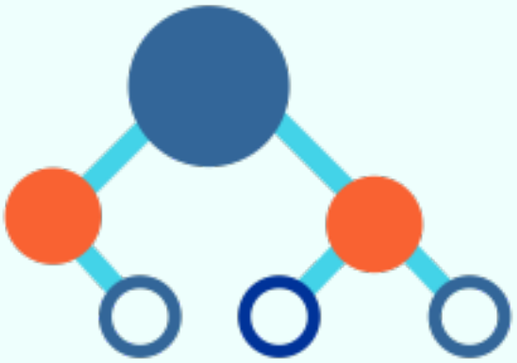
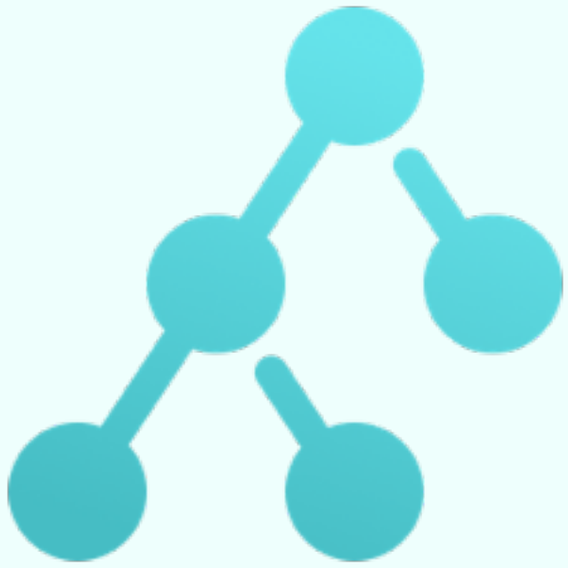
Data Structures

