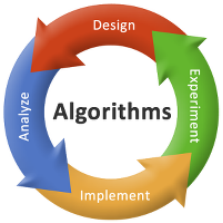
Overlay Graphs

Course: Algorithms

Faculty: Dr. Rajendra Prasath



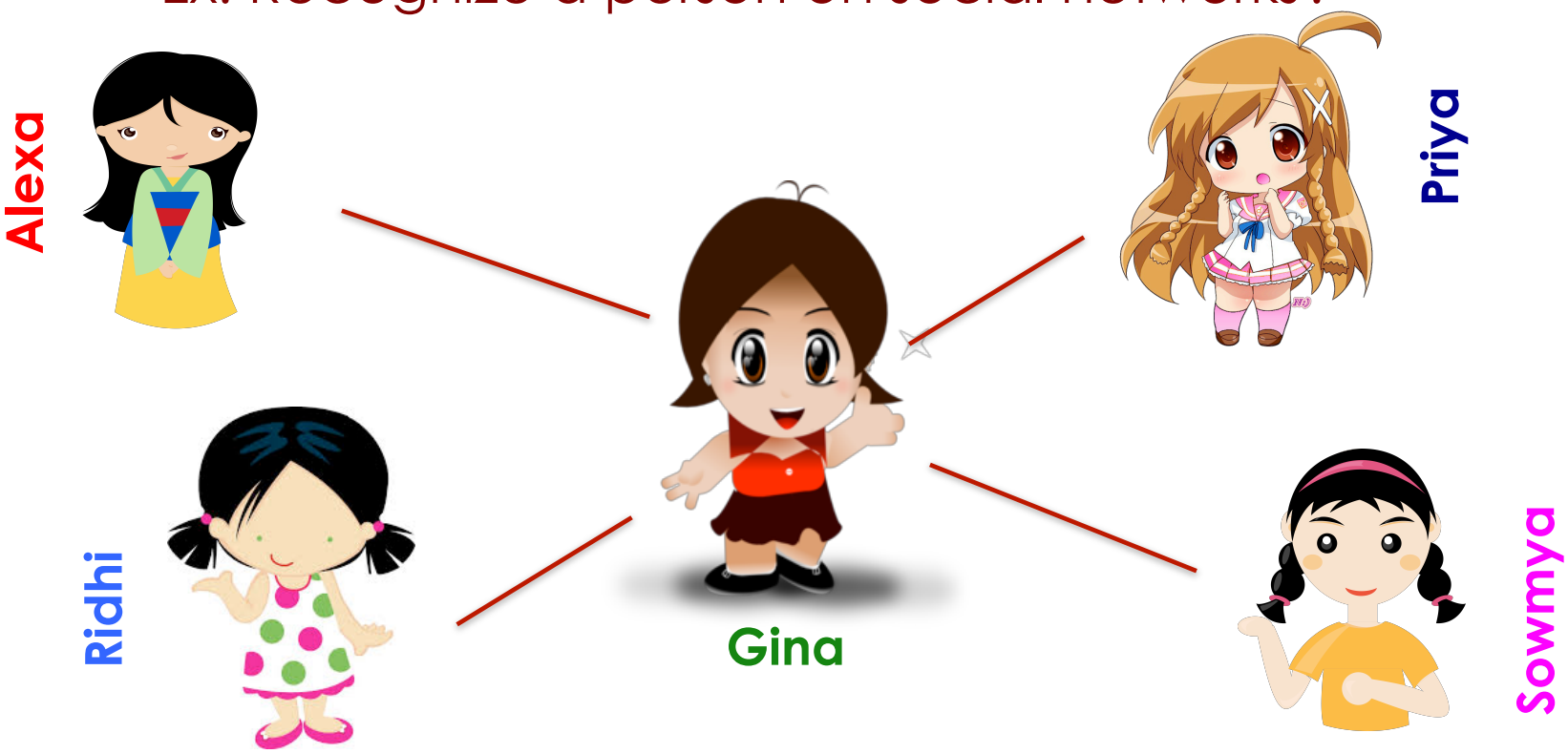
Autumn 2018

Overlay Graphs

This lecture covers the interesting aspects of overlay graphs. We will also look at certain applications with their implications

Recap: Graph Algorithms

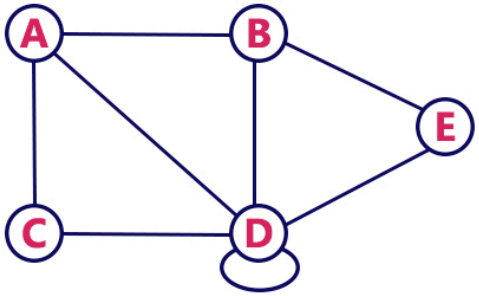
- Different Networks
  - Ex: Recognize a person on social networks?



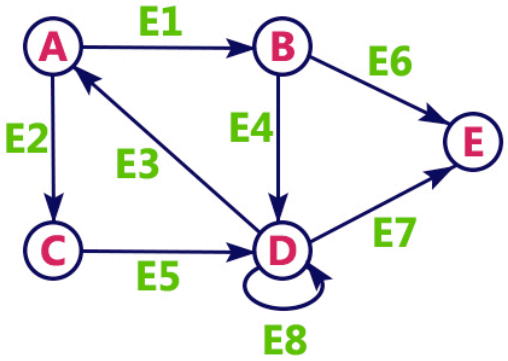
- Friend of a Friend is also a **Friend (Mutual Friend)**

Recap: Graph Representations

- Examples



	A	B	C	D	E
A	0	1	1	1	0
B	1	0	0	1	1
C	1	0	0	1	0
D	1	1	1	1	1
E	0	1	0	1	0



	E1	E2	E3	E4	E5	E6	E7	E8
A	1	1	-1	0	0	0	0	0
B	-1	0	0	1	0	1	0	0
C	0	-1	0	0	1	0	0	0
D	0	0	1	-1	-1	0	1	1
E	0	0	0	0	0	-1	-1	0

Recap: Graph Algorithms

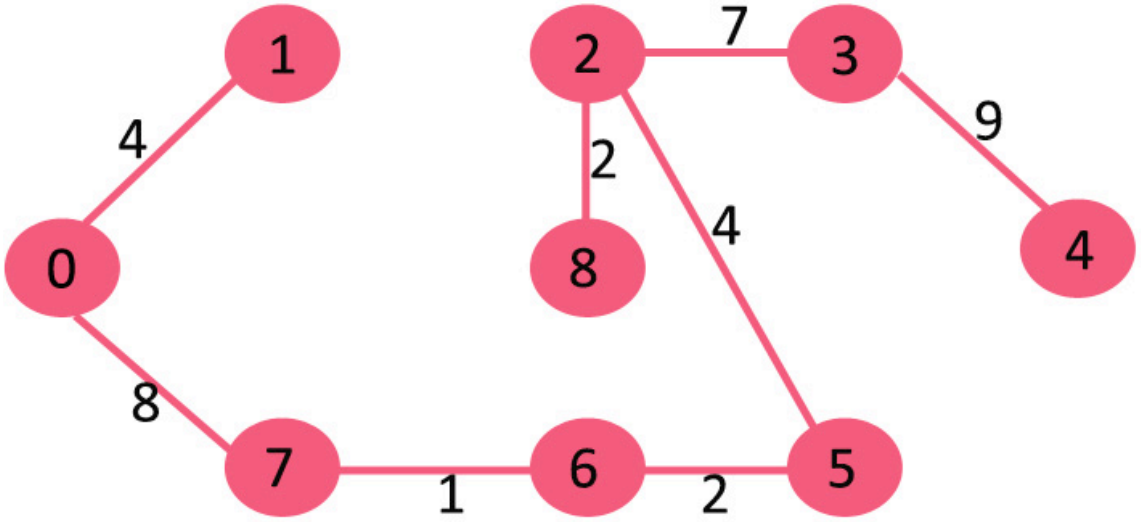
- Problems could be better represented and solved using Graph Algorithms
  - We focus on a traversal problem:
    - Given G=(V, E) and a vertex v
    - Find all w in V such that w connects v

- Two Graph Traversals:
  - Breadth First Search (BFS)
  - Depth First Search (DFS)
    - Getting the connected components

- Next: Minimum Spanning Trees ➔ ➔ ➔

Kruskal Algo – An Example

- Minimum cost:

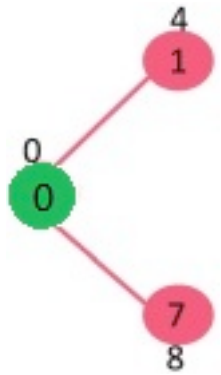


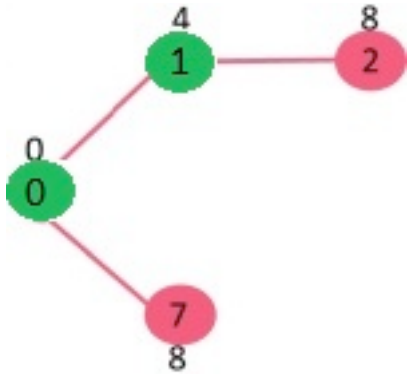
4 + 8 + 1 + 2 + 4 + 2 + 7 + 9

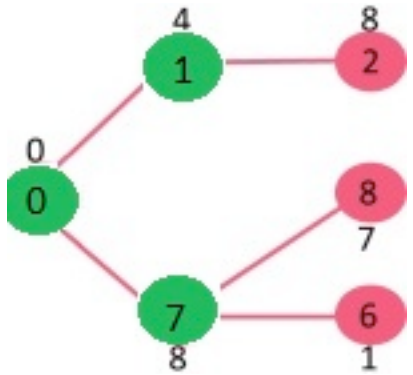
= 37 (Correct ?!)

