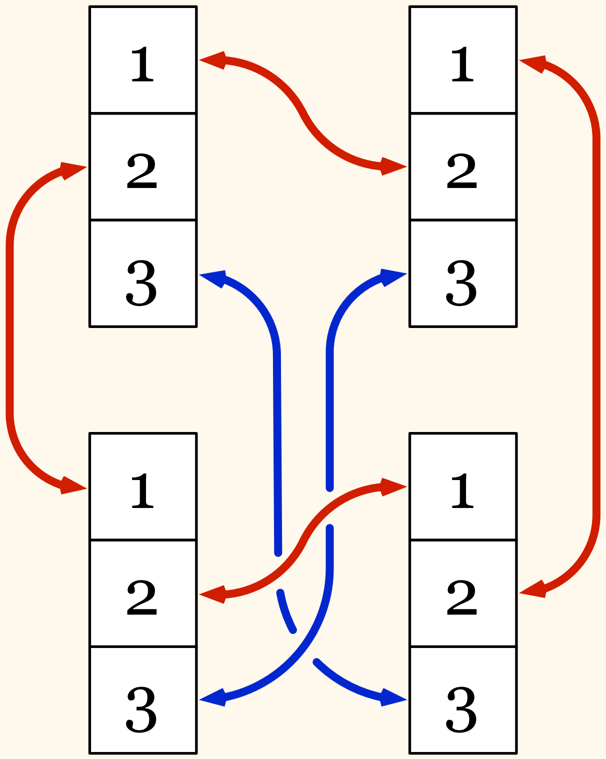
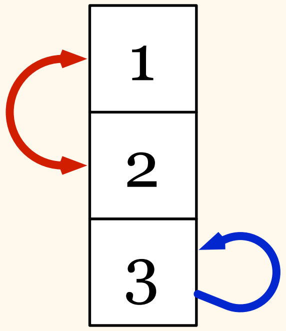
Deterministic Distributed Algorithms

www.iki.fi/suo/dda-2014



Jukka Suomela

University of Helsinki,
March–April 2014



Introduction

DDA Course
week 1

Practicalities

- Read the course web page:
  www.iki.fi/suo/dda-2014
- Pay attention to:
  - *course content* — theory, not practice
  - *course format* — not a typical lecture course
  - *social media* — two online forums,
    you can also use these for *real-time feedback*!

Course Content

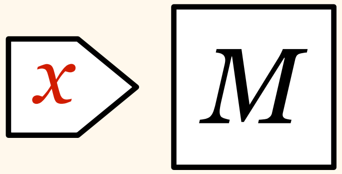
- Fundamental questions:
  - what can be computed?
  - what can be computed fast?
- Model of computation:
  - distributed systems

Traditional Perspective

Programmer:

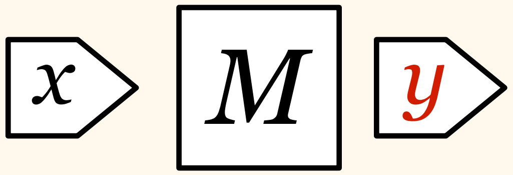
constructs
a machine

Adversary:



chooses any
valid input

Machine:



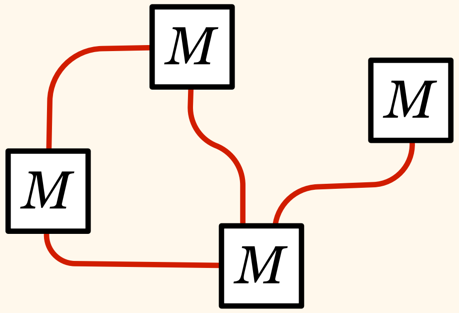
does computation,
prints a valid output

Distributed Algorithms

Programmer:

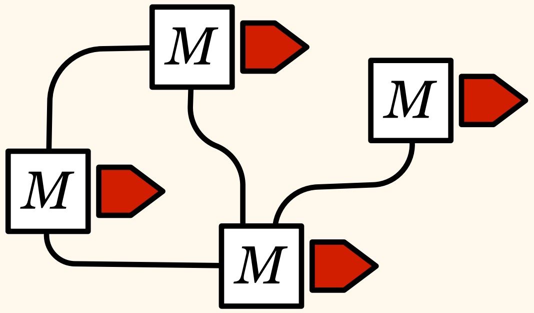
constructs
a machine

Adversary:



constructs
a *network*

Network:



does *communication*,
prints a valid output

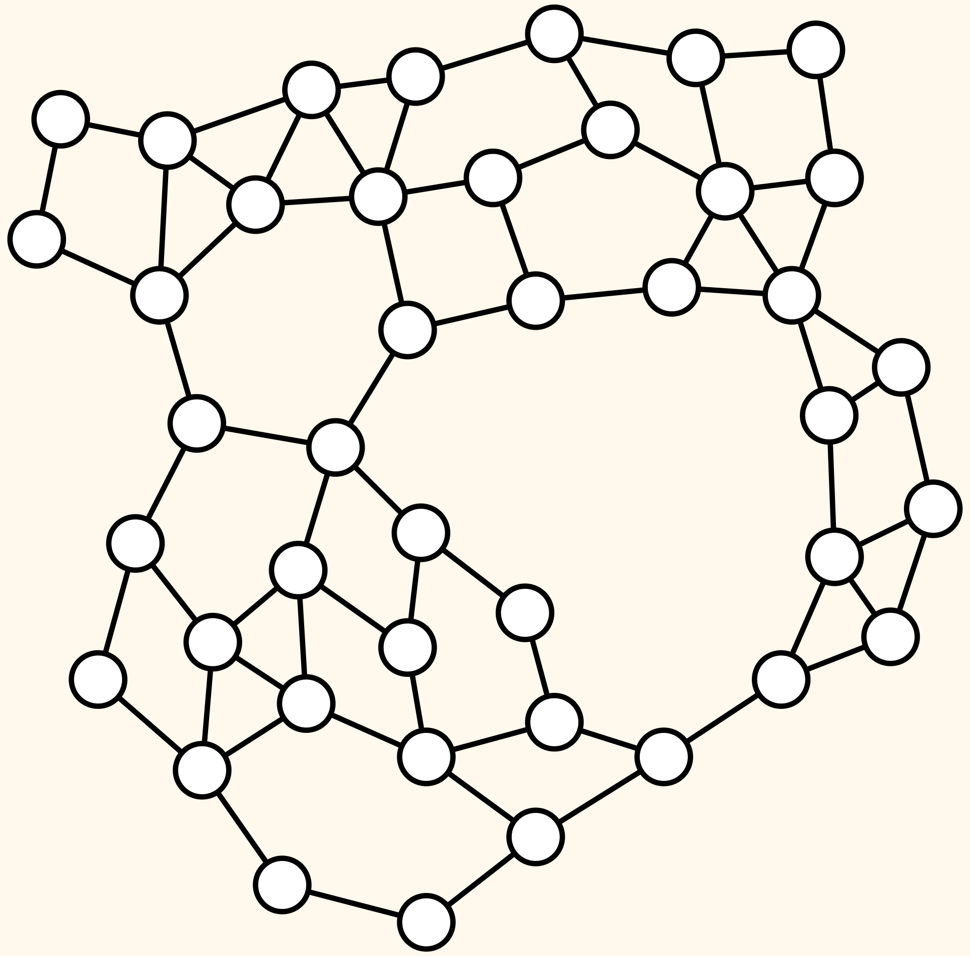
You Will Learn…

- A new mindset: how to reason about distributed and parallel systems
    - not a bad skill in the multi-core era
- Combinatorial optimisation
- Some math that has plenty of applications in computer science
    - graph theory, Ramsey theory, …

