u>different</u> stable matchings

$\Rightarrow$ if procedures above give <u>same</u> SM, then it is the <u>unique</u> SM for the instance.

Slight Modifications





What happens now?

The Moral.

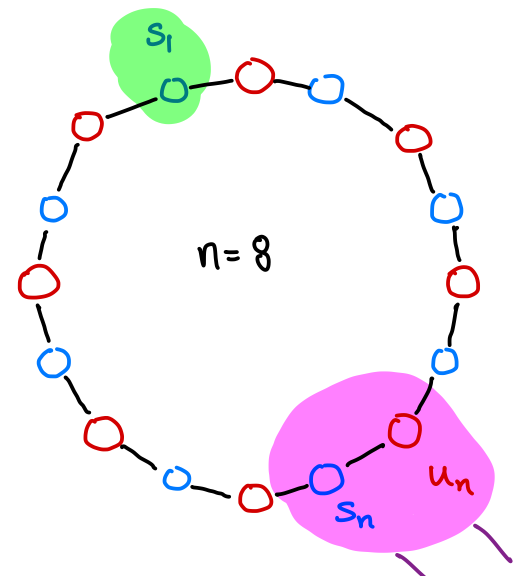
My (correct) output depends on input (prefs) of some company I've never heard of and some person I don't know!

- correct local output requires knowledge about nodes that are far away in the network

"A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable."

— Leslie Lamport

Obs. Correct output of S_1 depends on input of S_n and U_n, regardless of what algorithm is used to find SM



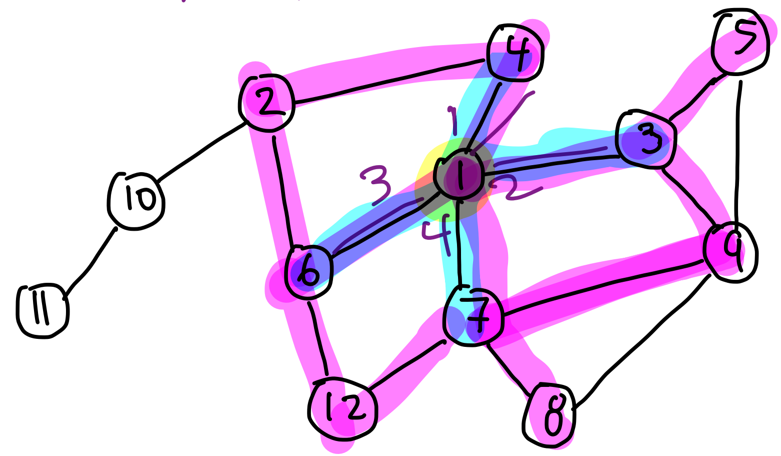
Next. We will use this observation to show that no distributed algorithm (in PO or LOCAL model) can find SMs "much faster" than GS on some instances

A Bit More Graph Theory

$E = \{(1,3), (1,4), (1,6), (1,7), \ldots\}$

vertices $= \{1, 2, \ldots, 12\}$

- $G = (V, E)$ a graph
- edges labelled w/ port ordering
- vertices have local inputs
  (possibly unique identifiers)

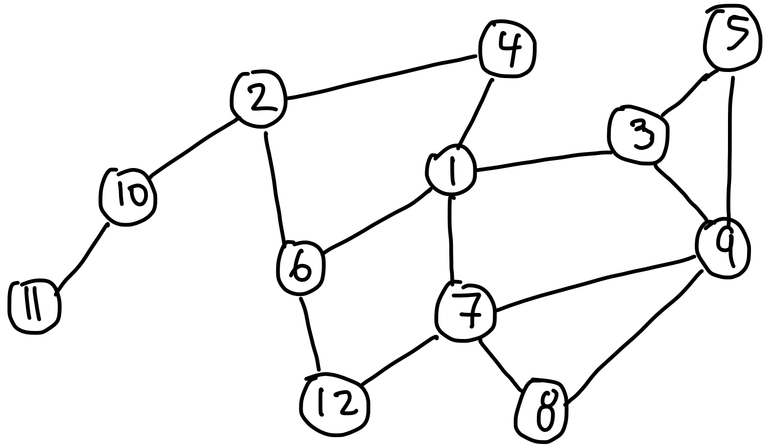


Def. Given vertex v, distance d ①
the d-neighborhood of v is the subgraph of G consisting of all vertices within distance d of v and edges w/ one endpoint within distance $d-1$ of v