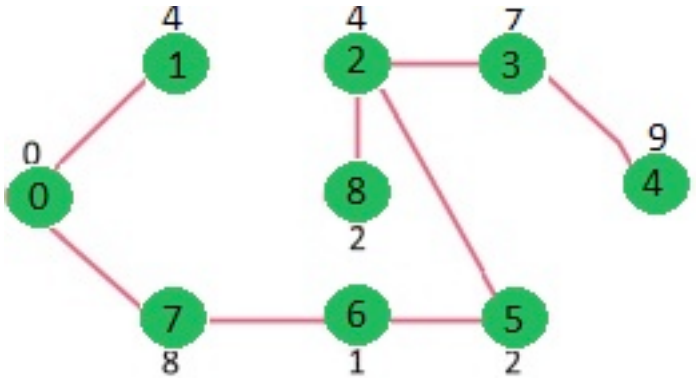
Prim Algo- Illustration

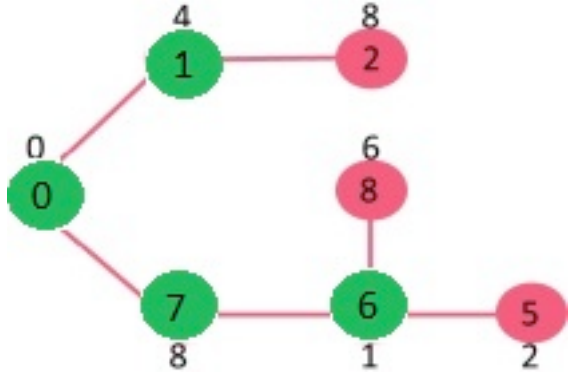
- Steps:





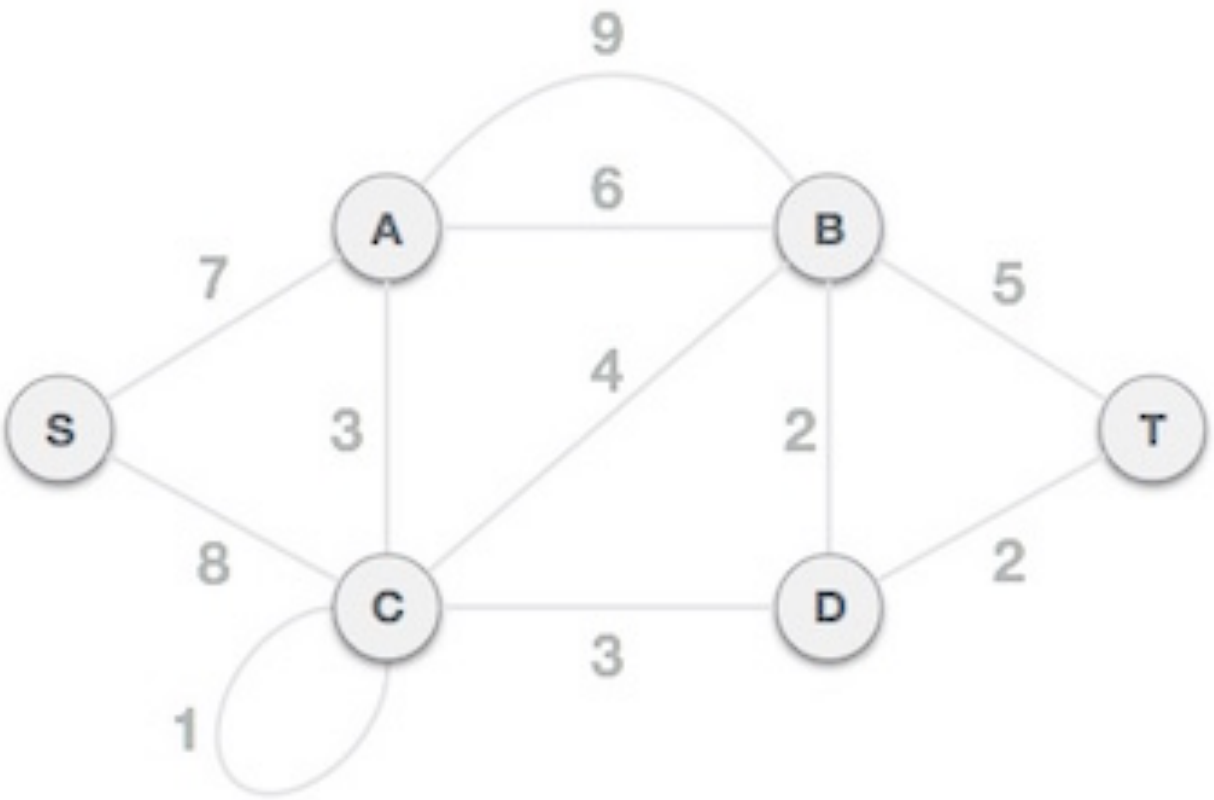






Prim Algo - Exercise

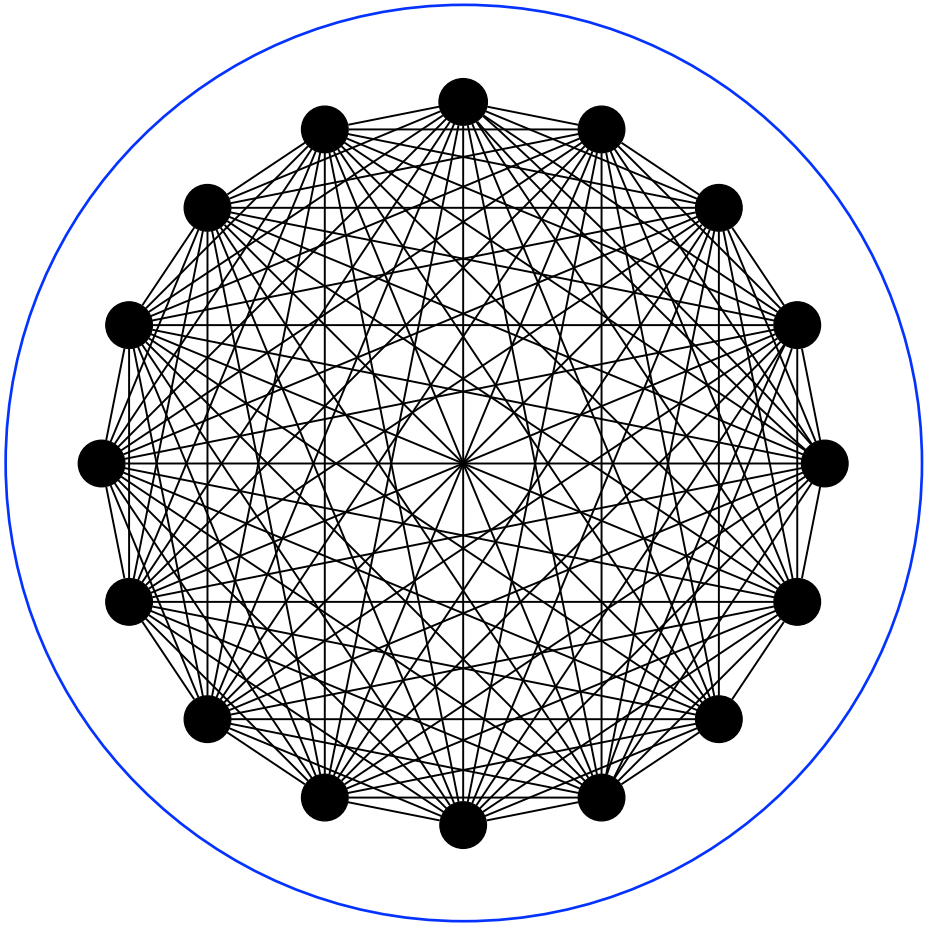
- Try Prims algorithm



Overlay Graphs (Networks)

Given a collection of (changing) servers:

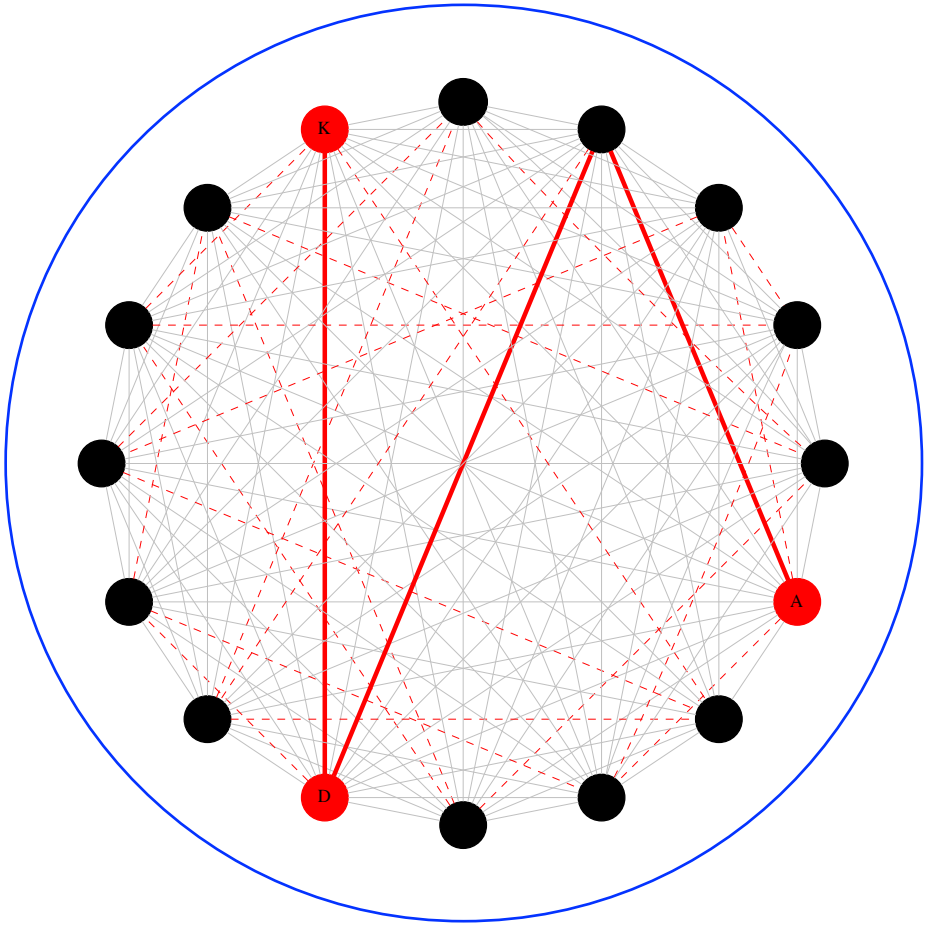
- Choose a good subset of the edges.



Overlay Networks

Given a collection of (changing) servers:

- Choose a good subset of the edges.
- Subgraph has low degree.
- Subgraph has low diameter.



