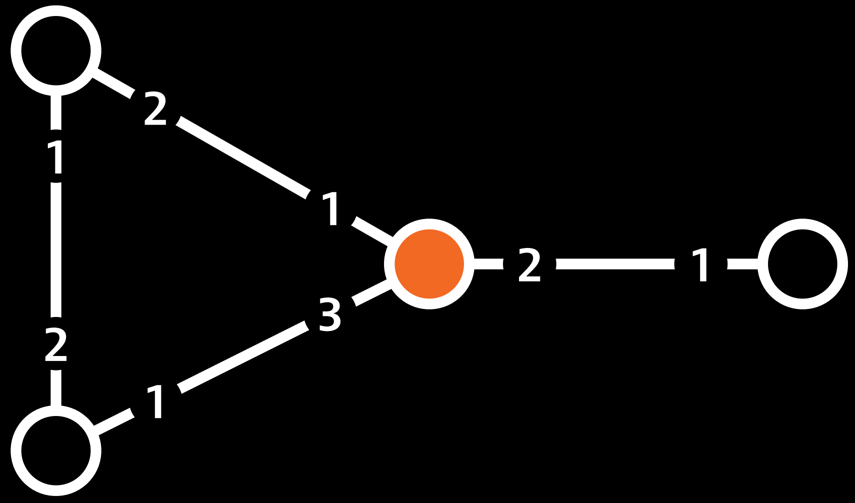
Aalto University

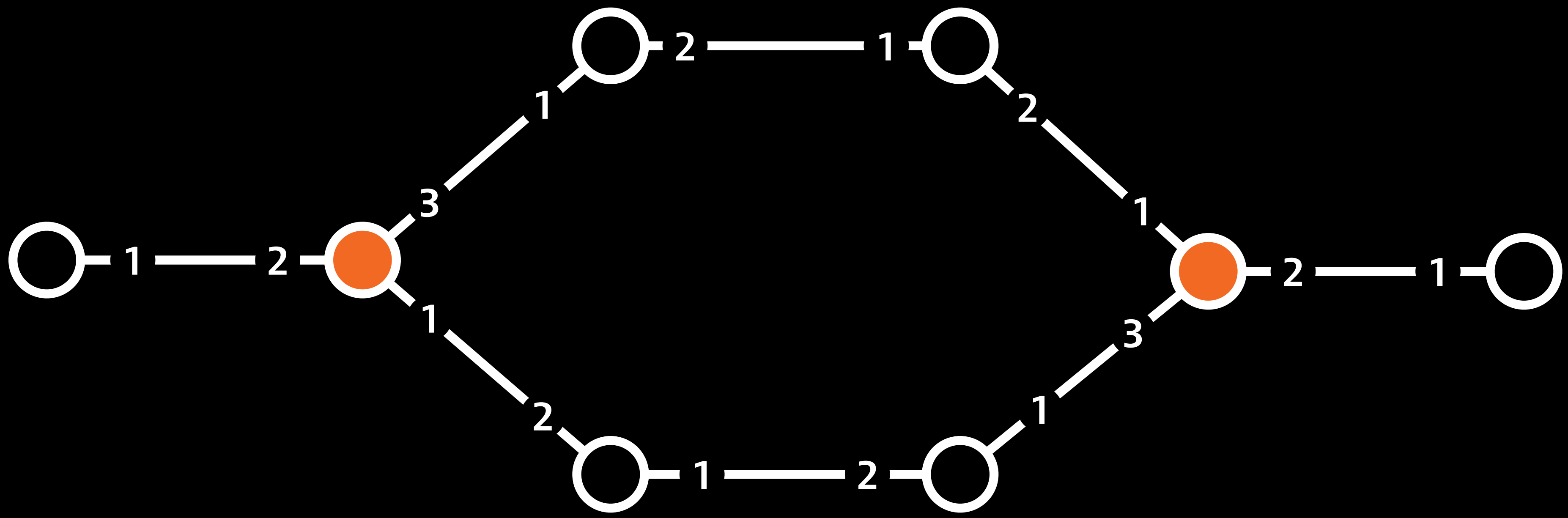
Distributed Algorithms 2024