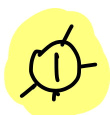
$$\Gamma_d(v) = (V', E')$$

V' = vertices within distance d of v

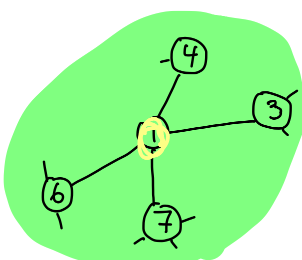
E' = edges w/ one or more endpoints within dist $d-1$ of v

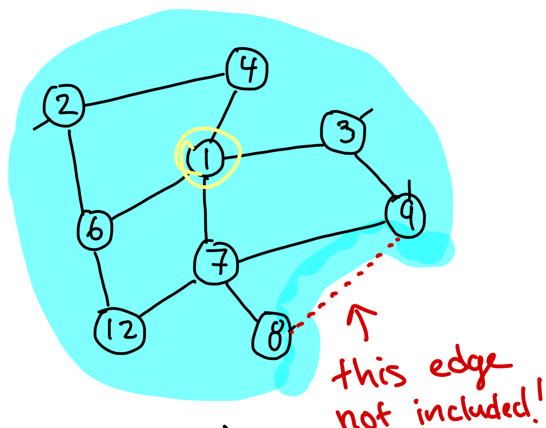


$\Gamma_0(1)$



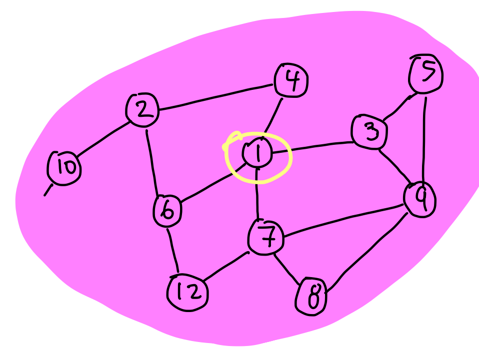
$\Gamma_1(1)$





$\Gamma_2(1)$

$\Gamma_3(1)$



Connecting Neighborhoods and Protocols.

Goal. In any distributed protocol, messages sent in round r are determined by distance $r-1$ neighborhoods.

- Intuitively: data can only travel one "hop" per round
  - $\Rightarrow$ takes $r-1$ rounds to travel $r-1$ hops
  - $\Rightarrow$ by round r, can only have gotten info. from nodes within dist $r-1$

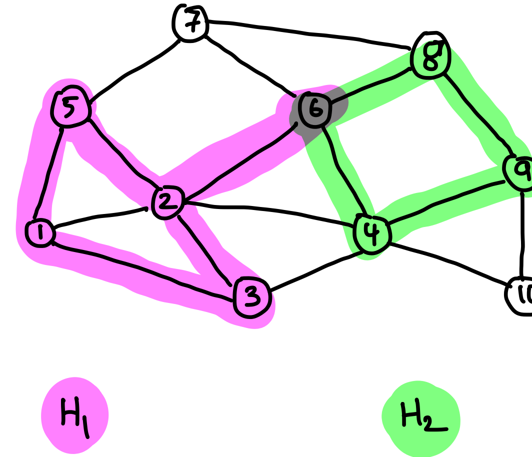
- Formally: prove by induction that each v's state in round r is determined by the initial states of nodes in $\Gamma_{r-1}(v)$
  - msgs sent in round r are determined by state in round r.