Ground Rules

Underlying network:

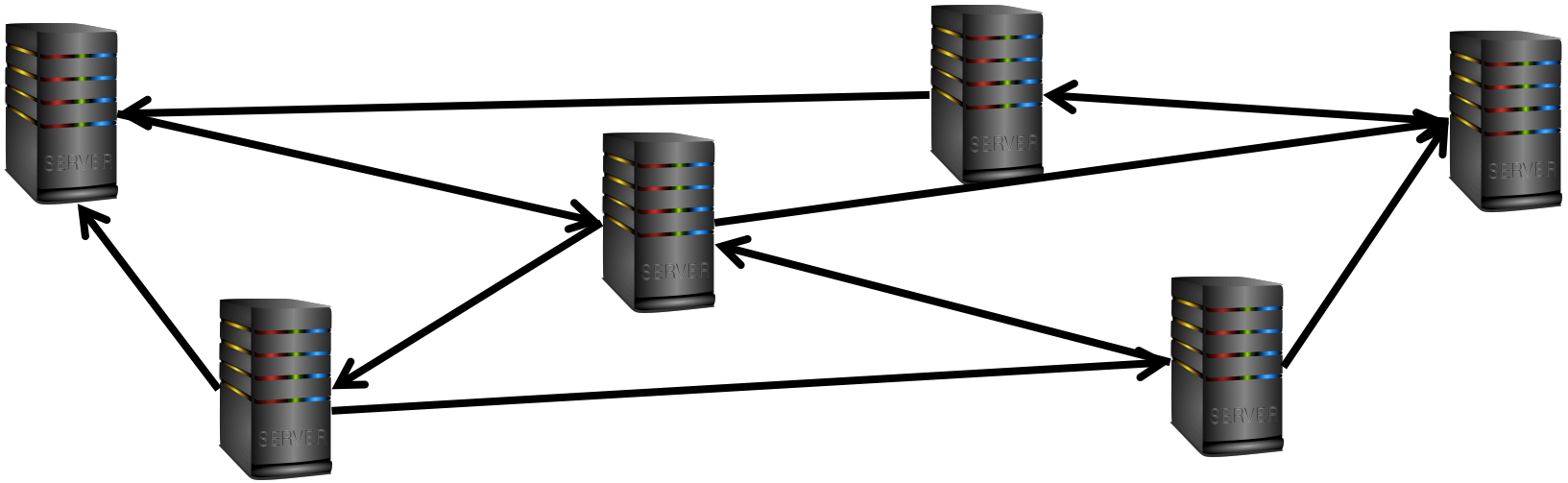
- Collection of nodes.
- Nodes arrive (join). ➔ Joining node is connected to *someone*.
- Nodes leave (fail)

Goals

Overlay Graphs:

- Low degree → constant or logarithmic.
- Low diameter → logarithmic or polylogarithmic

Note: every existing solution guarantees these properties.

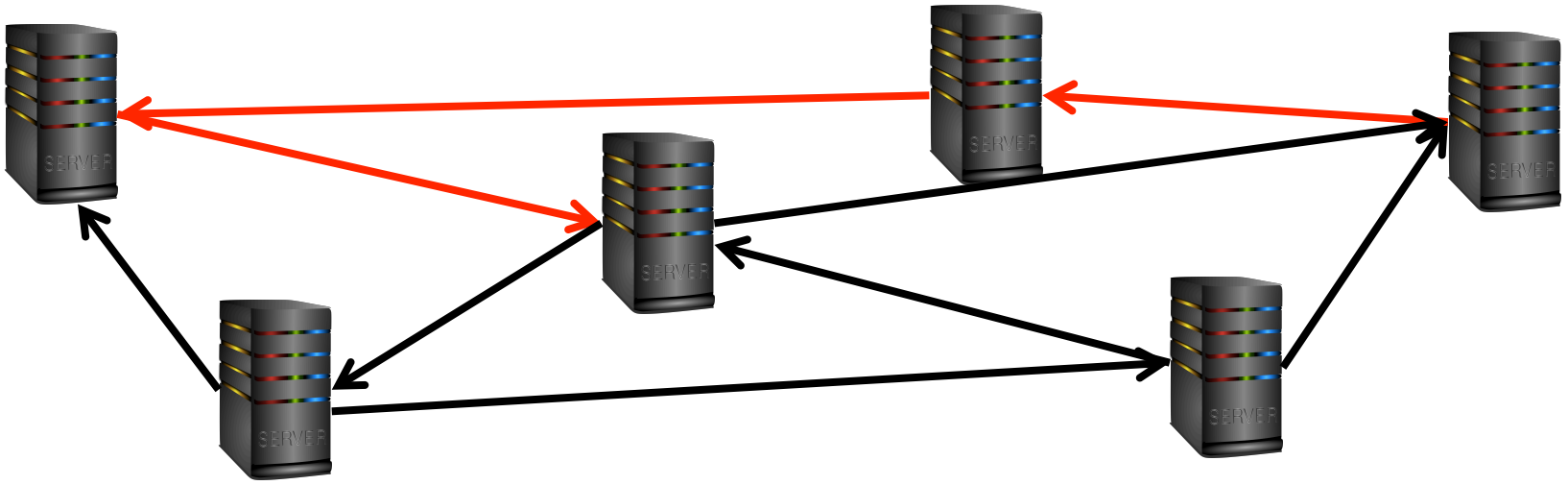


Goals

Routable:

- There exist short paths…
- … and we can find short paths.

Note: random graphs may not be good!



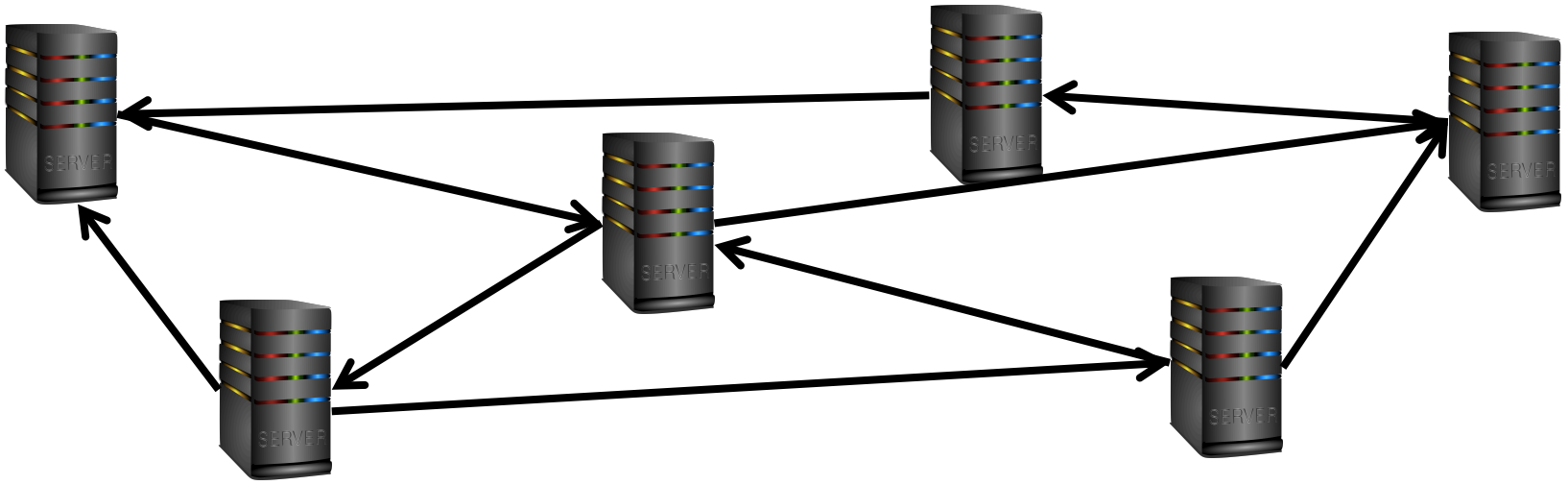
Goals

Other properties:

- Good expansion
- Good conductance

→ Random walks converge quickly

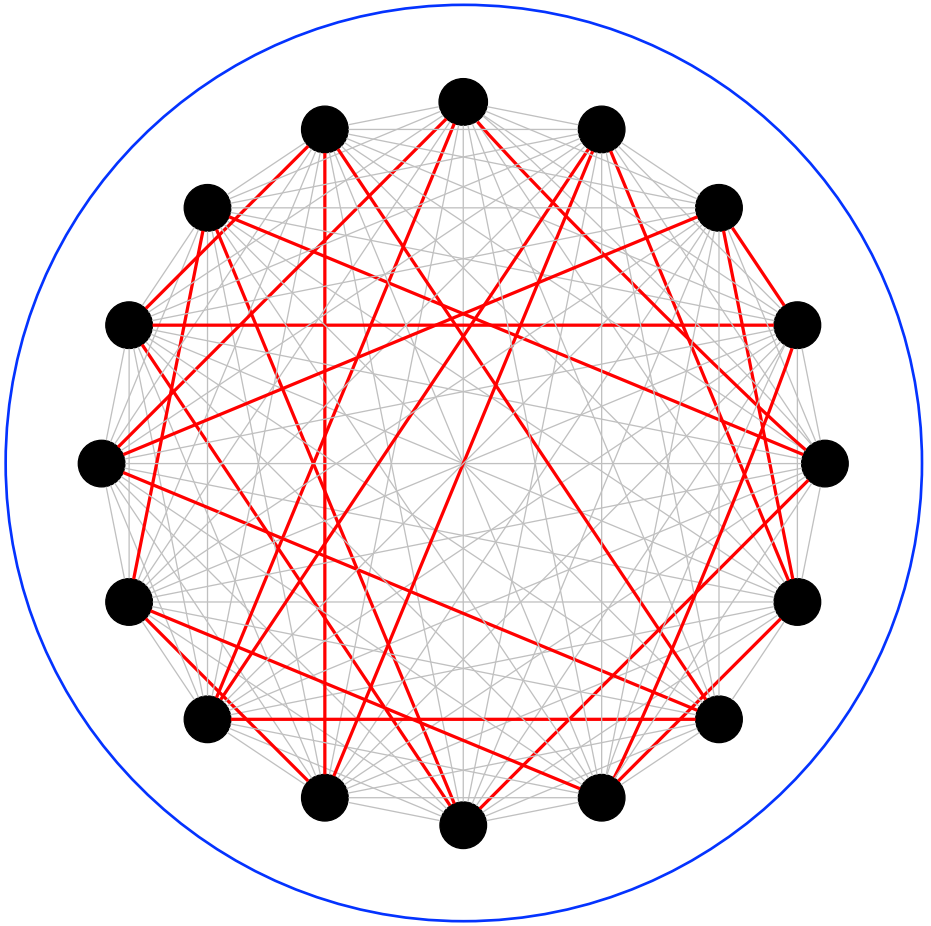
→ Diameter is small



Overlay Graphs

Given a collection of (changing) servers:

- Choose a good subset of the edges.
- Subgraph has low degree.
- Subgraph has low diameter.
- Maintain edges as servers come and go.