Exercise Session



Marco Vassena



Problem 3

Find if a list of words contains a reversed word

Problem 3

Find if a list of words contains a reversed word

cat

live

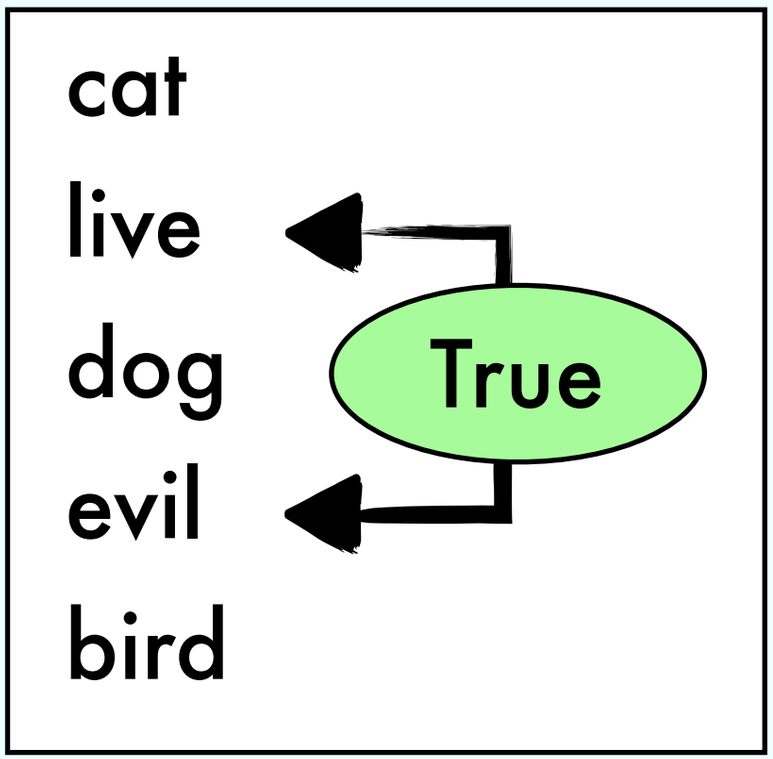
dog

evil

bird

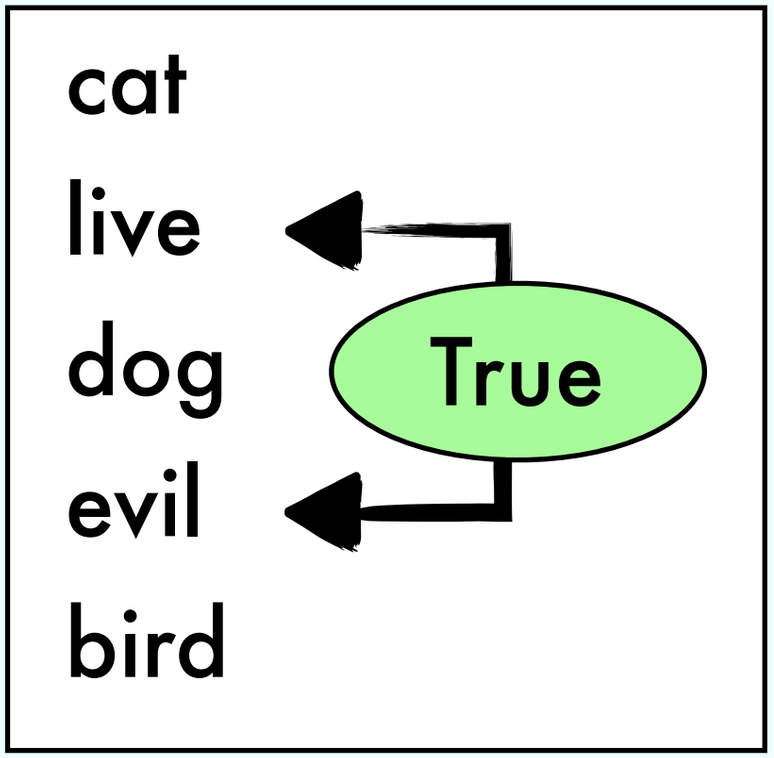
Problem 3

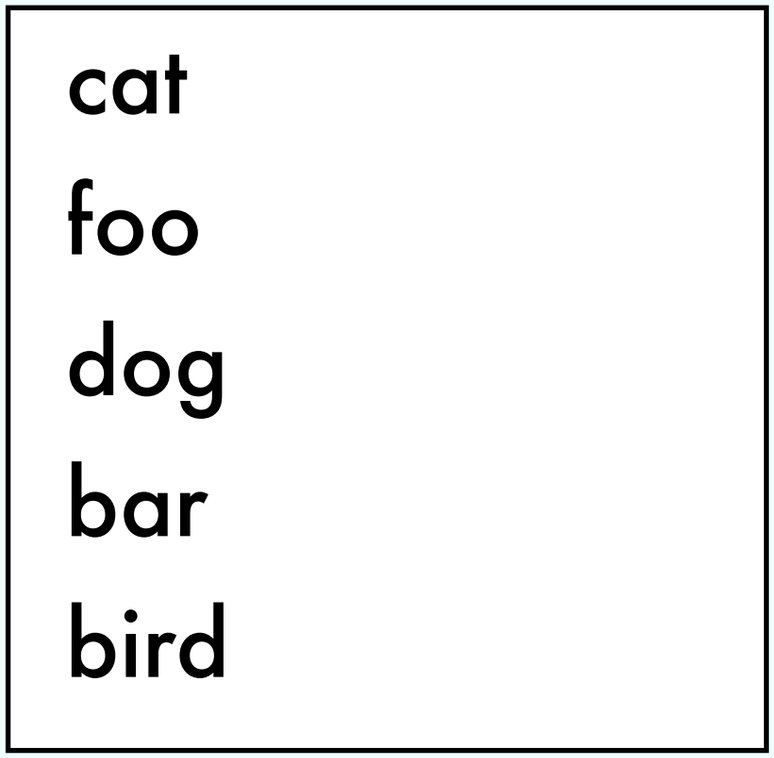
Find if a list of words contains a reversed word



Problem 3

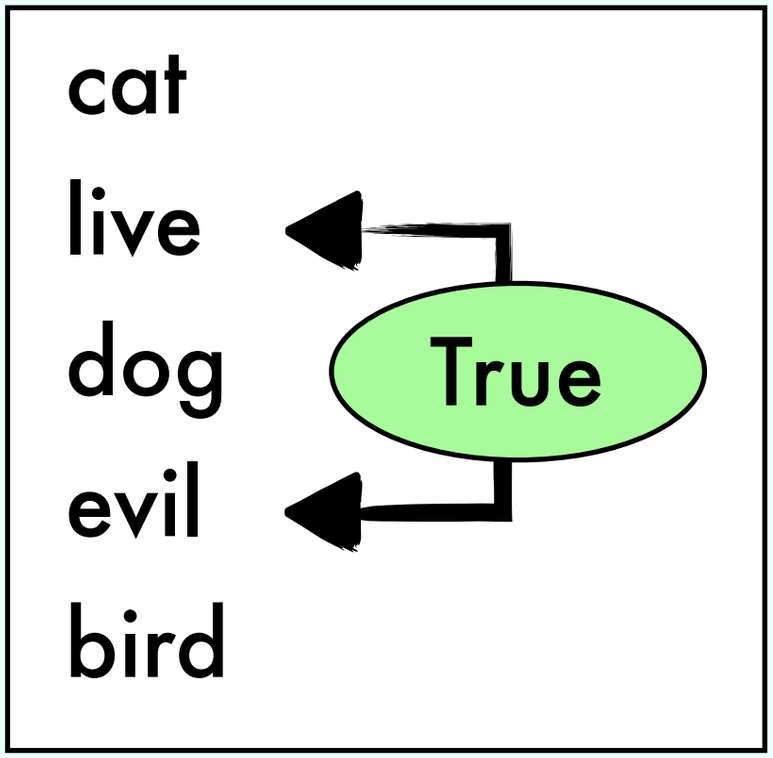
Find if a list of words contains a reversed word