Union of Subgraphs

Suppose H_1, H_2 are sub-graphs of a graph $G = (V, E)$. Then the union of H_1 and H_2, denoted $H_1 \cup H_2$, consists of

(1) all vertices in H_1 or H_2

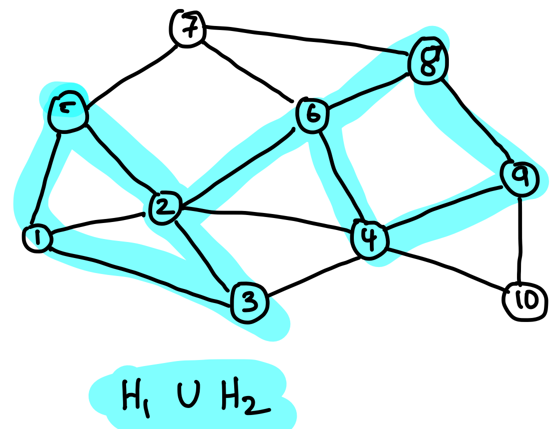
(2) all edges in H_1 or H_2



We can also form unions of many subgraphs:

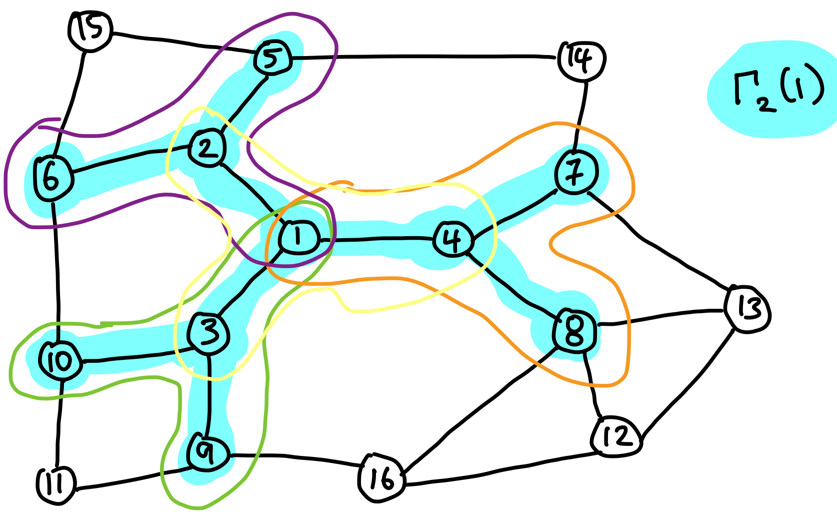
$$H_1 \cup H_2 \cup H_3 \cup \cdots \cup H_k = \bigcup_{i=1}^{k} H_i$$

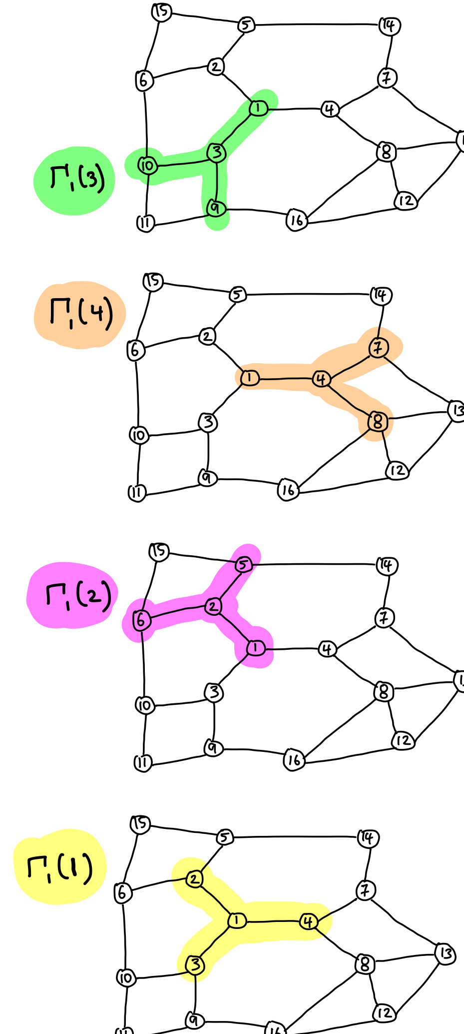
= vertices and edges in any of the subgraphs