Chord in Ring Network

Advantages:

- Tolerates high rate of churn.
- Overlay has many good properties.
- Simple to implement.

Disadvantages:

- Oblivious adversary
- Fragile → Easy to attack, at risk of correlated failures.
- Limitations on where nodes can join.
  → Once the ring is disrupted, all is lost!
- Only supports one topology.

How to build an overlay?

From any initial state:

- Build a good overlay.
- Fast construction.

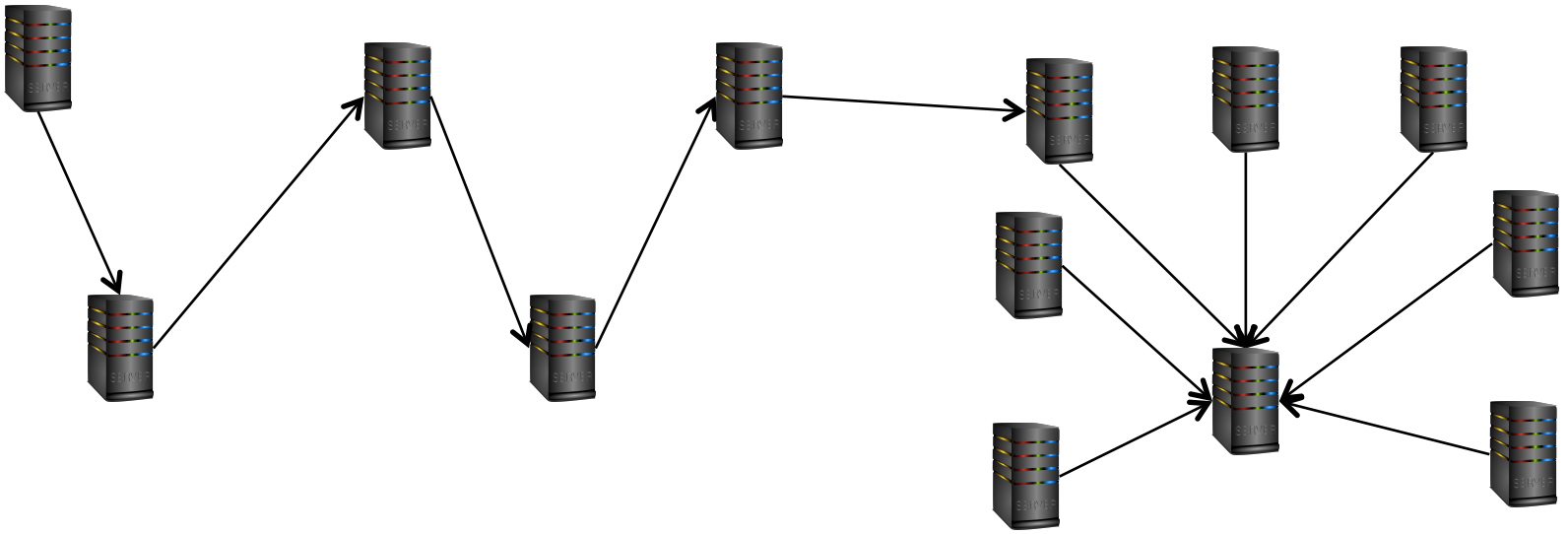
Self-stabilization:

- Arbitrary initial state corruption.
- Converges to a good state

Arbitrary initial state - Assume

Dynamic Graph Model:

- Graph is given in an arbitrary **connected** topology.
- State of the nodes may be corrupted.

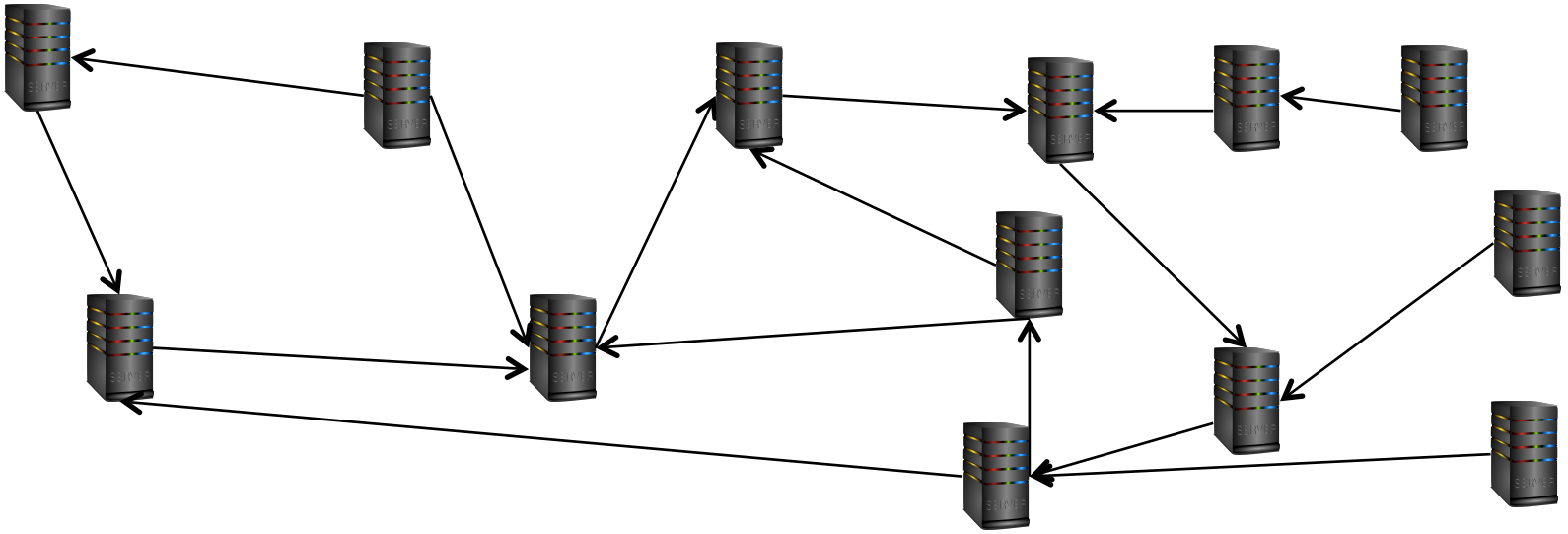


Construct a good overlay

In every round:

- Exchange messages with neighbors.
- Adjust edges.
- Improve the overlay.

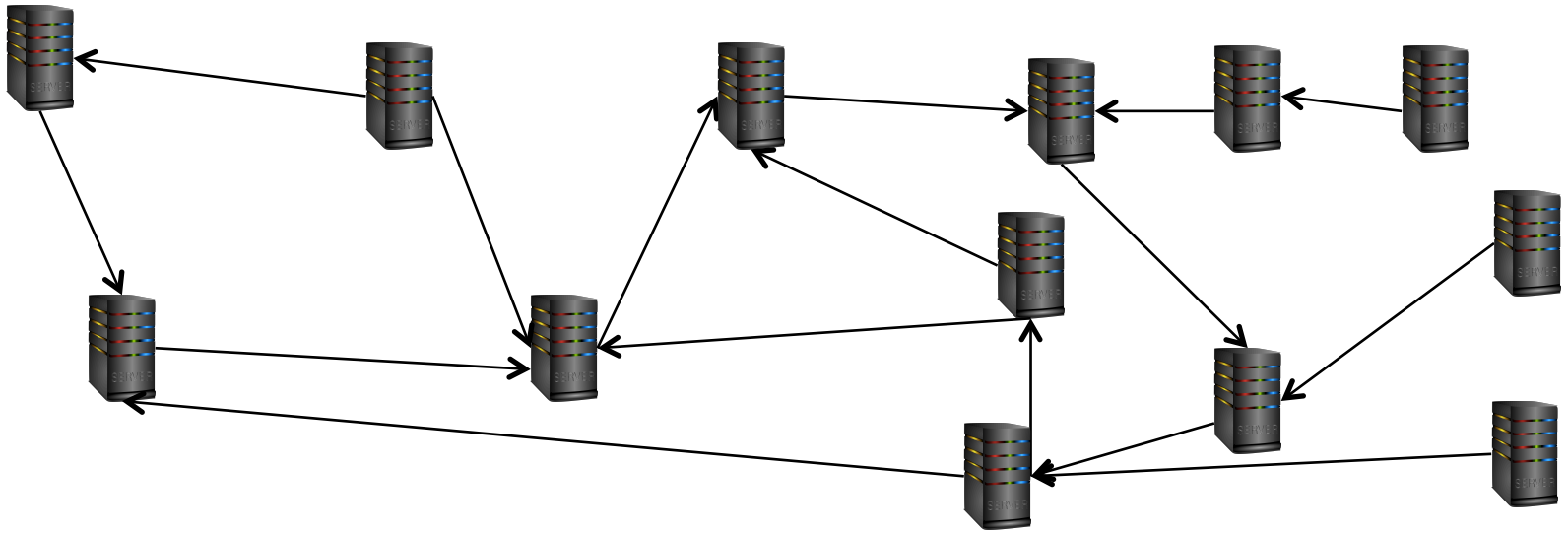
No further joins/leaves allowed (until overlay is constructed).



Stabilization

Eventually:

- Overlay is constructed.
- Good properties are guaranteed.
- In good state, joins and leaves may be supported.



How to build it?

Overlay guarantees:

- Fast searches / routing.
- Fast joins / leaves.
- Small diameter.
- Small degree.