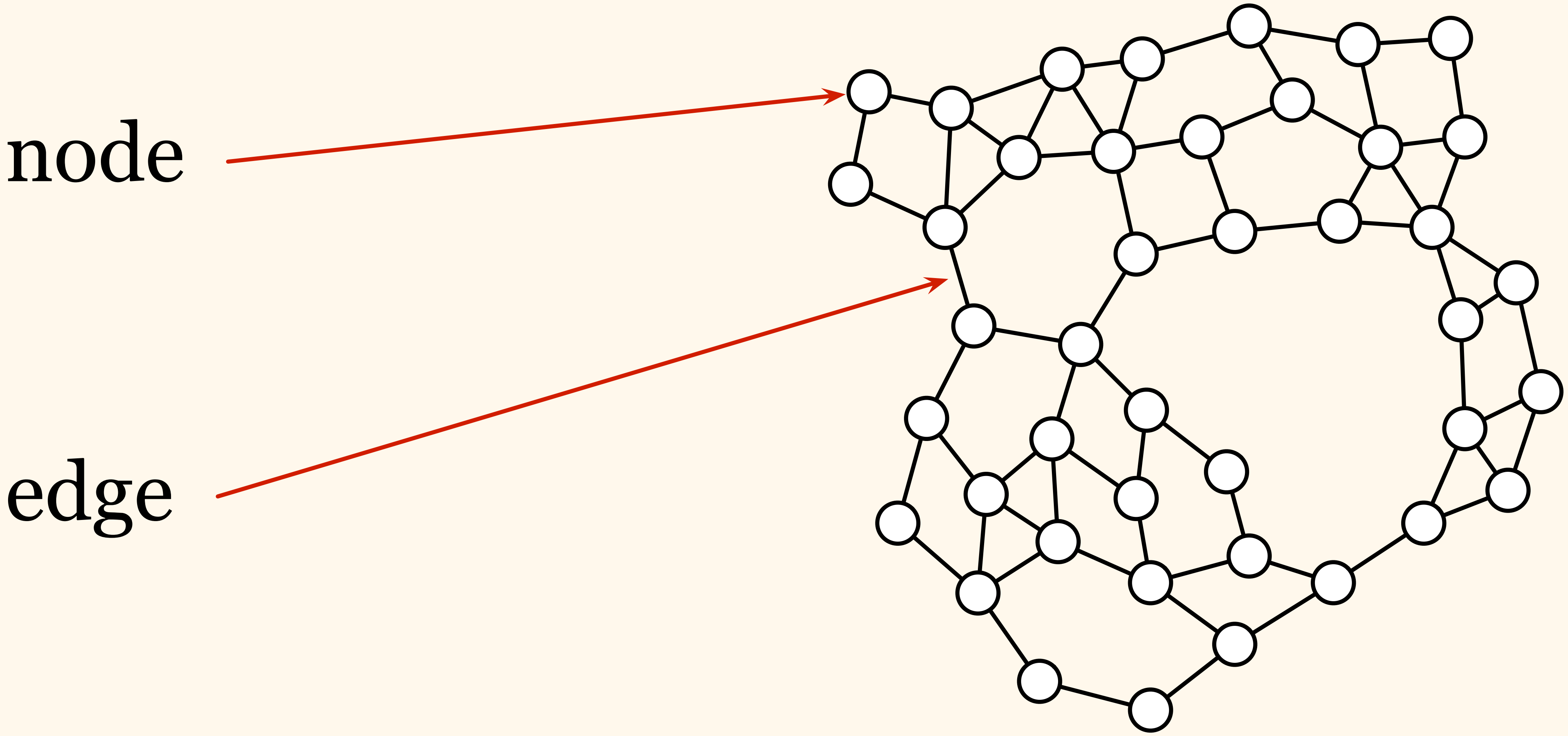
Plan: Two Models

- Week 1: some graph theory
- Weeks 2–4: *“port-numbering model”*
  - weeks 2 and 4: positive results,
    week 3: negative results
- Weeks 5–6: *“unique identifiers”*
  - week 5: positive results,
    week 6: negative results