Neighborhood Covering Lemma. Let $G=(V,E)$ be a graph and $v \in V$ a vertex in G. For any distance $d \geq 2$, the distance d neighborhood of v is the union of v's neighbors distance $d-1$ neighborhoods. Symbolically:

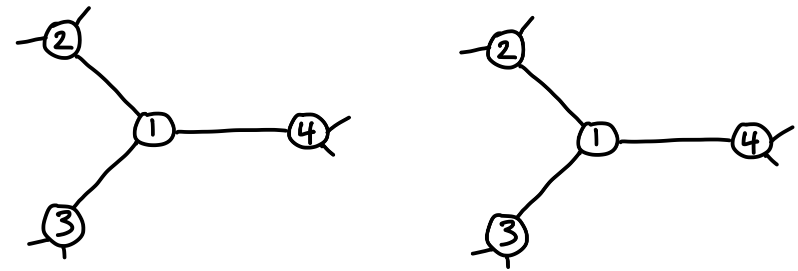
$$\Gamma_d(v) = \bigcup_{w \in \Gamma_1(v)} \Gamma_{d-1}(v)$$





For Now. Take neighborhood covering lemma on faith

→ see notes for a proof



Want to show: state of each vertex v is determined by the initial state of vertices in $\Gamma_{r-1}(v)$. Formally:

Locality Lemma. Suppose Π is a PO protocol, $G = (V, E)$ is a graph together w/ local inputs for each vertex v. Then for every round r, and vertex v, the state of v in round r is determined by the initial state of $\Gamma_{r-1}(v)$.

Proof idea. The state of v in round 1 is determined only by v's local input

$\Rightarrow$ msgs sent in round 1 are functions of local inputs

State in round 2 is function of v's local input and msgs received in round 1

$\Rightarrow$ msgs sent by v in round 2 are functions of inputs of distance 1 neighborhoods

...

<u>Proof</u>. Want to show that for all vertices v and rounds r, v's state is determined by $\Gamma_{r-1}(v)$.

Argue by induction on r.

Base case: $r = 1$.

- initial state of v is determined by v's local input: hence $\Gamma_0(v)$.

Inductive Step: $r-1 \Rightarrow r$

Suppose in round $r-1$, every u's state is determined by $\Gamma_{r-2}(v)$

Then msgs sent by each u in rnd $r-1$ are determined by $\Gamma_{r-2}(u)$

<u>PO protocol</u> consists of

<u>Sets</u>:

- Inputs = allowable local inputs
- States = allowable local states
- Outputs $\subseteq$ States halting states w/ outputs
- Messages = allowable messages

<u>Functions</u> (d = degree of node)

- <u>init</u>: Inputs $\rightarrow$ States

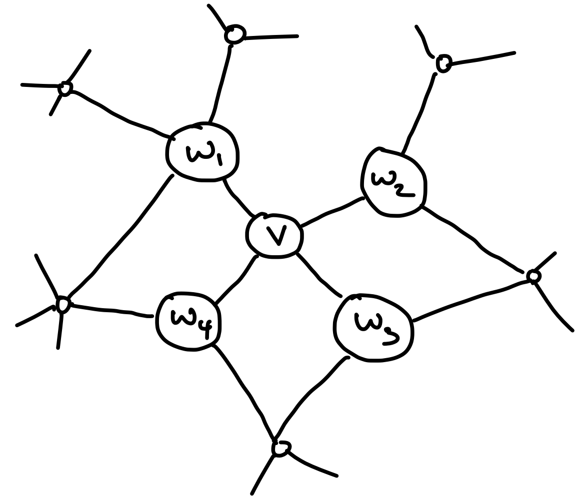
  determine initial state from initial input
- <u>send</u>: States $\rightarrow$ Messages^d

  determine the d messages to send to neighbors from current state
- <u>receive</u>: States $\times$ Messages^d $\rightarrow$ States

  determine how to update state from received messages

Inductive step continued...