Overlay construction issues:

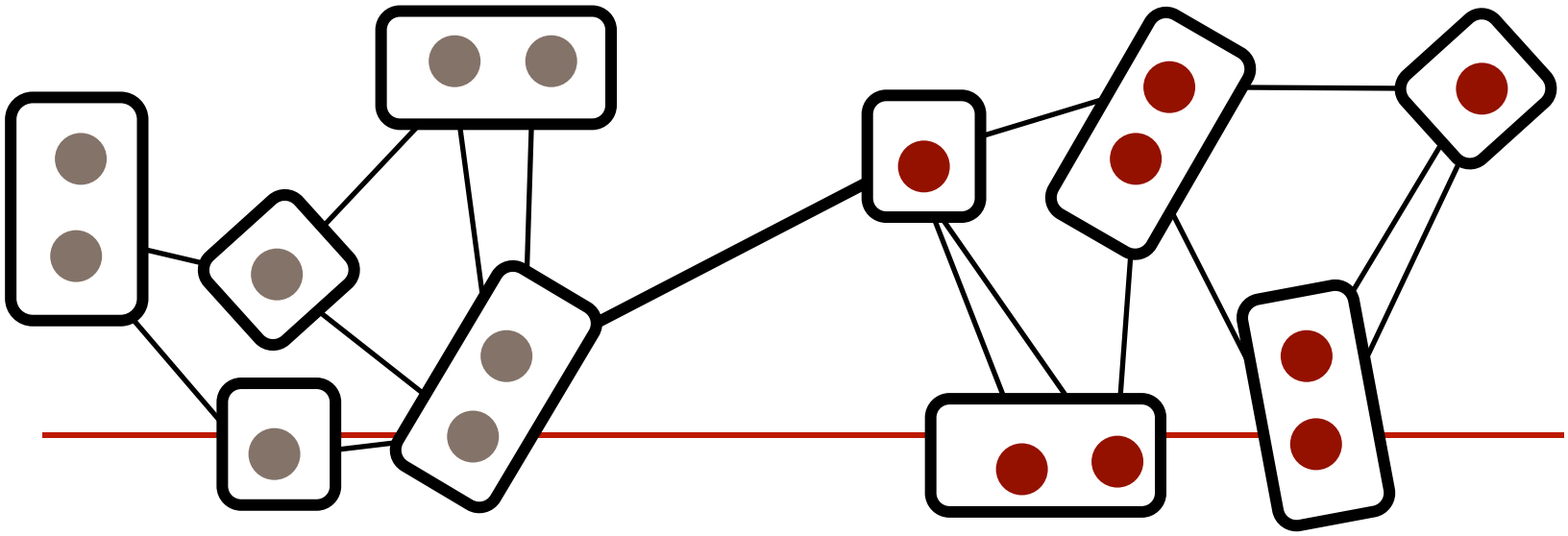
- Initially, diameter of the graph may be large.
- How do nodes find their neighbors efficiently?
- How do nodes sort themselves properly into linked lists?
- Leverage parallelism? → Not one insertion at a time!

Combining Two Overlays

Assume two overlay networks:

- Each overlay is a properly mapped virtual topology.
- The two overlays are connected by one edge.

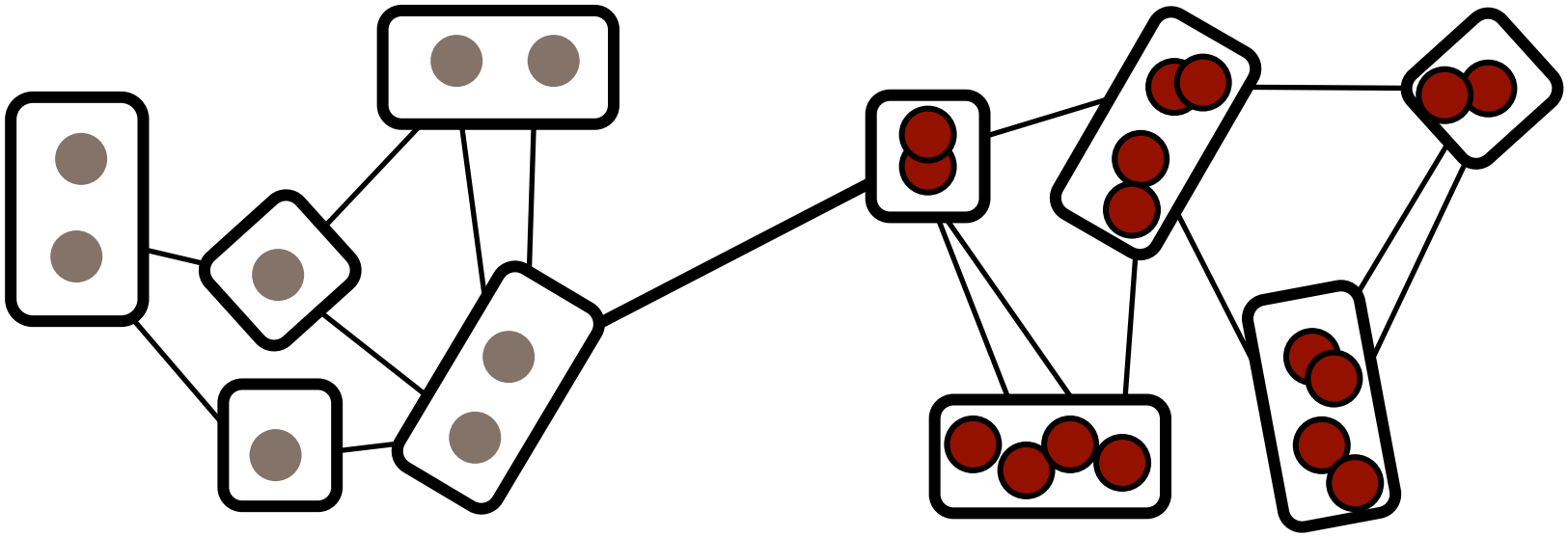
Key challenge: merge the two overlay networks.



Combining Two Overlays

Step 1: Grow the virtual topology

- Double one overlay (using topology mapping G_n ➔ G_{2n}).
- Creates excess virtual nodes.



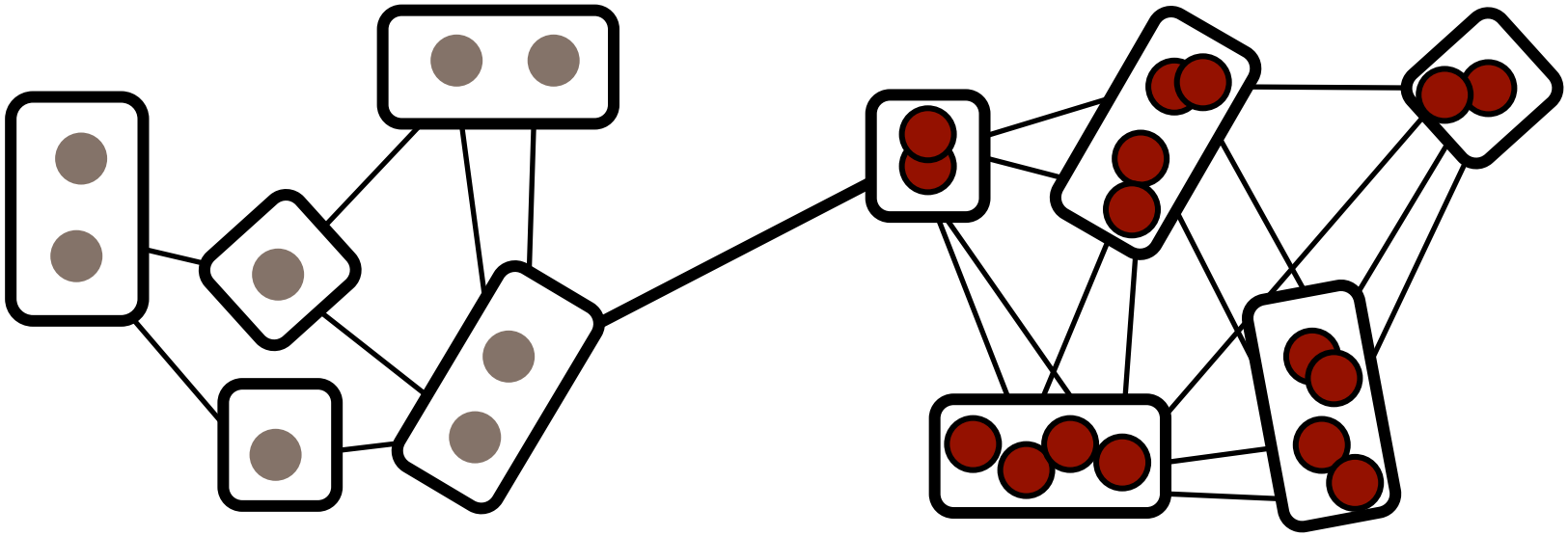
Combining Two Overlays

Step 1: Grow the virtual topology

- Double one overlay (using topology mapping G_n ➔ G_{2n}).
- Creates excess virtual nodes.

Step 2: Create new edges

- Use permutation routing.

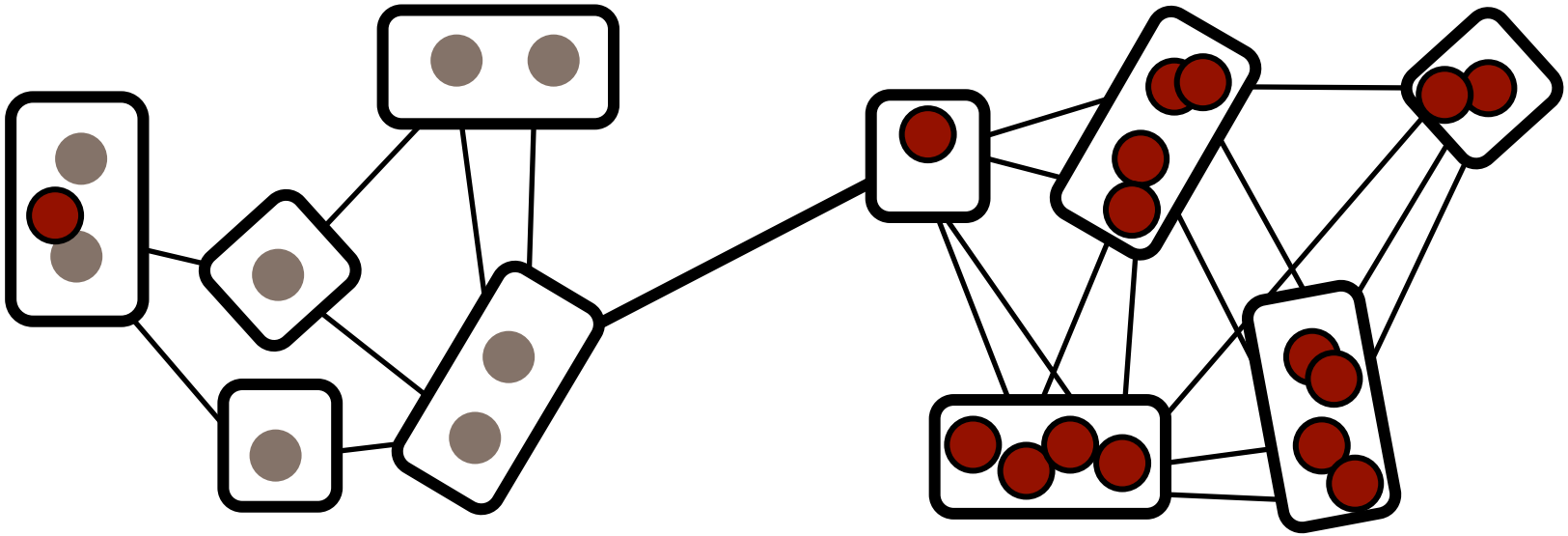


Combining Two Overlays

Step 3: Distribute excess virtual nodes

- Send new virtual nodes on a random walk of old topology.
- (Use random sampling of target topology.)