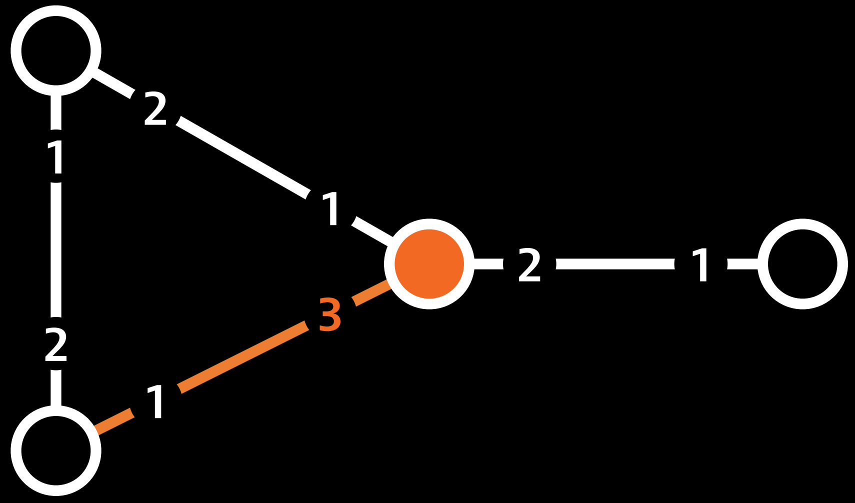
7 Covering maps



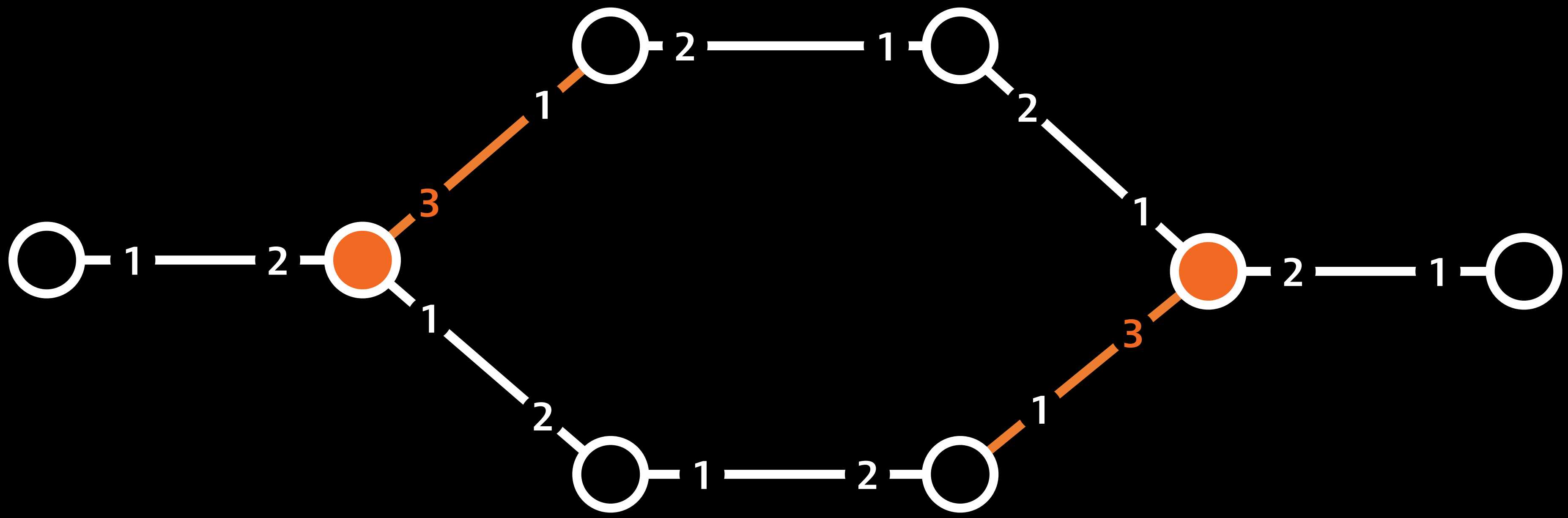
You are in a room
with three doors,
labeled 1, 2, and 3.