Graphs



Graphs



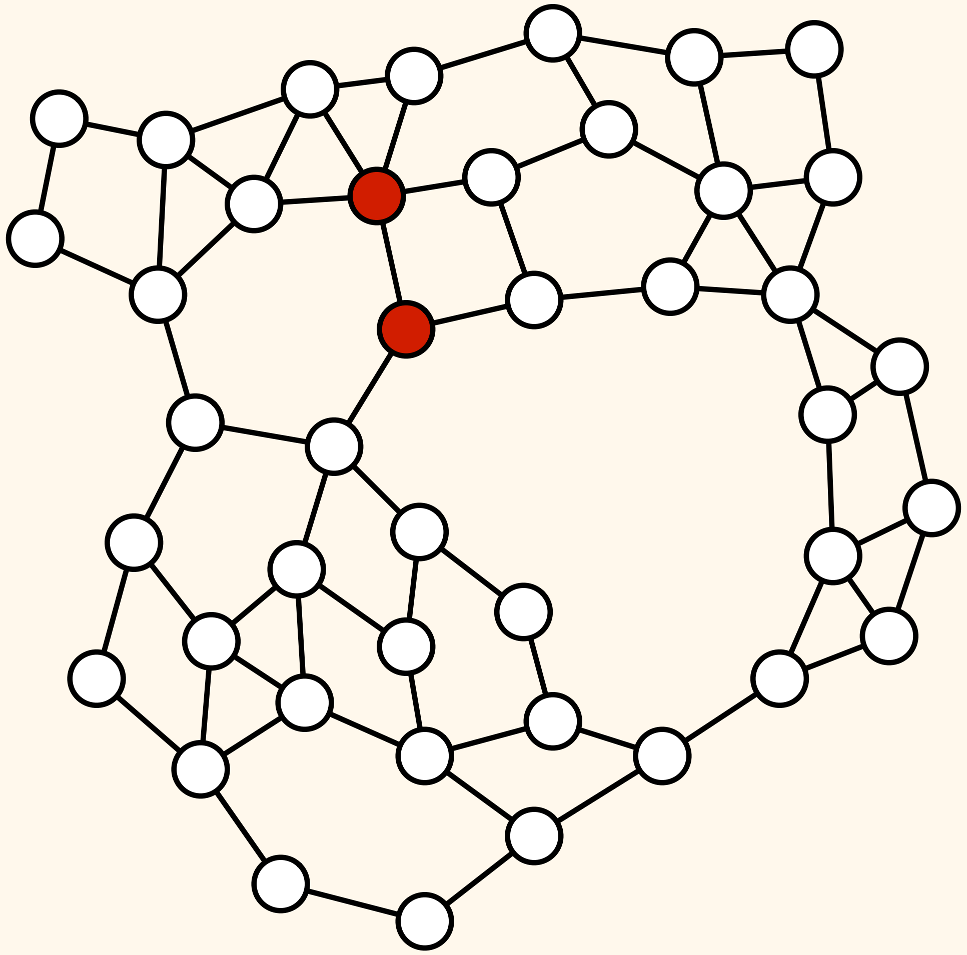
node

edge

Graphs

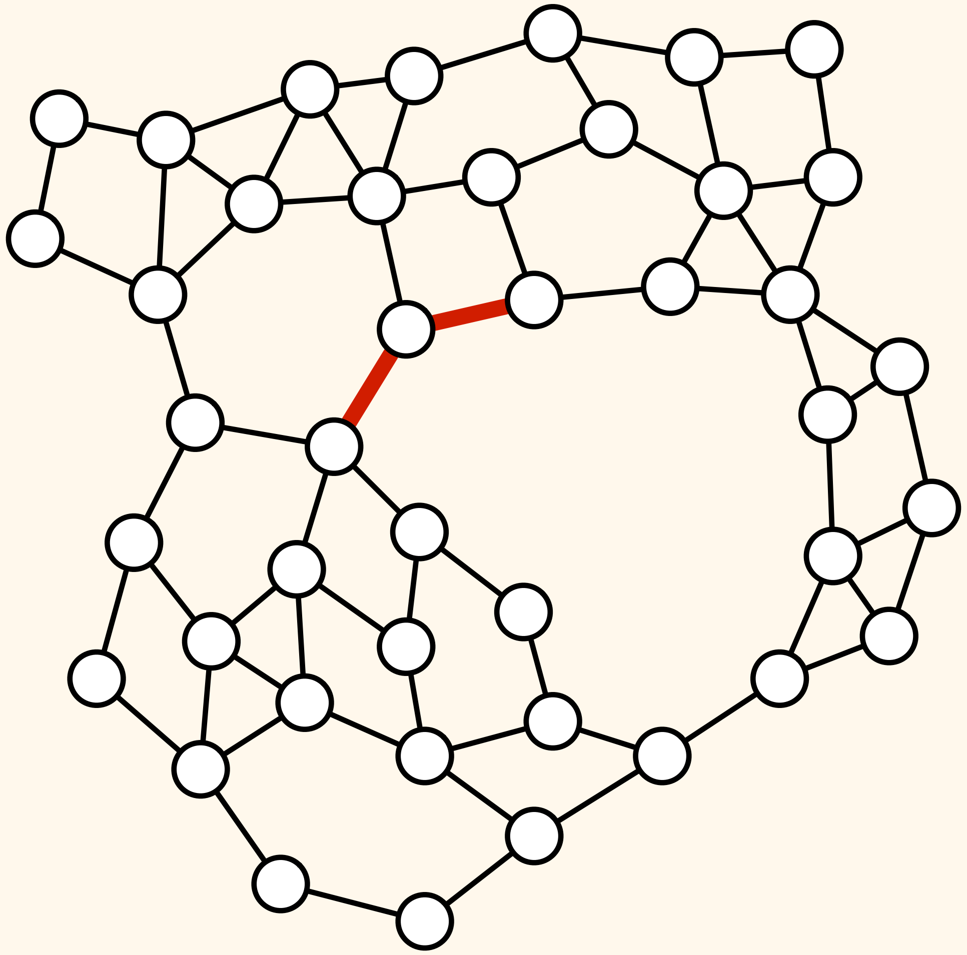
adjacent nodes

neighbours



Graphs

adjacent edges



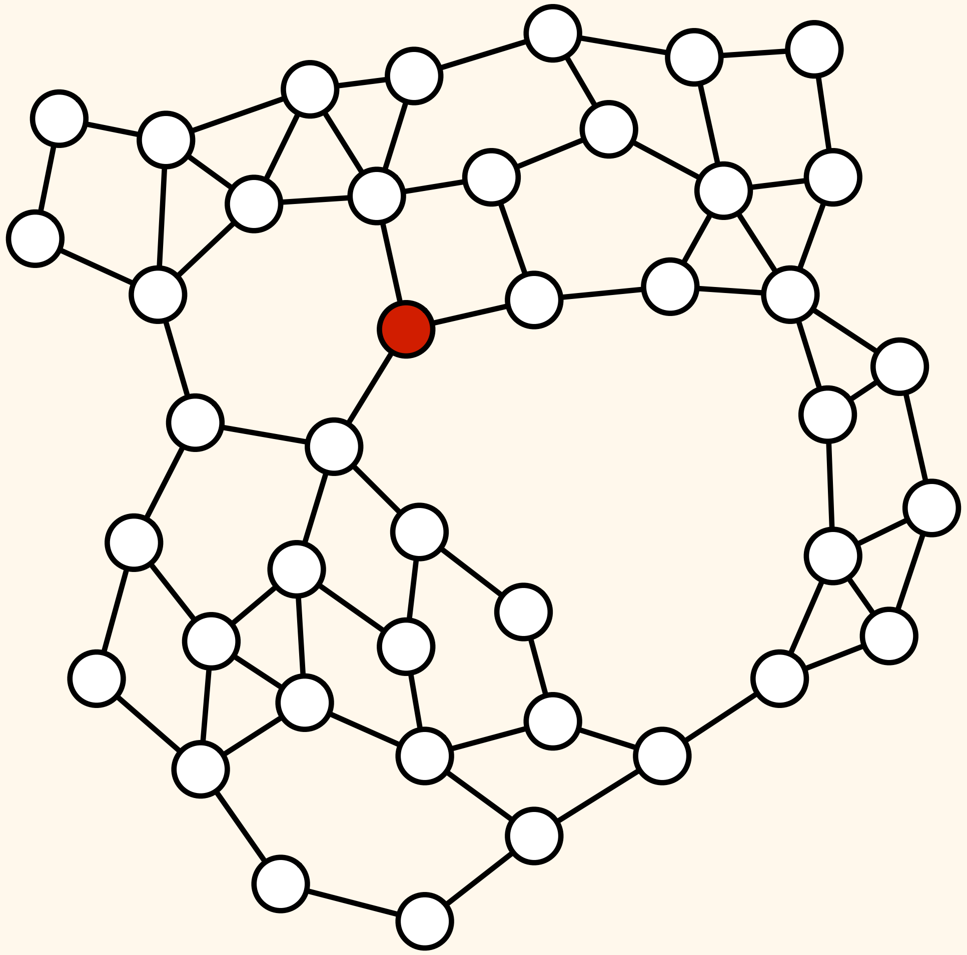
Graphs

node with 3 neighbours

adjacent to 3 nodes

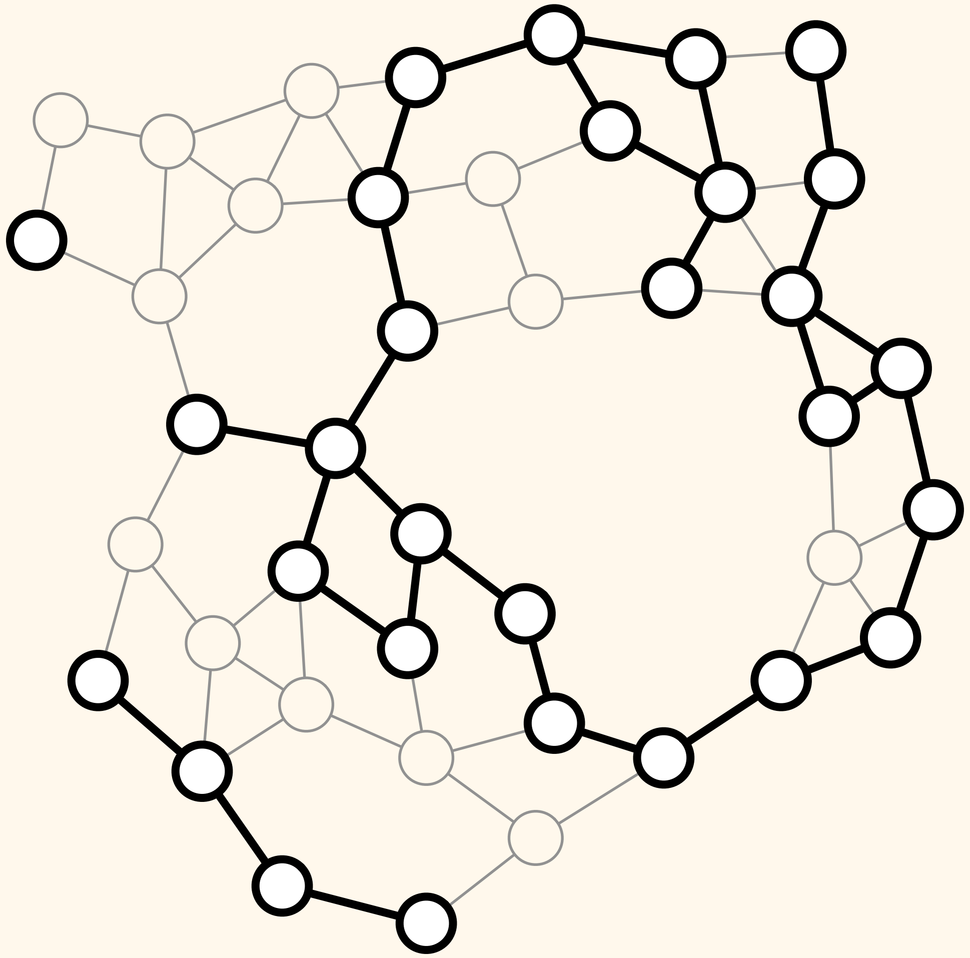