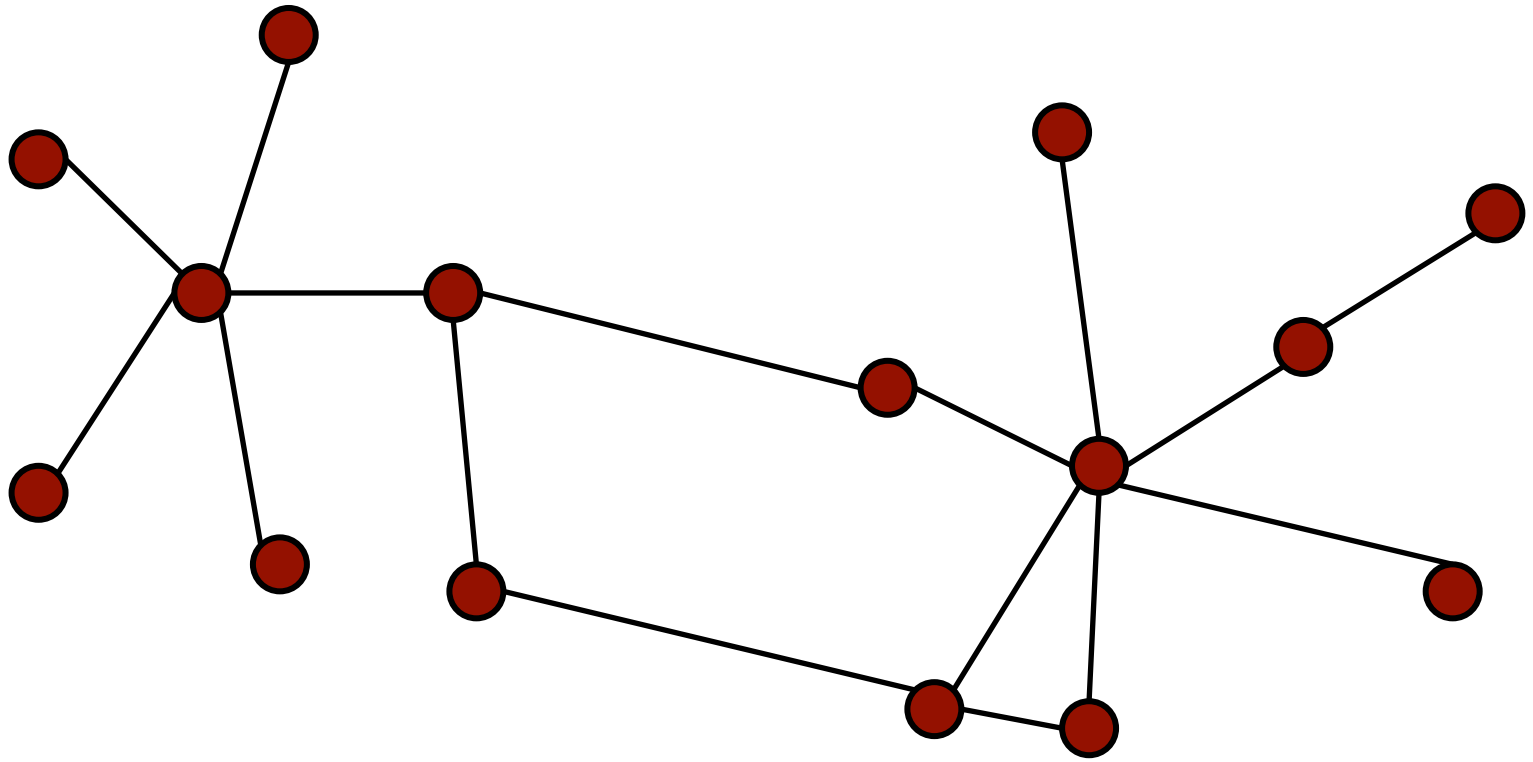
Problem: only one bridge.