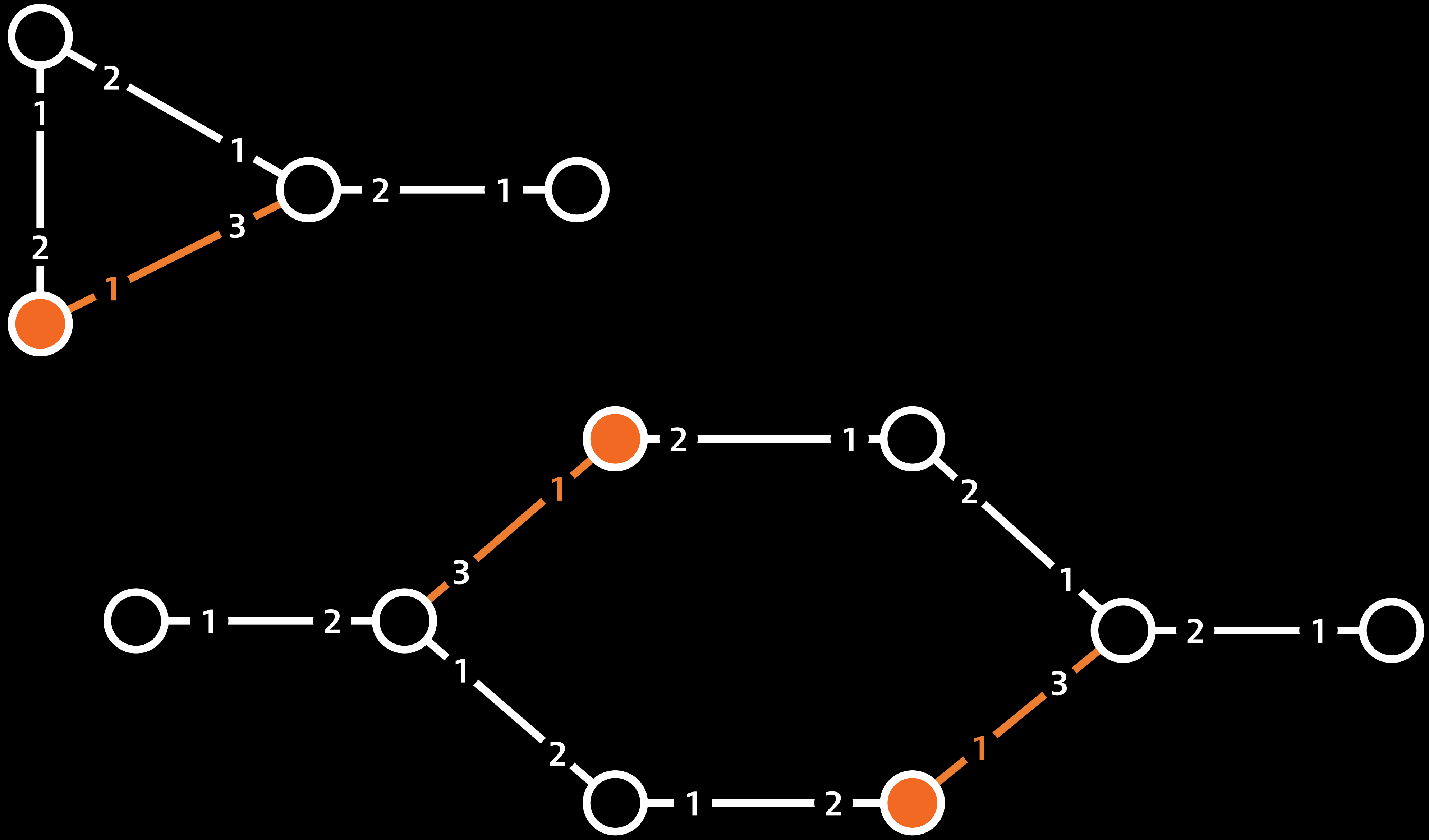
> _

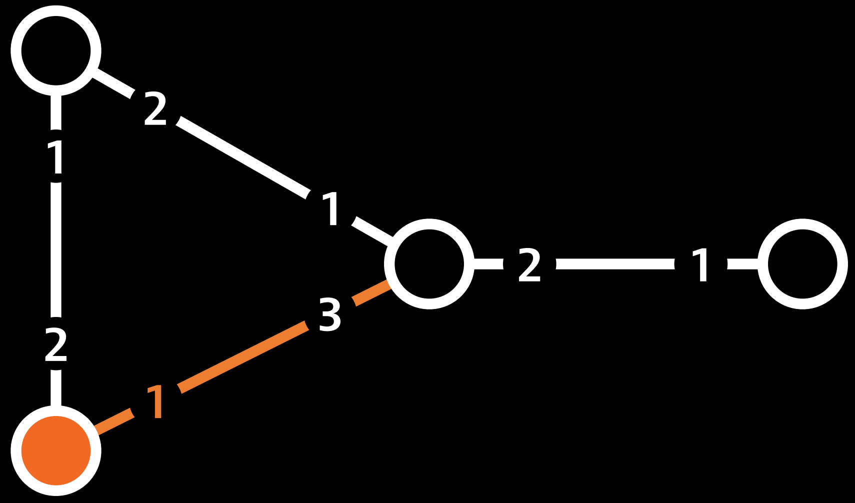




You are in a room
with three doors,
labeled 1, 2, and 3.
> open door 3