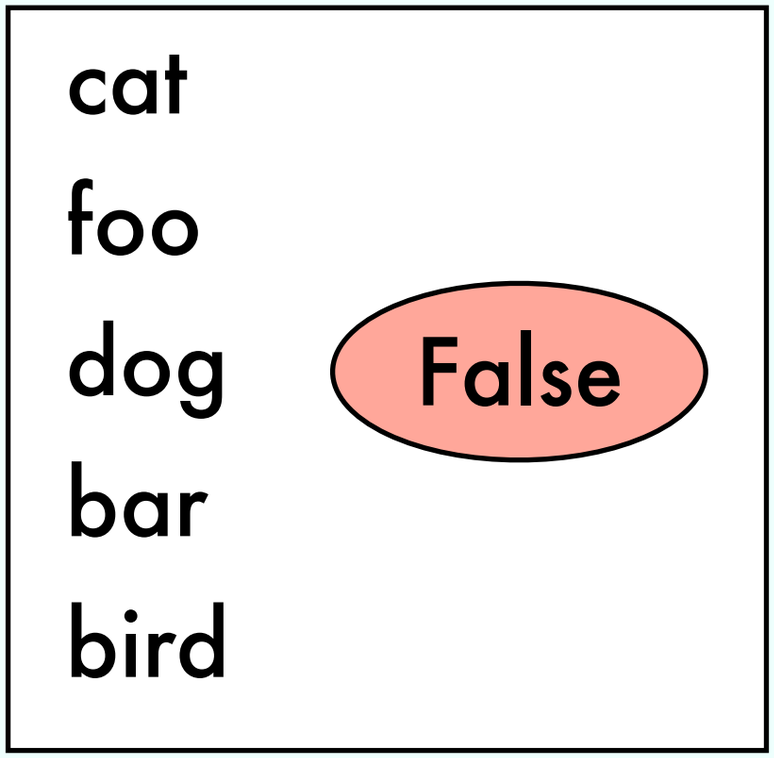




Problem 3

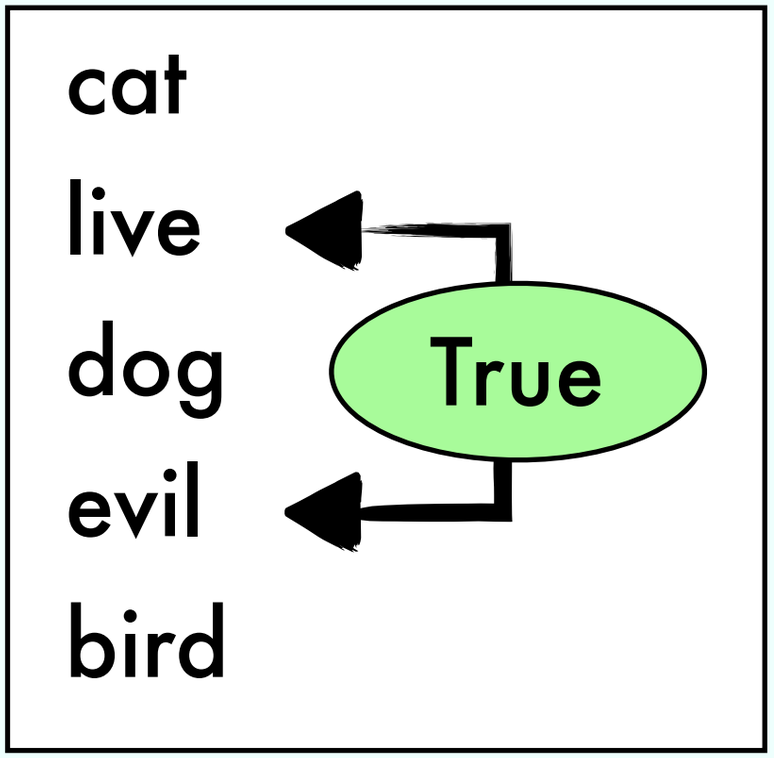
Find if a list of words contains a reversed word