Overlay Construction

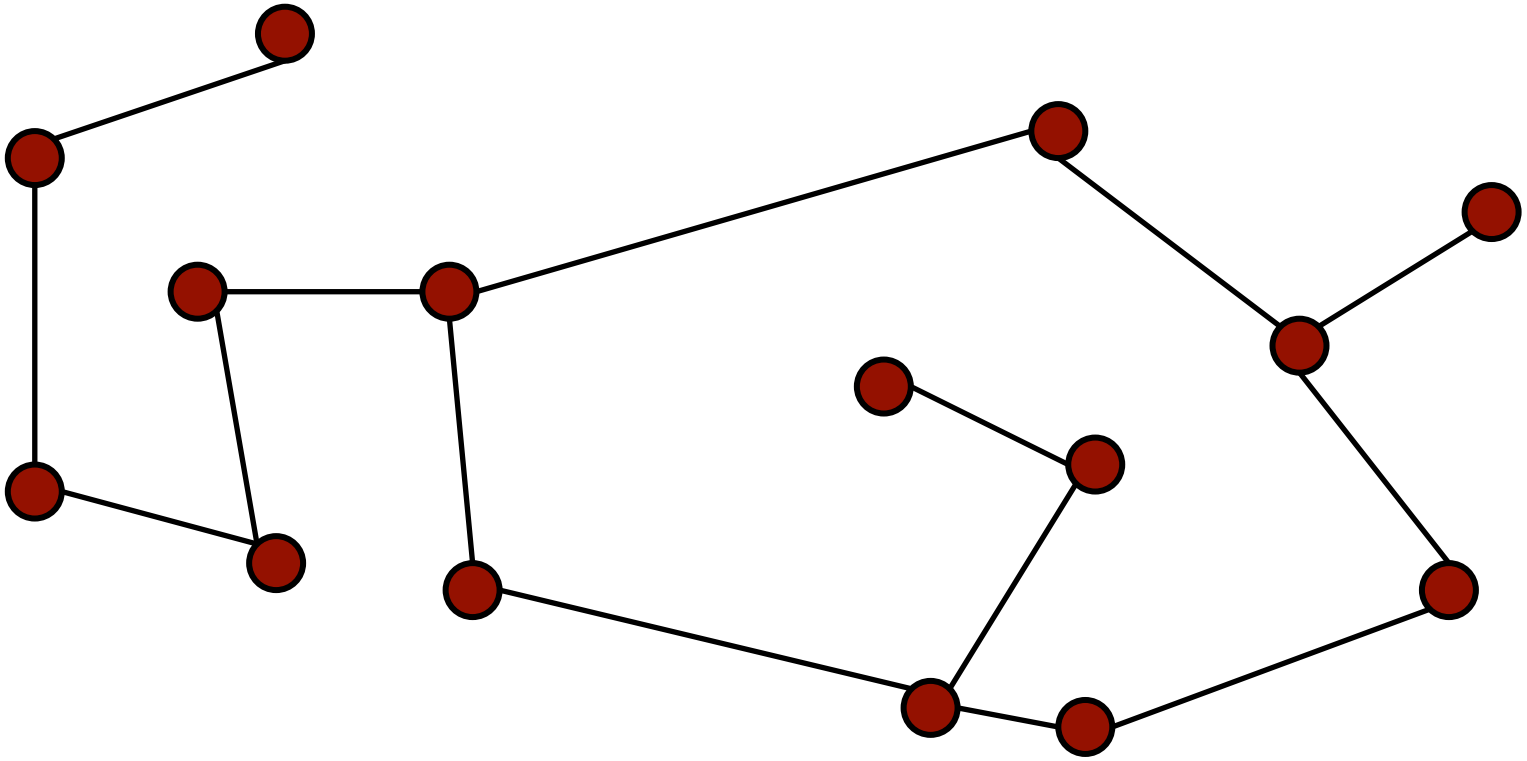
Divide-and-Conquer Algorithm

Initially: connected graph



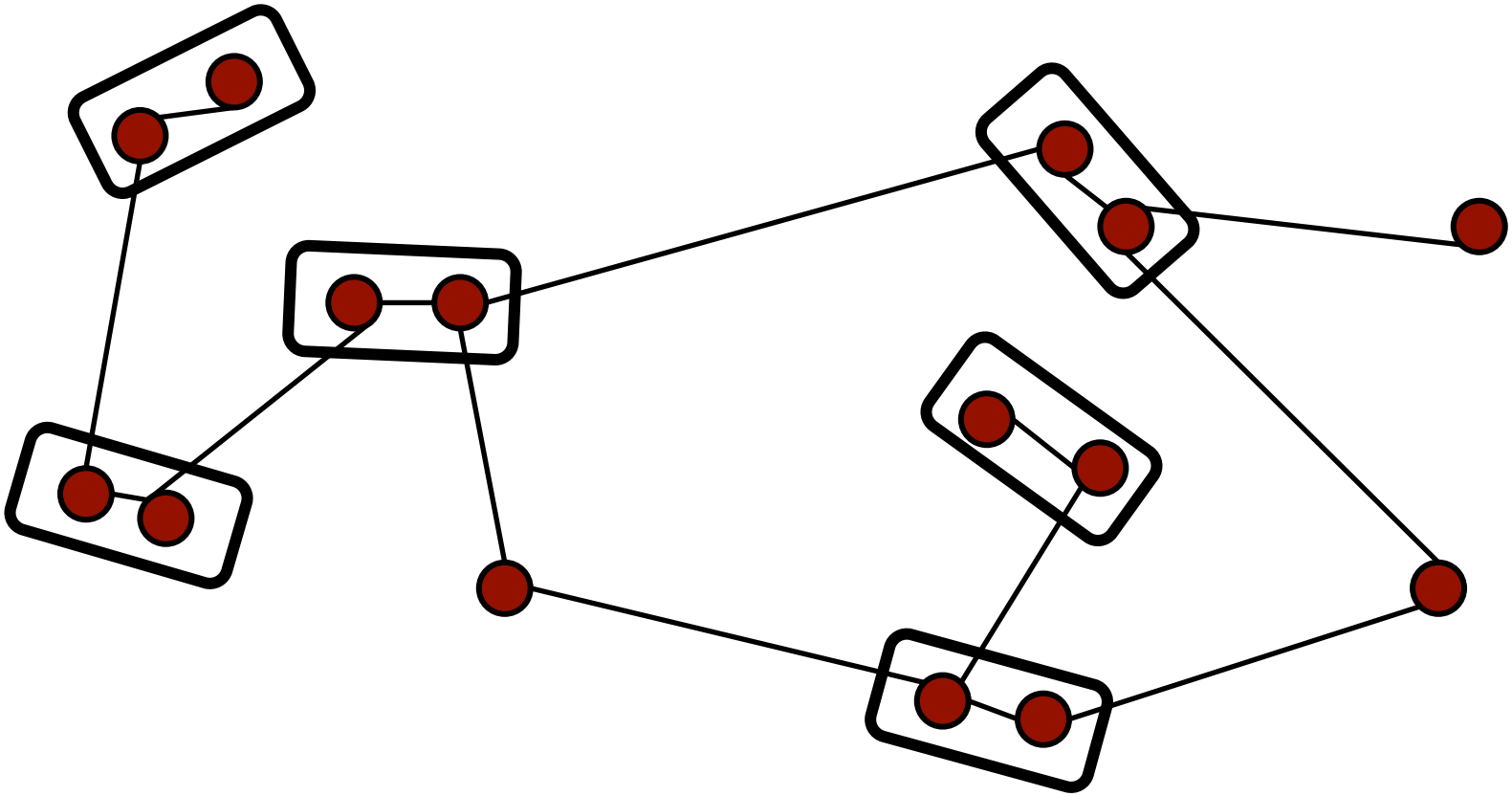
Divide and Conquer Algo

Sparsify: reduce the degree



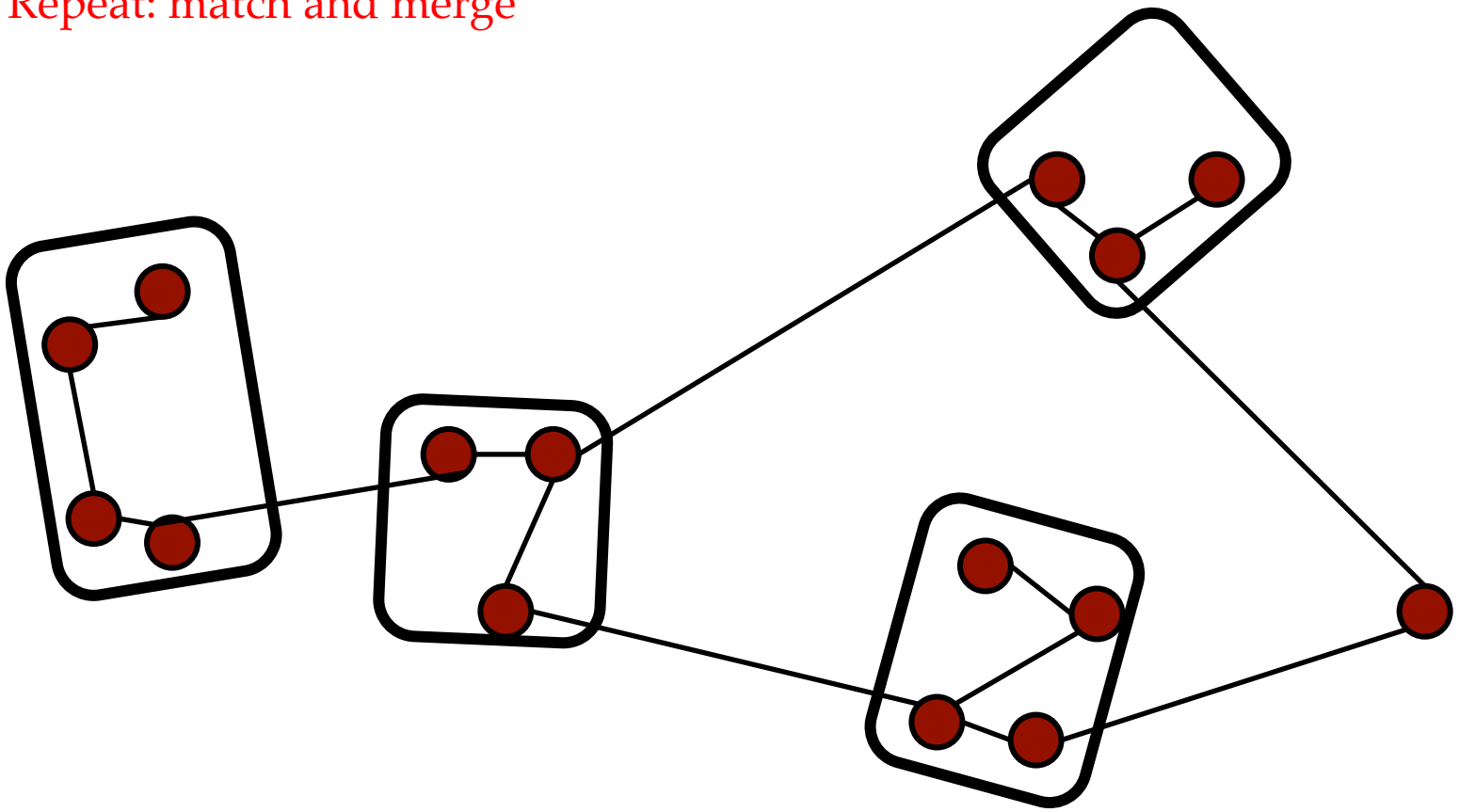
Divide and Conquer Algo

Repeat: match and merge



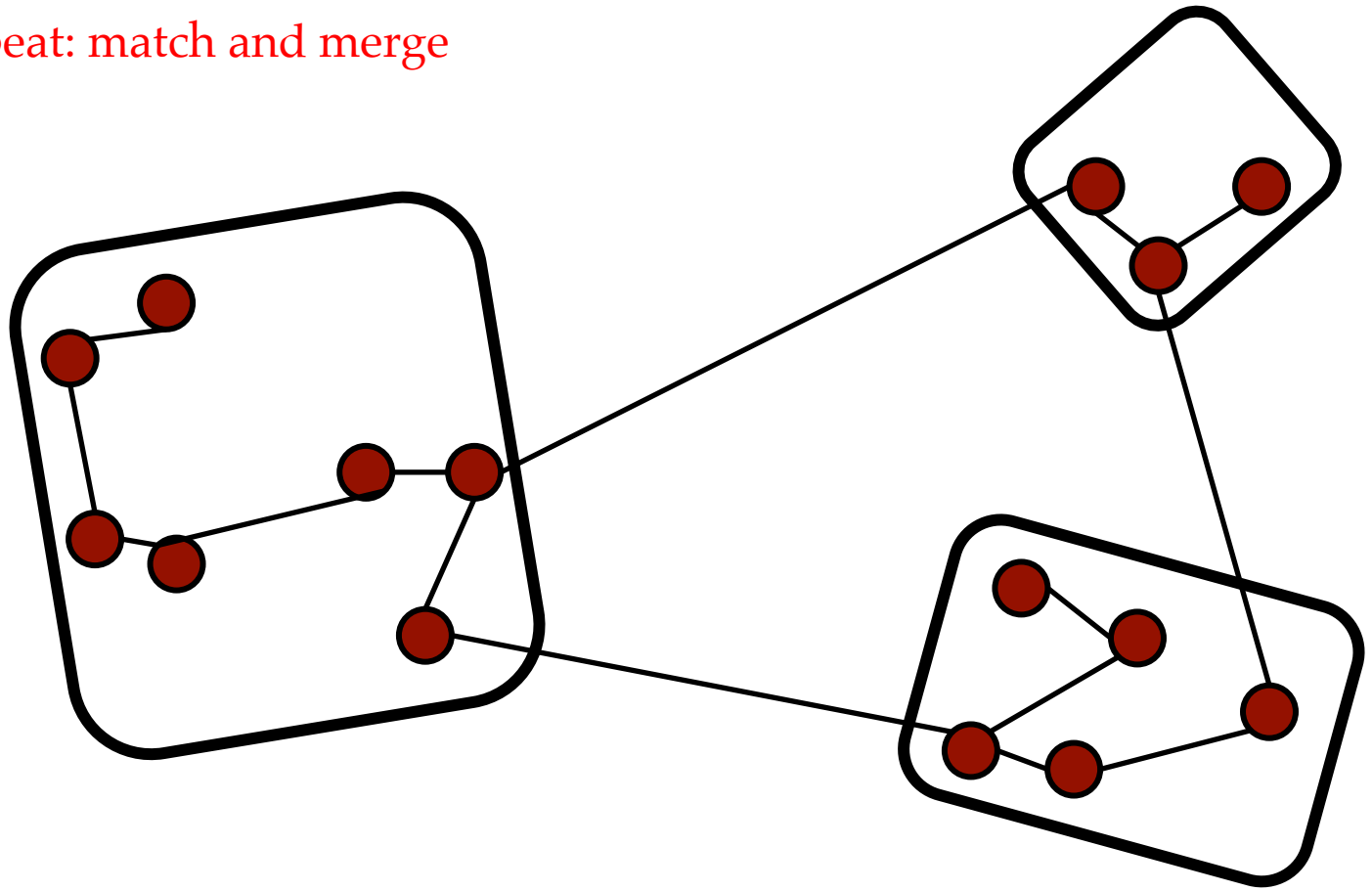
Divide and Conquer Algo

Repeat: match and merge



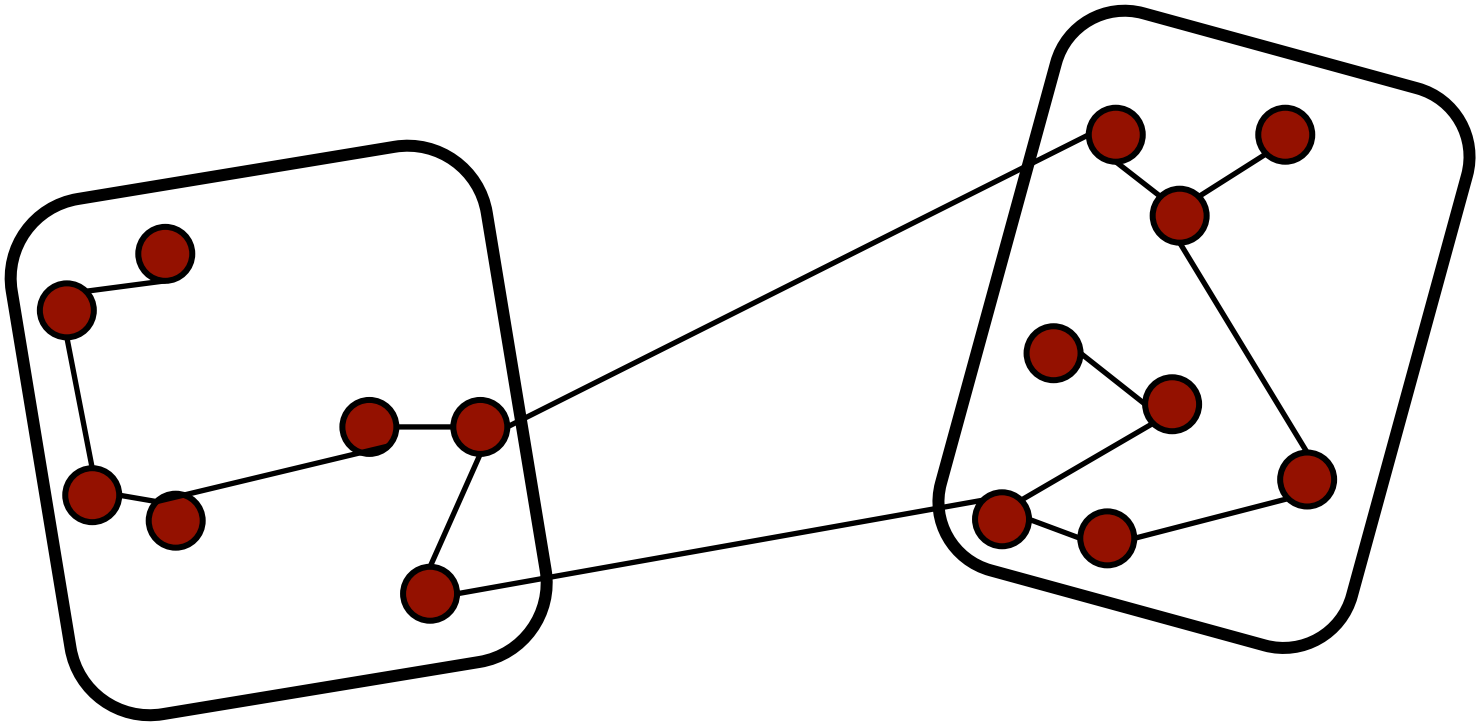
Divide and Conquer Algo

Repeat: match and merge



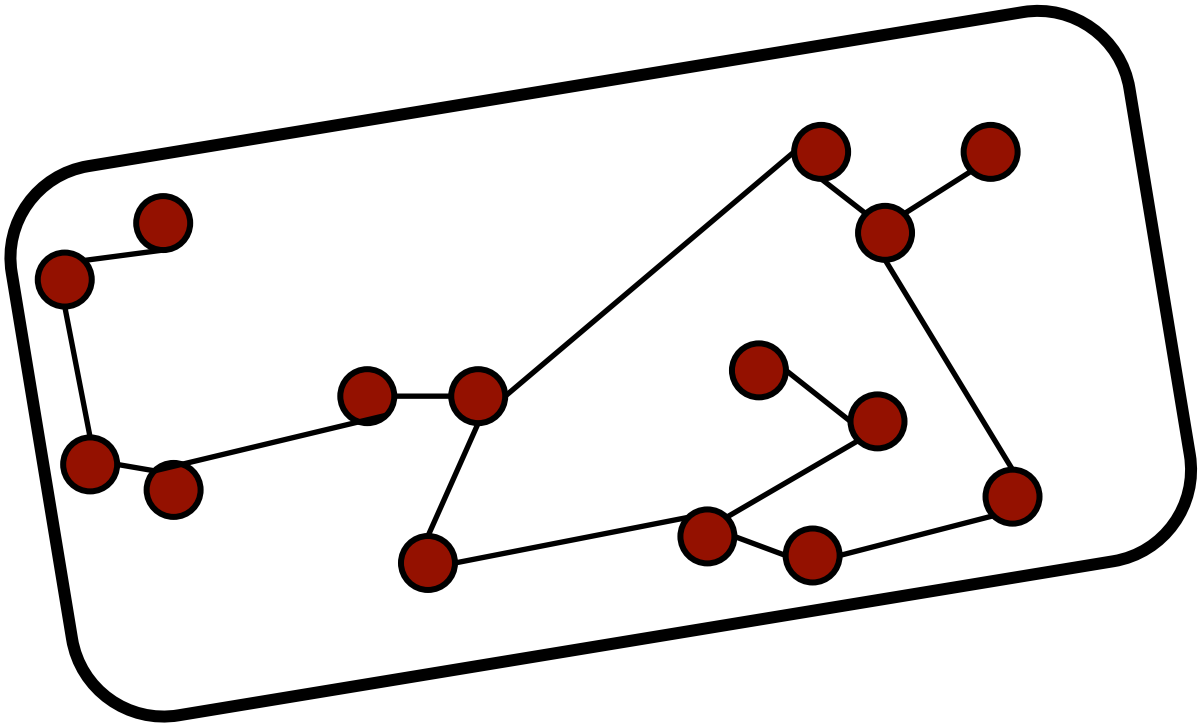
Divide and Conquer Algo

Repeat: match and merge



Divide and Conquer Algo

Repeat: match and merge



Efficiency Analysis

Merging:

- Random walks: O(log n) cost
- Bridge doubling: O(log n) iterations
- Overall: O(polylog n) time to merge topologies.

Divide-and-Conquer:

- Collection has small diameter ➔ O(log n) cost to coordinate.