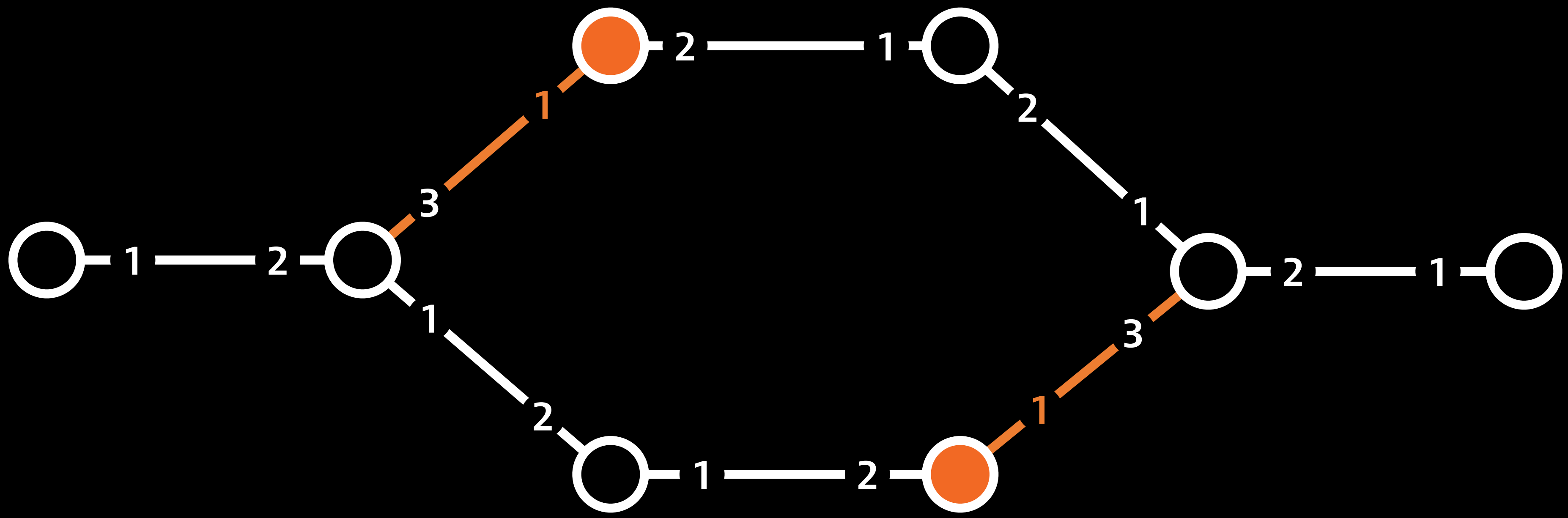


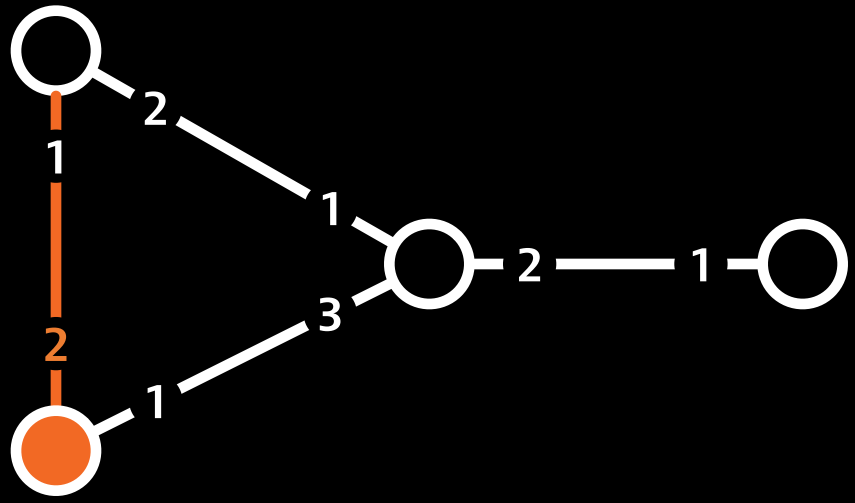




You enter a room with two doors, labeled 1 and 2. You just came in through doorway 1.