Then msgs sent by each u in rnd $r-1$ are determined by $\Gamma_{r-2}(u)$



Consider vertex v w/ neighbors $w_1, w_2, \dots, w_k$ $\quad (k = \deg(v))$

Msg received from w_i in round $r-1$ is determined by $\Gamma_{r-2}(w_i)$

$\Rightarrow$ all msgs received in round $r-1$ are determined from

$$\Gamma_{r-2}(v) \cup \Gamma_{r-2}(w_1) \cup \Gamma_{r-2}(w_2) \cup \cdots \cup \Gamma_{r-2}(w_k) \quad = \Gamma_{r-1}(v)$$

$\Rightarrow$ v's state in round $r-1$ is determind by $\Gamma_{r-1}(v)$. //

A More Algorithmic View of Locality Lemma

In round 1, consider $\Gamma_{r-1}(v)$

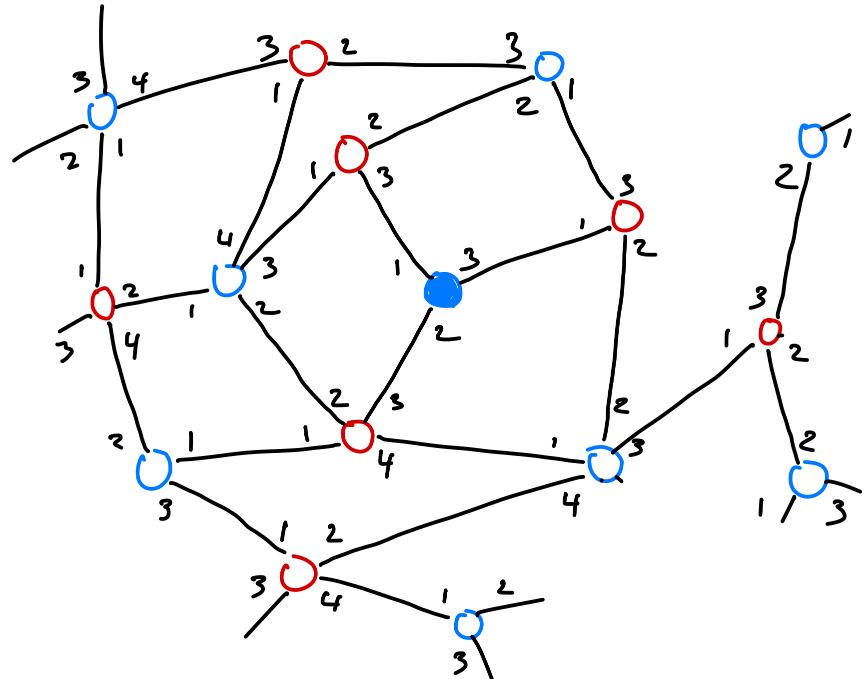
- compute all init. states
- compute all msgs
- determine all <u>received</u> msgs in $\Gamma_{r-2}(v)$