- Collection merging: O(polylog n) cost per merge step.
- Number of iterations: O(log n) matchings
- Overall: O(polylog n) time to form overlay.

Help among Yourselves?

- **Perspective Students** (having CGPA above 8.5 and above)
- **Promising Students** (having CGPA above 6.5 and less than 8.5)

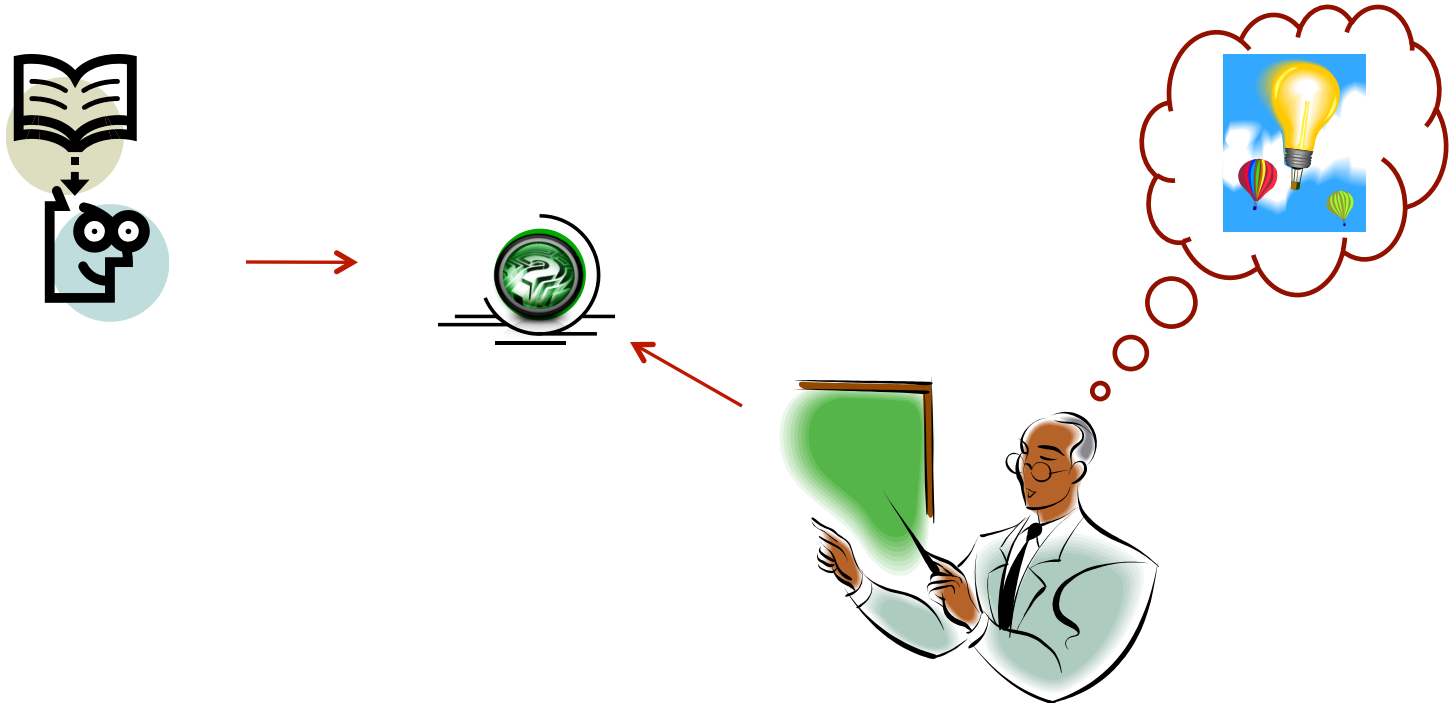
- **Needy Students** (having CGPA less than 6.5)
  - Can the above group help these students? (Your work will also be rewarded)

- You may grow a culture of **collaborative learning** by helping the needy students

Assistance

- You may post your questions to me at any time
- You may meet me in person on available time or with an appointment
- TA s would assist you to clear your doubts.
- You may leave me an email any time (email is the best way to reach me faster)

Thanks …



… Questions ???