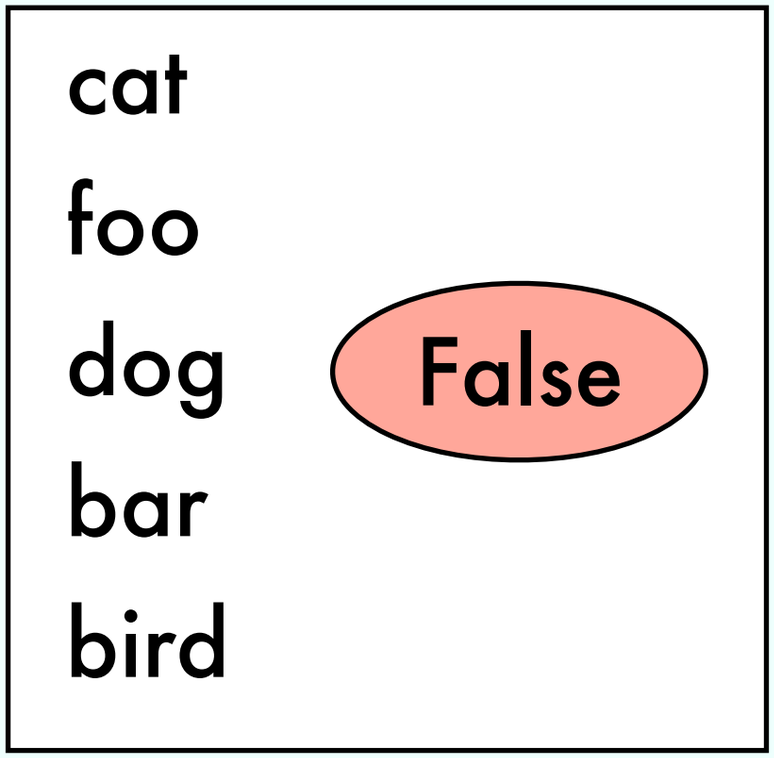




Problem 3

Find if a list of words contains a reversed word



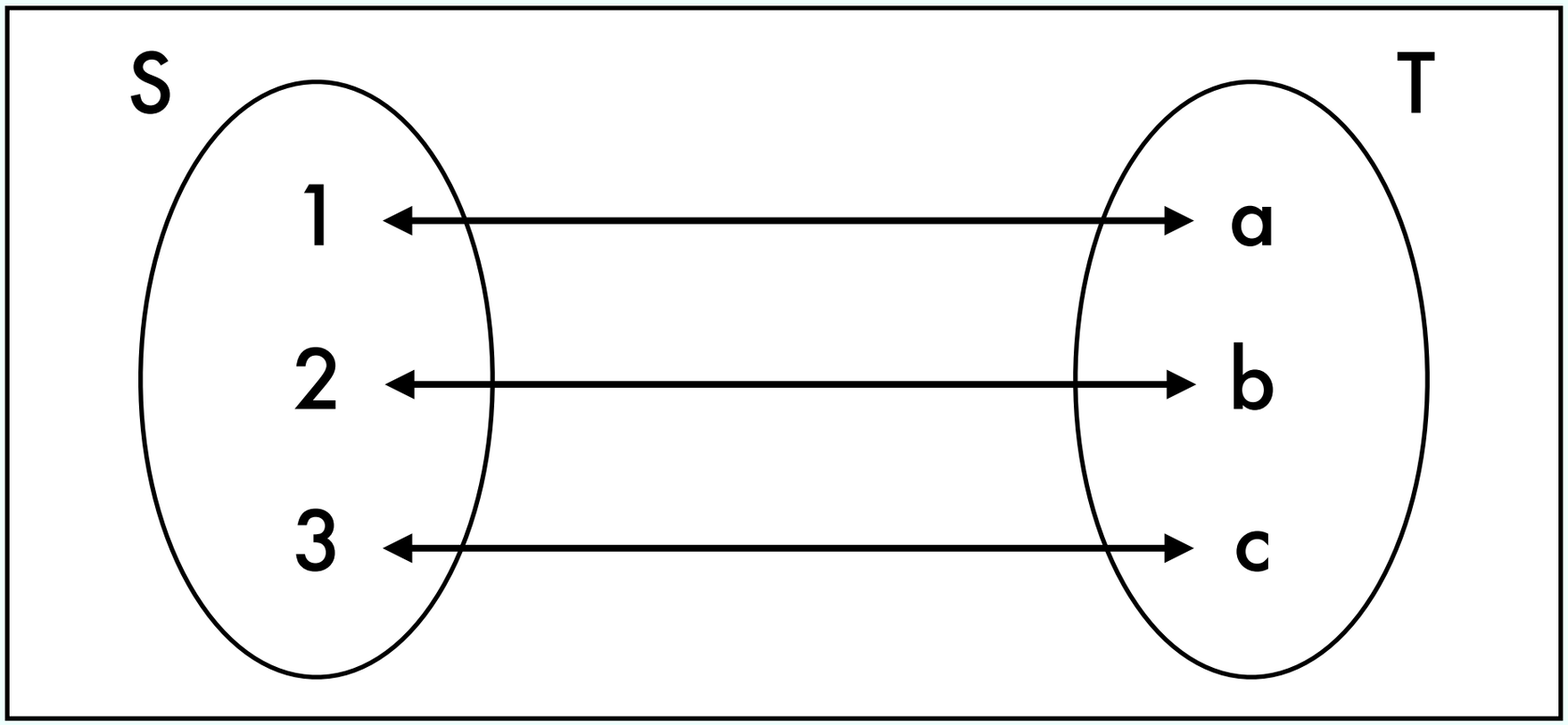


Use hash-based data structures

Analyze the time complexity

```
// ∑_{w∈ws} |w| = N
S = new HashSet();
for (String w : ws)
  if (reverse(w) ∈ S)
    then return true;
    else S.insert(w);
return false;
```