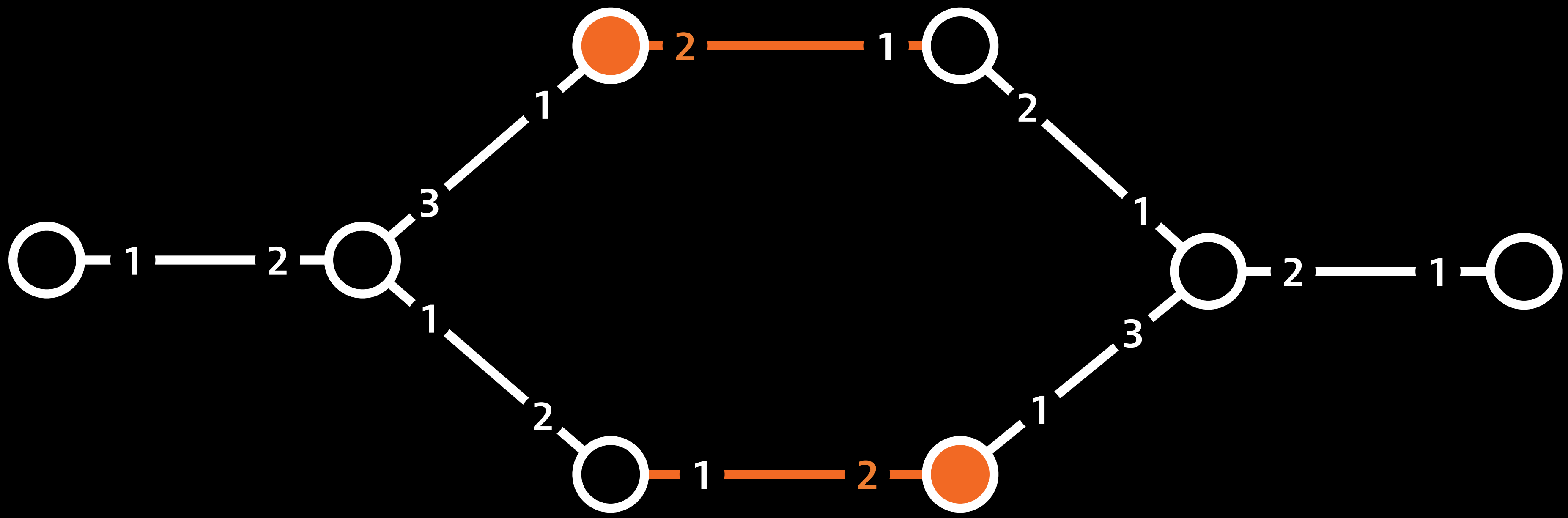
> _

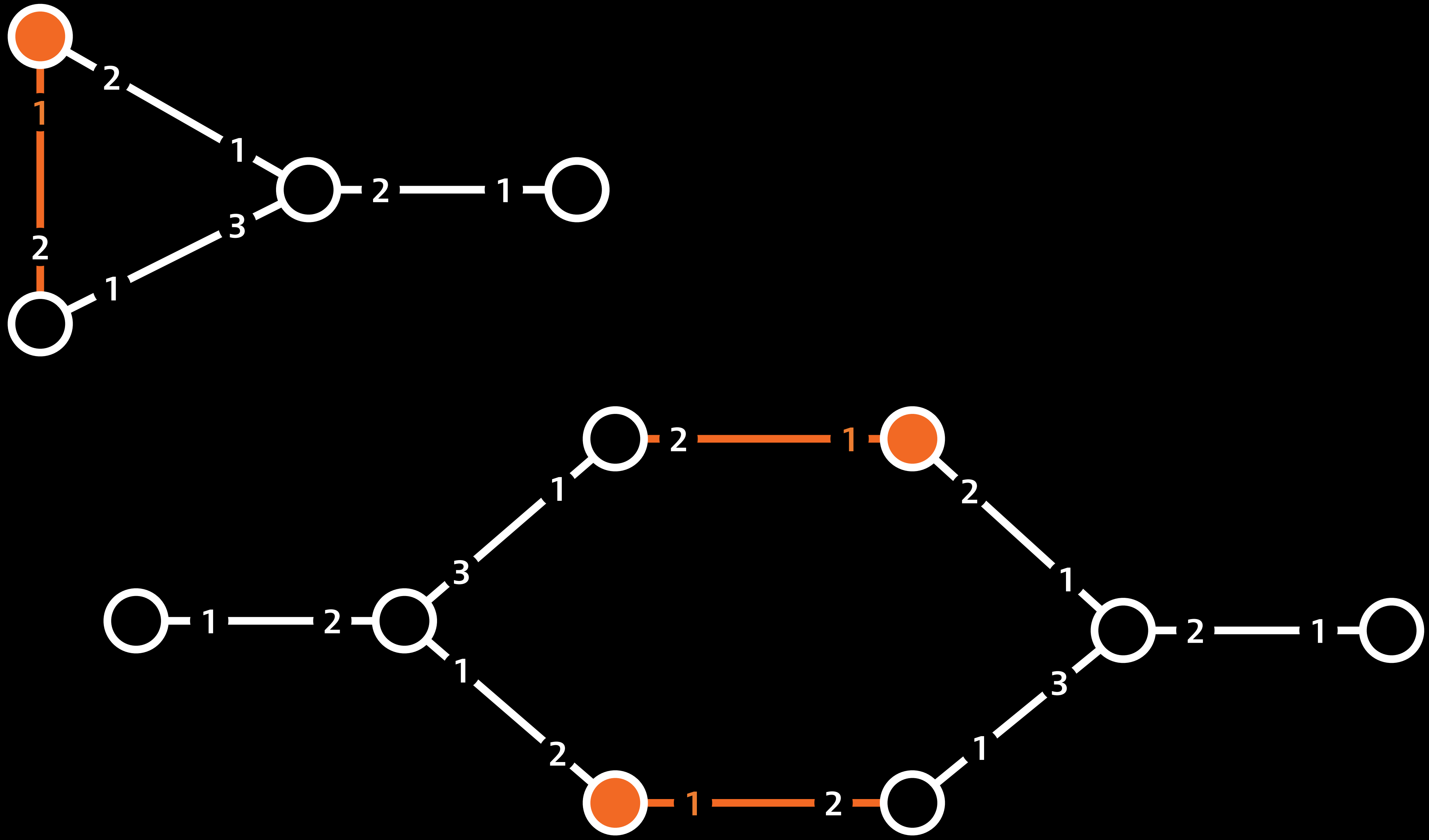


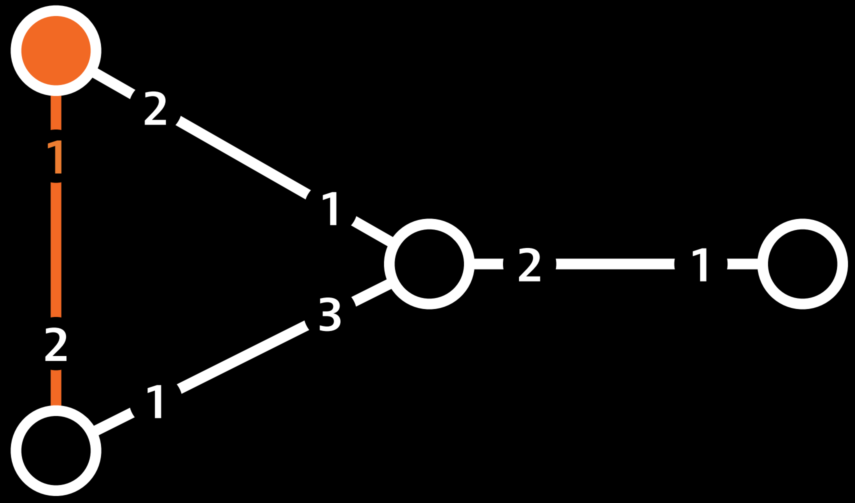


You enter a room with two doors, labeled 1 and 2. You just came in through doorway 1.