In round 2, consider $\Gamma_{r-2}(v)$

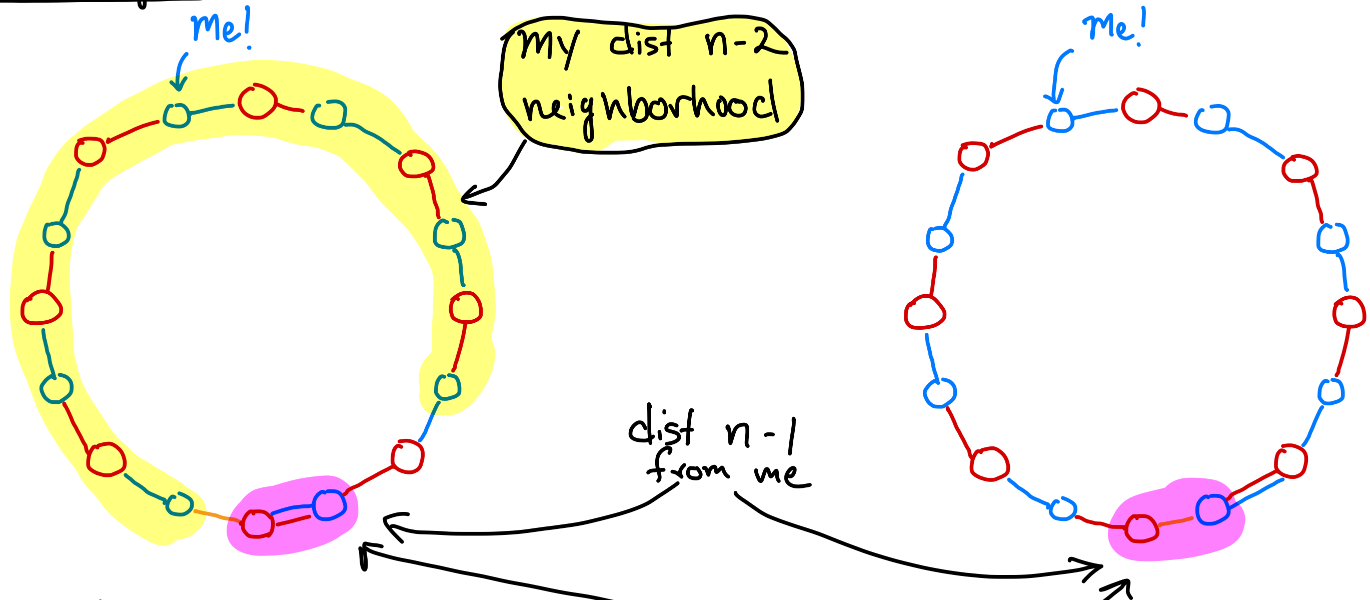
- compute all rnd 2 states
- compute all rnd 2 msgs
- find all <u>received</u> msgs in $\Gamma_{r-3}(v)$

⋮

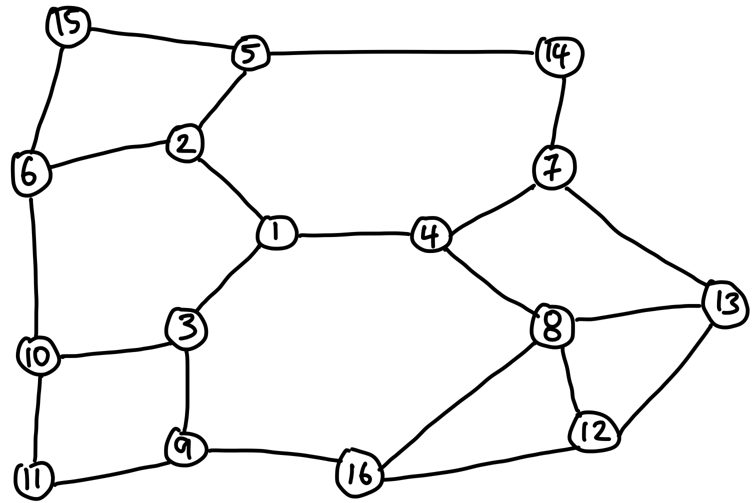
In round r-1, get all states / received msgs in $\Gamma_0(v) = \{v\}$.



Consider Again.



- n students, n internships
- my correct output depends on input of these
  - $\Rightarrow$ correct output is not determined by $\Gamma_{n-2}(\text{me})$
  - $\Rightarrow$ correct output cannot be determined in $\leq n-1$ rounds by any algorithm.


15
5
14
2
6
7
1
4
13
3
8
10
12
9
16
11