incident to 3 edges

degree is 3



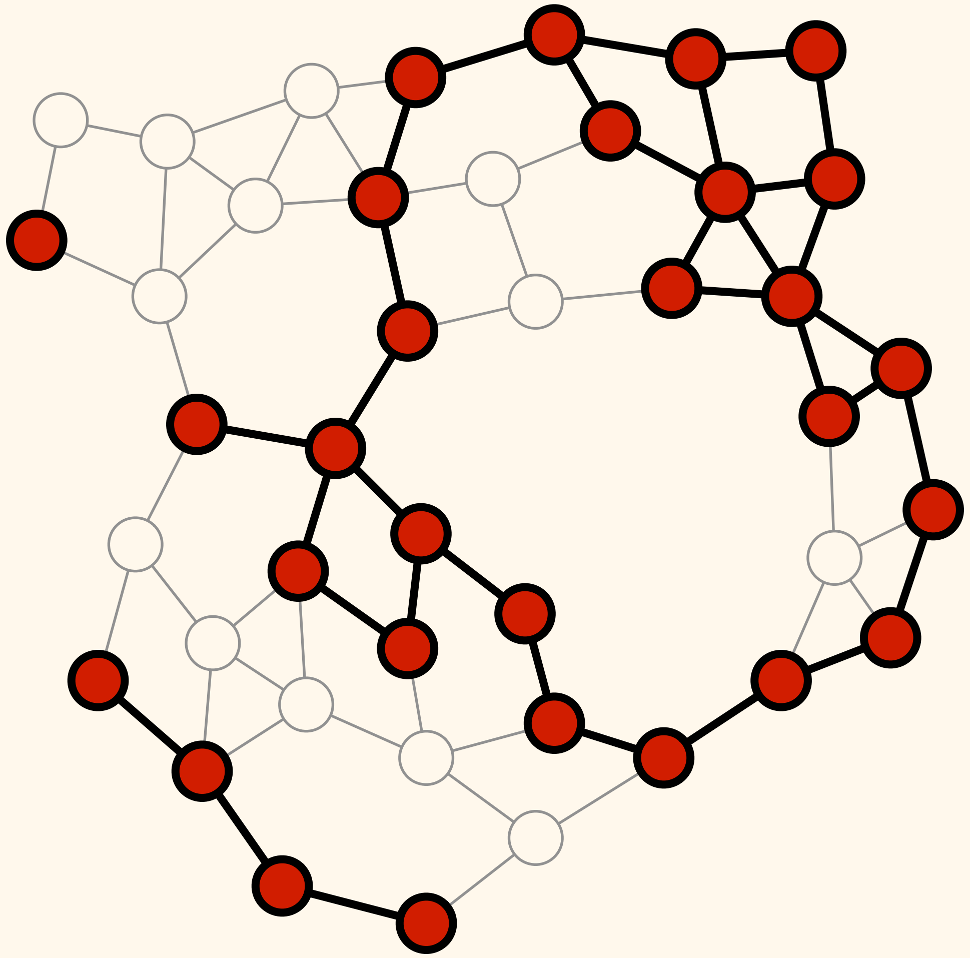
Graphs

subgraph



Graphs

subgraph induced
by the red nodes

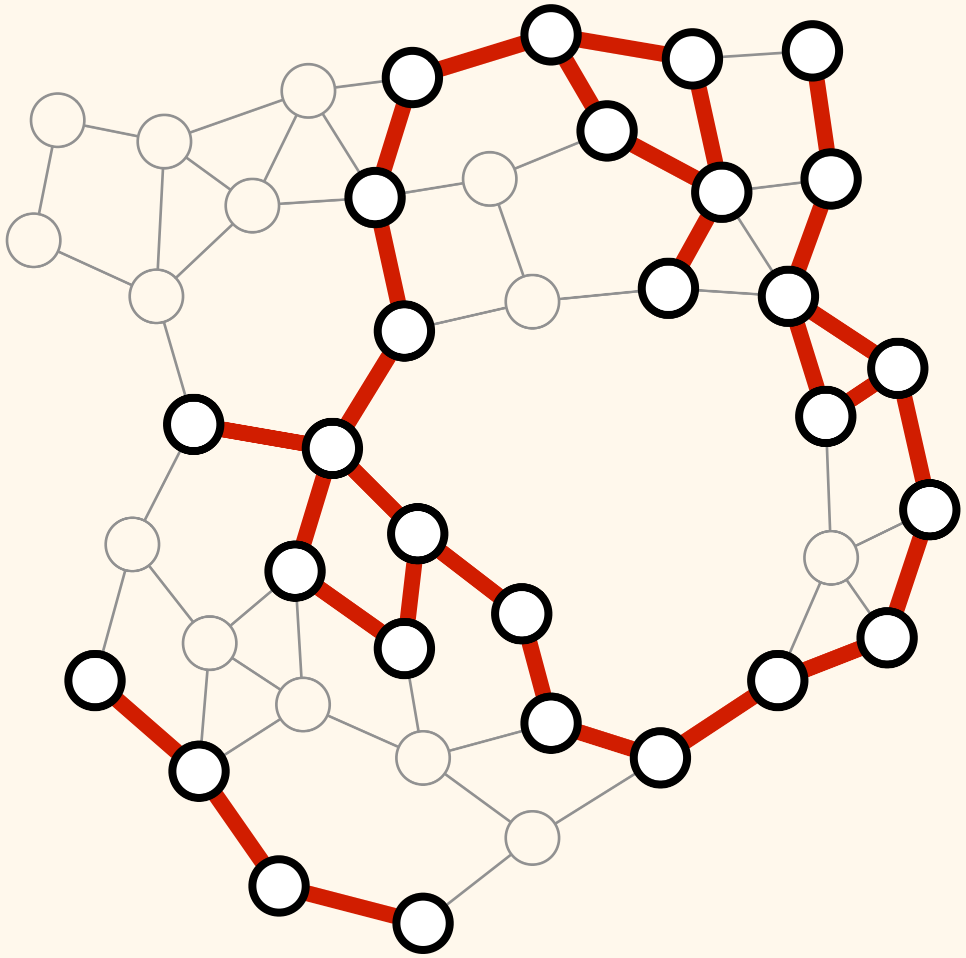


all red nodes

all edges that join
a pair of red nodes

Graphs

subgraph induced
by the red edges



all red edges