> open door 2

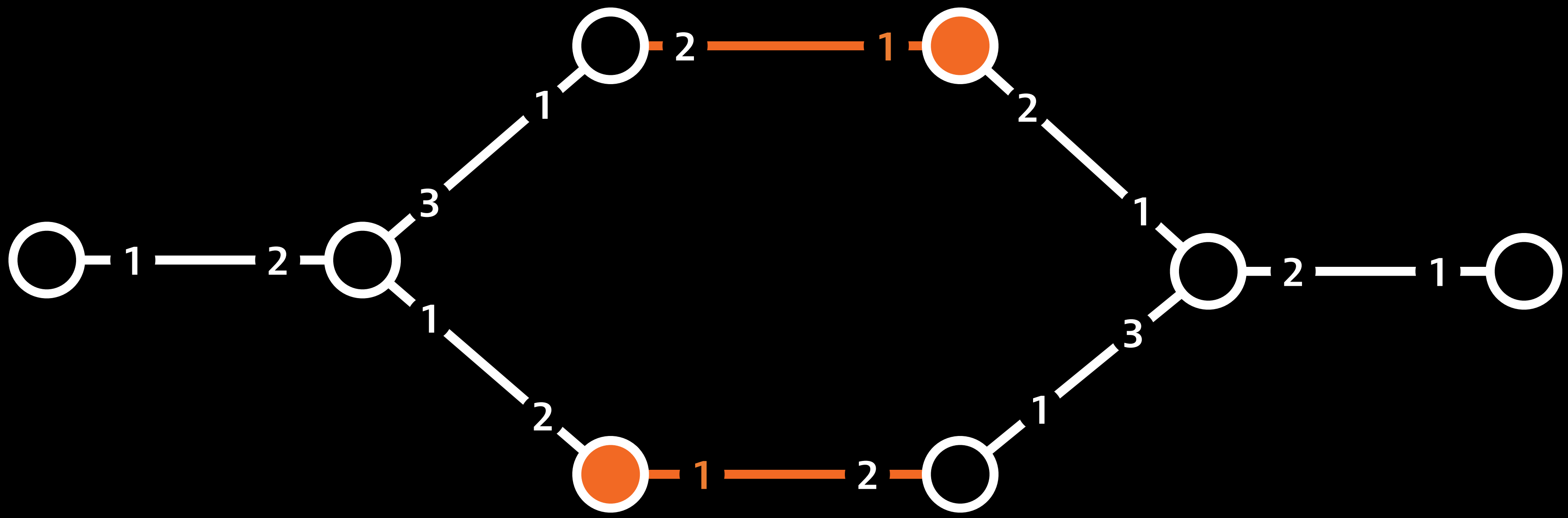


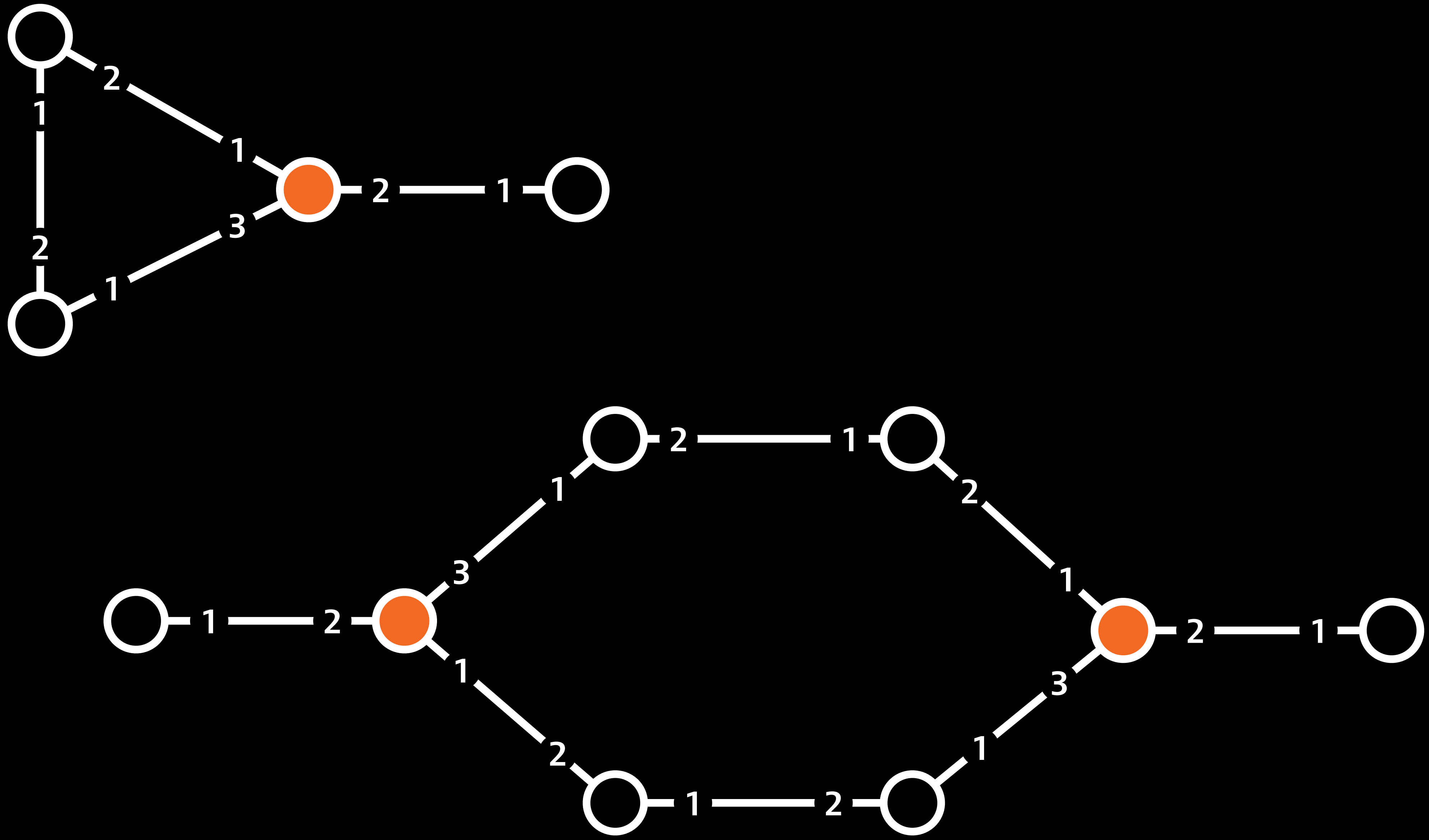

2
1
1
2 1
3
2
1
2 1
1 2
3
1 2
1
1 2
2 1
3
1
2
1 2



You enter a room with two doors, labeled 1 and 2. You just came in through doorway 1.

> _





High-level plan

- **Goal:**
  - show that problem *X* cannot be solved in the port-numbering model

High-level plan

- **Goal:**
  - show that problem *X* cannot be solved in the port-numbering model
- **General approach:**
  - construct port-numbered networks so that some nodes *u*, *v*, … will always produce the ***same output***
  - show that if *u*, *v*, … have the same output, then it is ***not a feasible solution*** for *X*

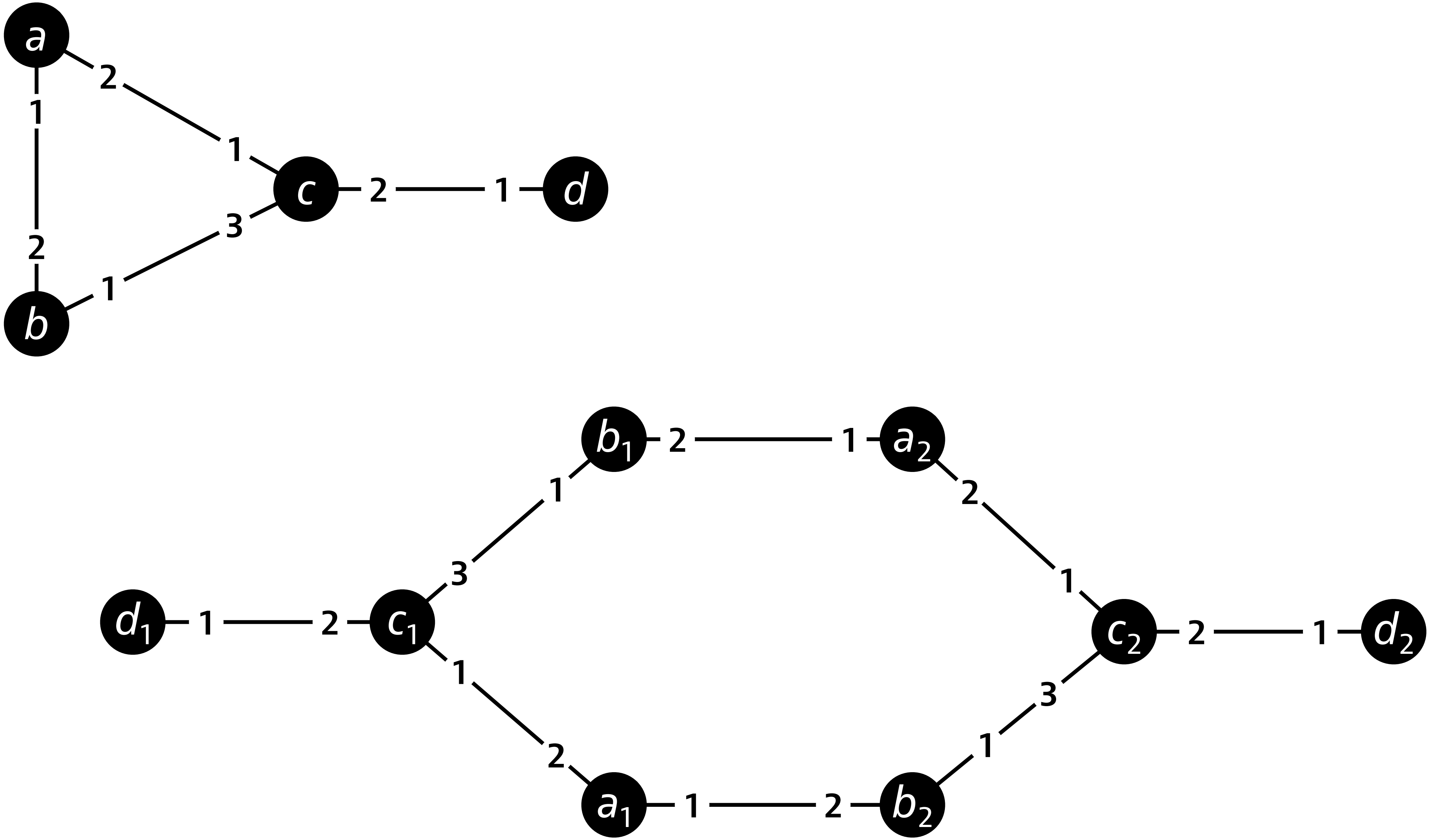
High-level plan

- **Goal:**
  - show that problem X cannot be solved in the port-numbering model
- **General approach:**
  - construct port-numbered networks so that some nodes u, v, … will always produce the ***same output***
  - show that if u, v, … have the same output, then it is ***not a feasible solution*** for X

Covering maps used here

Covering map

- Two port-numbered networks:
  - $N = (V, P, p)$
  - $N' = (V', P', p')$
- Surjection $f\colon V \to V'$ that preserves:
  - inputs
  - degrees
  - connections
  - port numbers


a
b
c
d
b_1
a_2
d_1
c_1
c_2
d_2
a_1
b_2

Covering map

- "Fools" any deterministic algorithm
- If f is a covering map from N to N', then:
  - v and $f(v)$ have the same state ***before*** round 1
  - v and $f(v)$ send the same messages in round 1
  - v and $f(v)$ receive the same messages in round 1
  - v and $f(v)$ have the same state ***after*** round 1

Covering map

- "Fools" any deterministic algorithm
- If f is a covering map from N to N', then:
  - v and $f(v)$ have the same state ***before*** round T
  - v and $f(v)$ send the same messages in round T
  - v and $f(v)$ receive the same messages in round T
  - v and $f(v)$ have the same state ***after*** round T

Common steps

- Starting point: graph problem X
- Which graph G would be a “hard instance”?
- How to choose a port numbering N of G ?
- How to choose the other network N' ?
- How to construct mapping from N to N' ?

Example: 2-node path

Example: 4-node path

Aalto University

Distributed Algorithms 2024

8 Local neighborhoods

High-level plan

Algorithm A runs in $\boldsymbol{T}$ **rounds** and solves problem X

➞ A is a **mapping from radius-$\boldsymbol{T}$ neighborhoods** to local outputs

Such a mapping cannot solve X correctly

➞ Problem X is not solvable in T rounds

Example: coloring

- **Problem:** find a vertex coloring with the smallest possible number of colors
- **Proof:** *three different approaches!*

Example: coloring

- **Idea 1:** consider a path, ***fix solutions in two neighborhoods***, construct another path

Example: coloring

- **Idea 2:** consider an odd cycle, *look at a node that outputs “3”*, construct a path

Example: coloring

- **Idea 3:** if we can 2-color paths locally, *then we can also 2-color odd cycles*

What about…

- PN model?
- CONGEST model?
- Randomized algorithms?

Example: is it a forest?

- **Input is a forest:** ***all*** nodes output ***"yes"***, **otherwise:** ***at least one*** node outputs ***"no"***
- **Questions:**
  - is this solvable in PN, and how fast?
  - is this solvable in LOCAL, and how fast?
  - does it help if we know n?