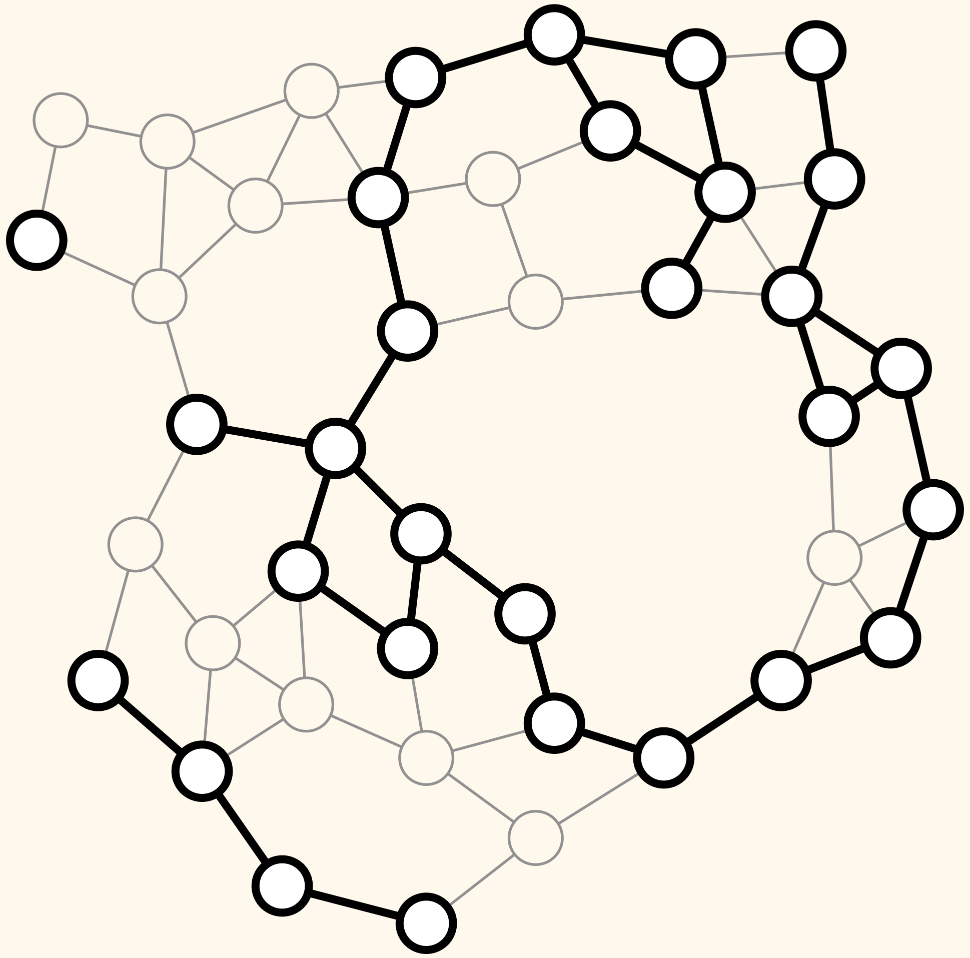
all nodes that are
incident to red edges

Graphs

not a node-induced subgraph

not an edge-induced subgraph

not a spanning subgraph



Graphs

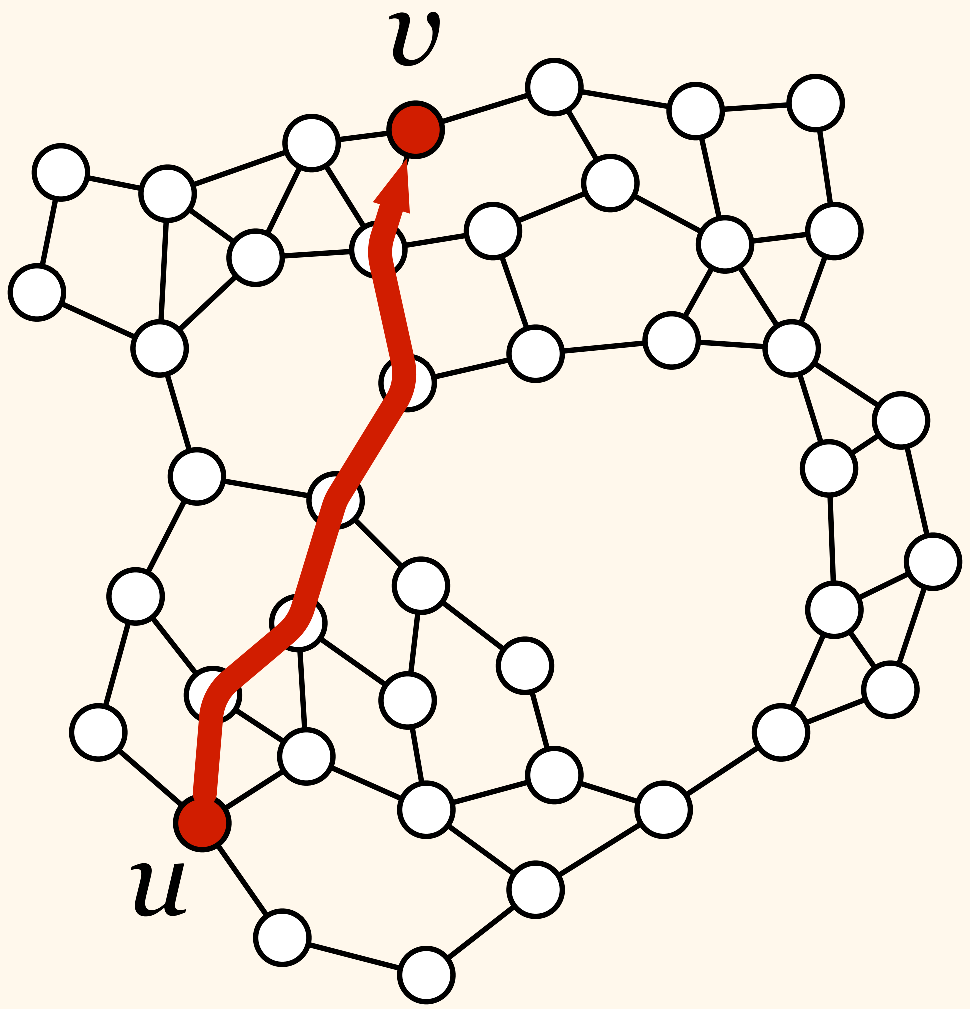
a shortest path from *u* to *v*

length 6

(six *edges*, seven *nodes*)

dist(u, v) = 6

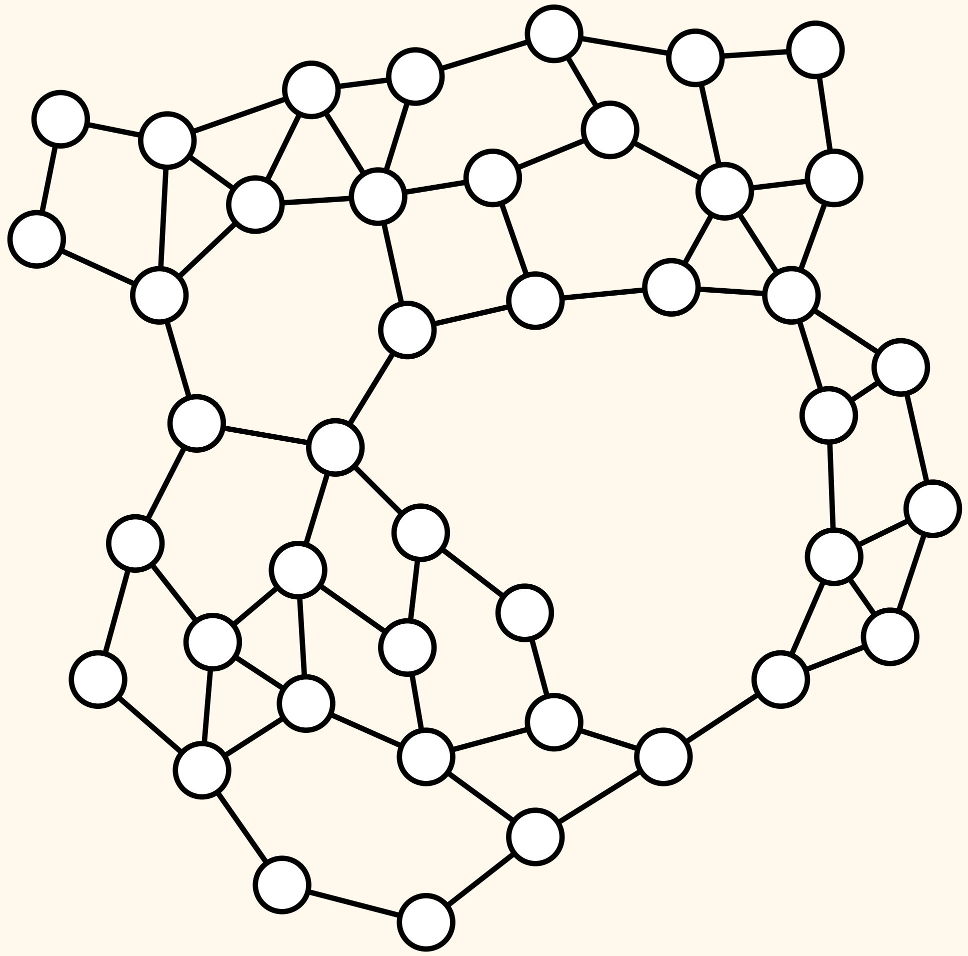
diameter ≥ 6



Graphs

connected graph

one connected component

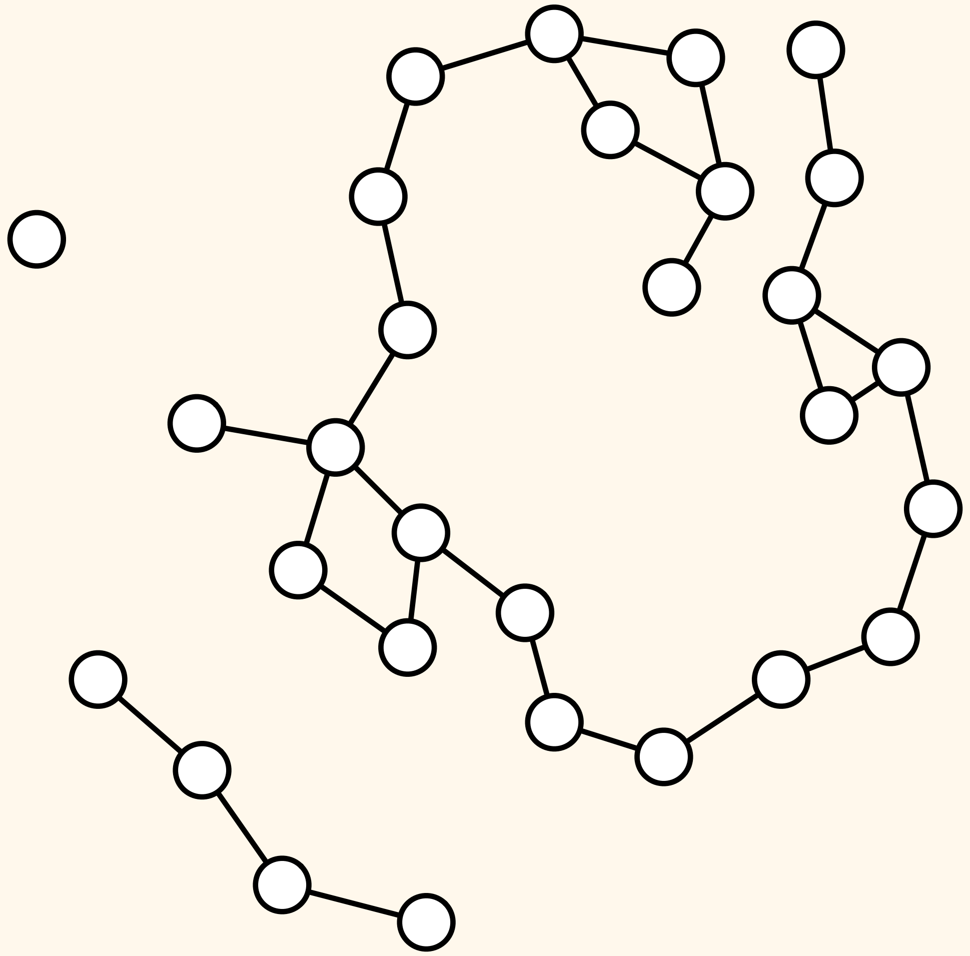


Graphs

not a connected graph

three connected components

one isolated node

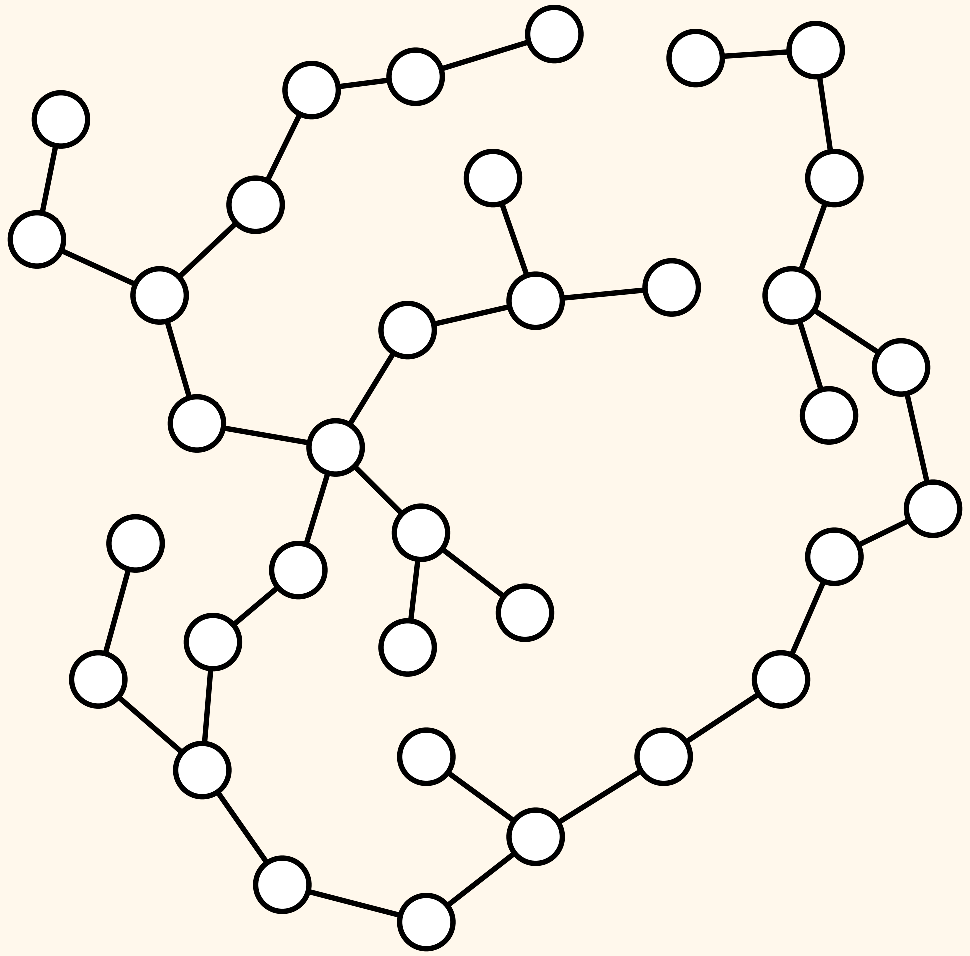


Graphs

tree

connected

no cycles

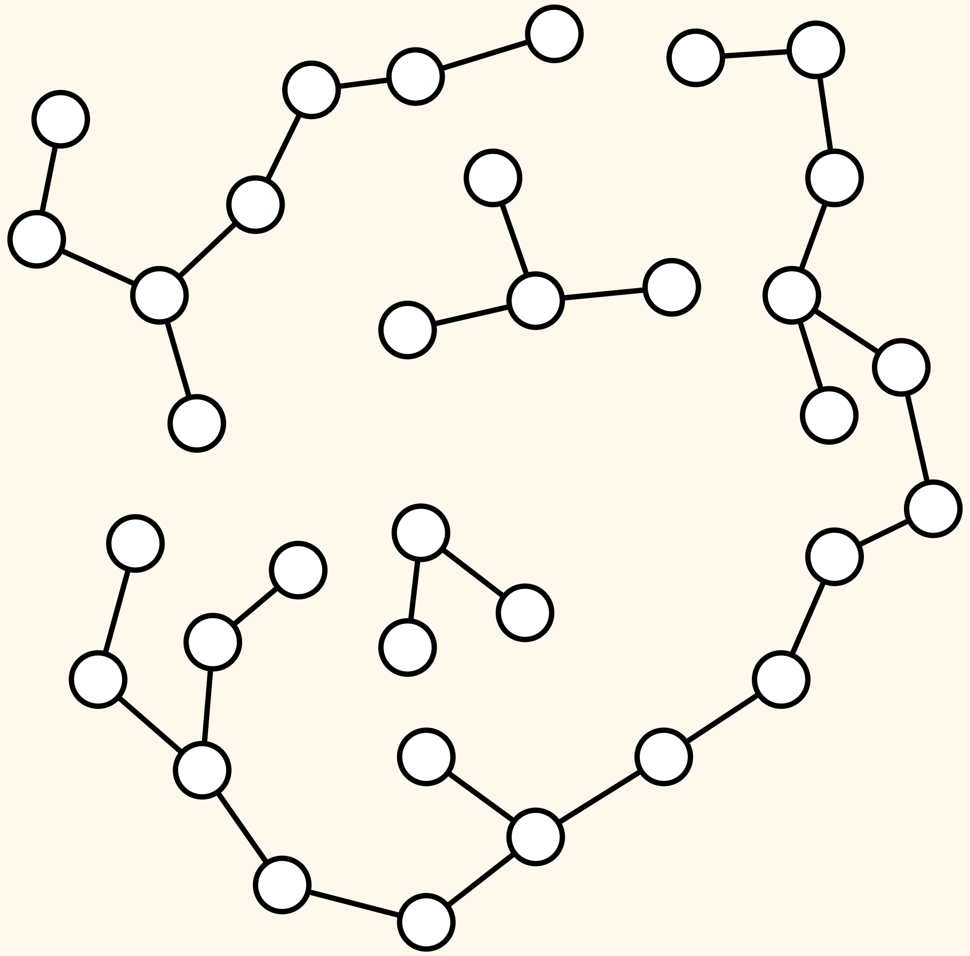


Graphs

forest

four connected components

no cycles

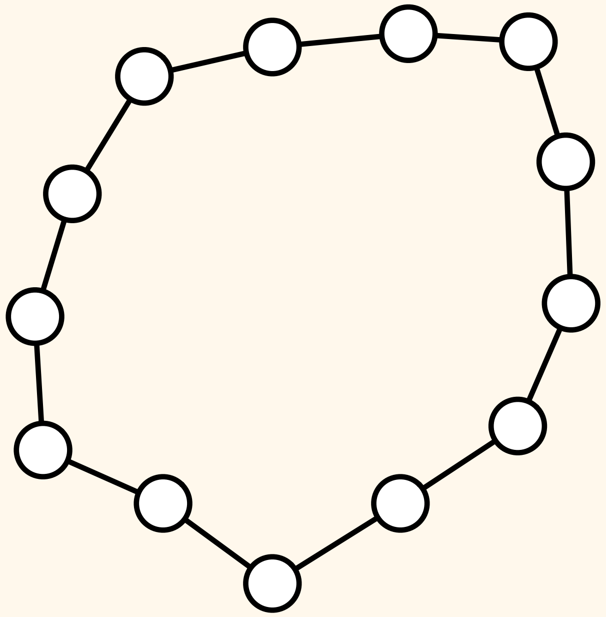


Graphs

cycle graph

connected

2-regular



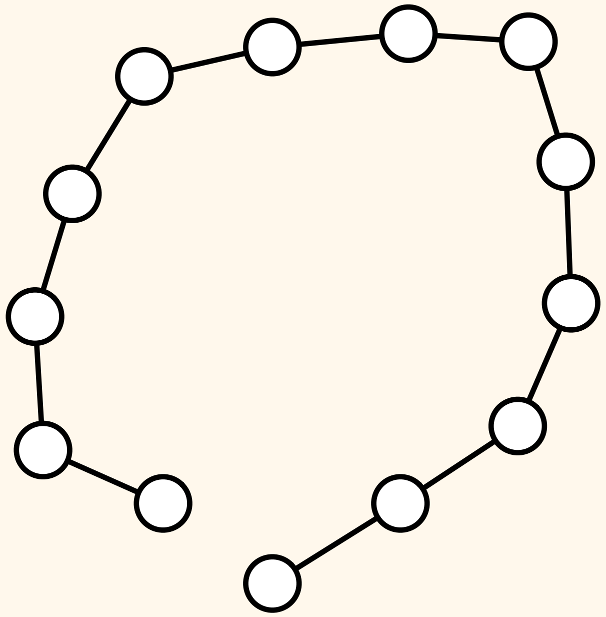
Graphs

path graph

tree

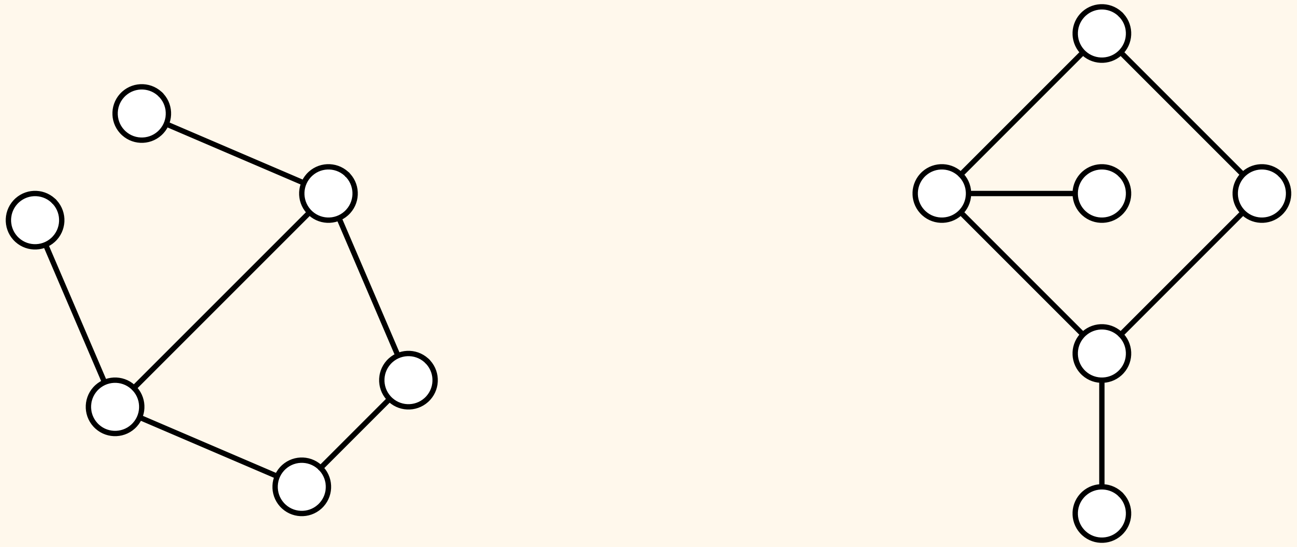
connected

maximum degree 2



Graphs

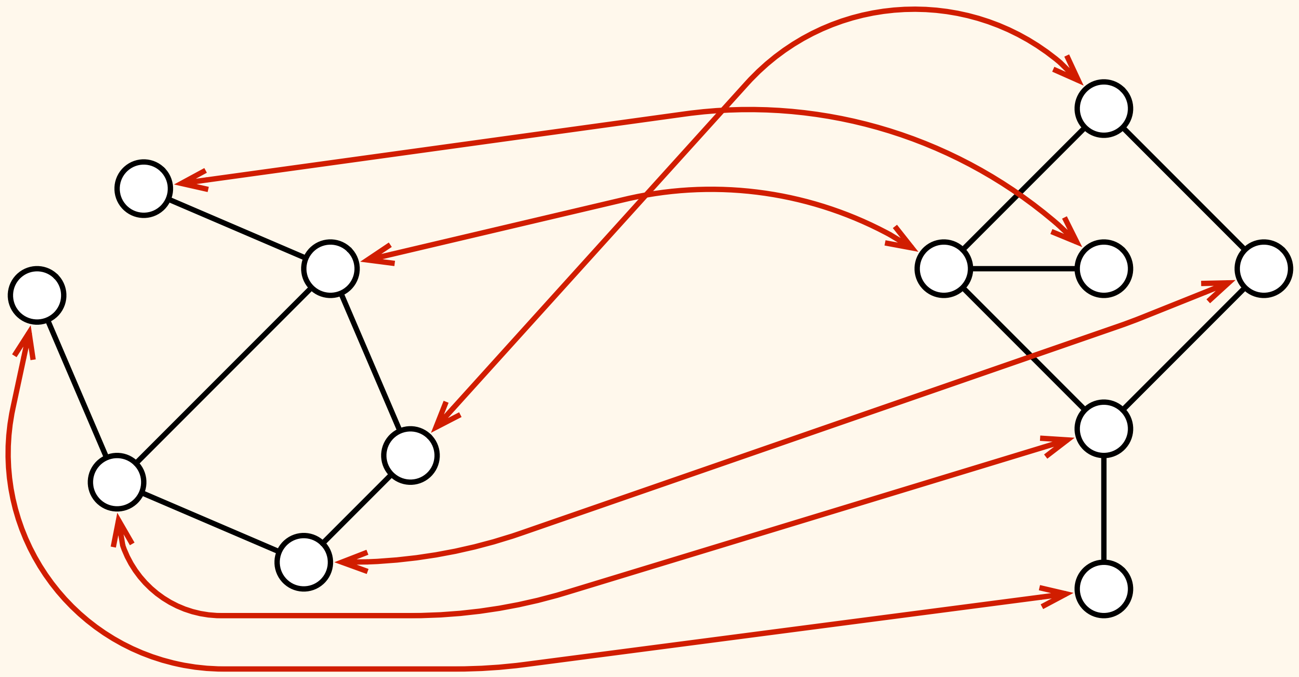
two isomorphic graphs



Graphs

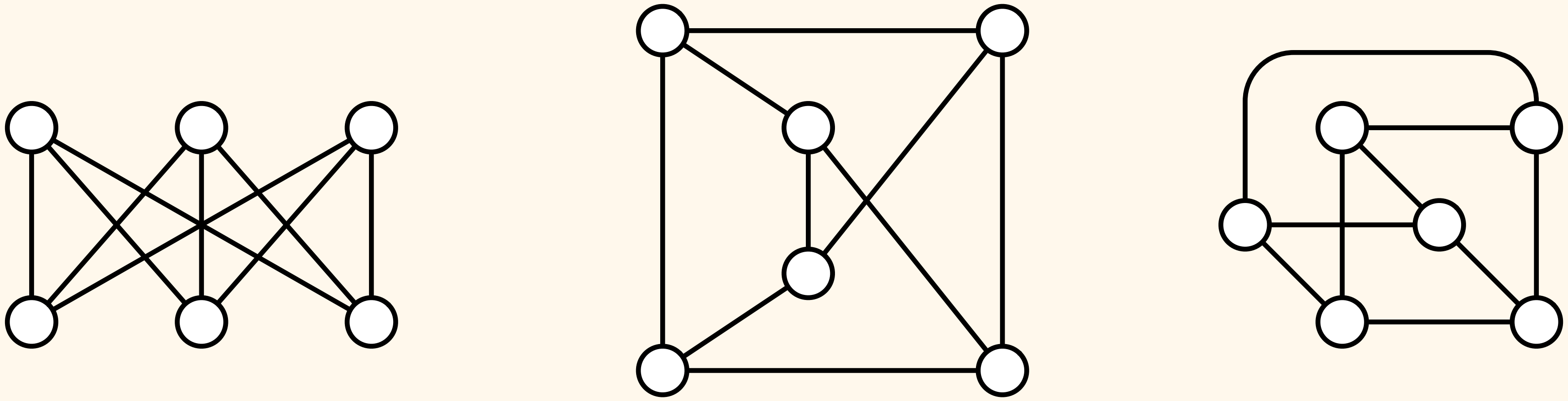
two isomorphic graphs

bijection that preserves the structure



Graphs

three isomorphic graphs



Graph Problems

Graph Problems

- Recall the definitions:
  - independent set — vertex cover — dominating set
  - matching — edge cover — edge dominating set
  - vertex colouring — domatic partition
  - edge colouring — edge domatic partition
- Examples in the course material…

Optimisation

- Maximisation problems:
  - *maximal* = cannot add anything
  - *maximum* = largest possible size
  - α-approximation = at least $1/\alpha$ times maximum
- Example: independent set
  - maximal is trivial to find greedily,
    maximum may be very difficult to find

Optimisation

- Minimisation problems:
    - *minimal* = cannot remove anything
    - *minimum* = smallest possible size
    - α-approximation = at most α times minimum
- Example: vertex cover
    - minimal is trivial to find greedily, minimum may be very difficult to find

Optimisation

Terminology:

“α-approximation of minimum vertex cover”

implies two properties:

1. vertex cover
2. at most α times as large as minimum vertex cover

Approximations are always feasible solutions!