Bijection



Invariant

$\forall\ s \in S, t \in T$

$s \mapsto t$ if and only if $t \leftarrowtail s$

Exercise 2 from 12/8

Design a data structure for bijection

Operation	Time Complexity
`insert(s, t)`	O(log N)
`source(t)`	O(log N)
`target(s)`	O(log N)

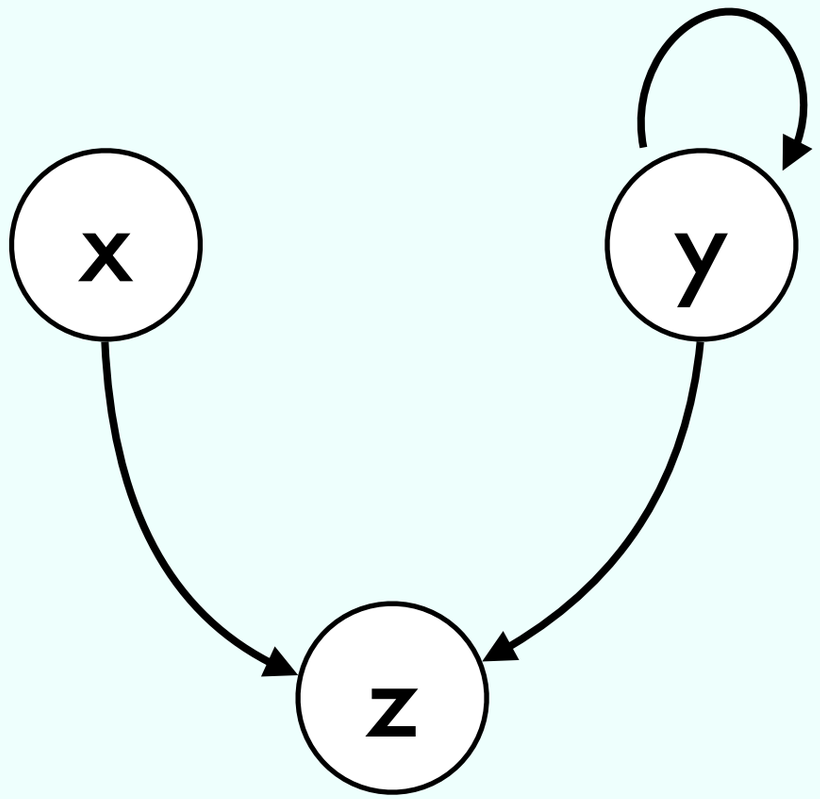
Equivalence Relation

R is an equivalence relation, iff

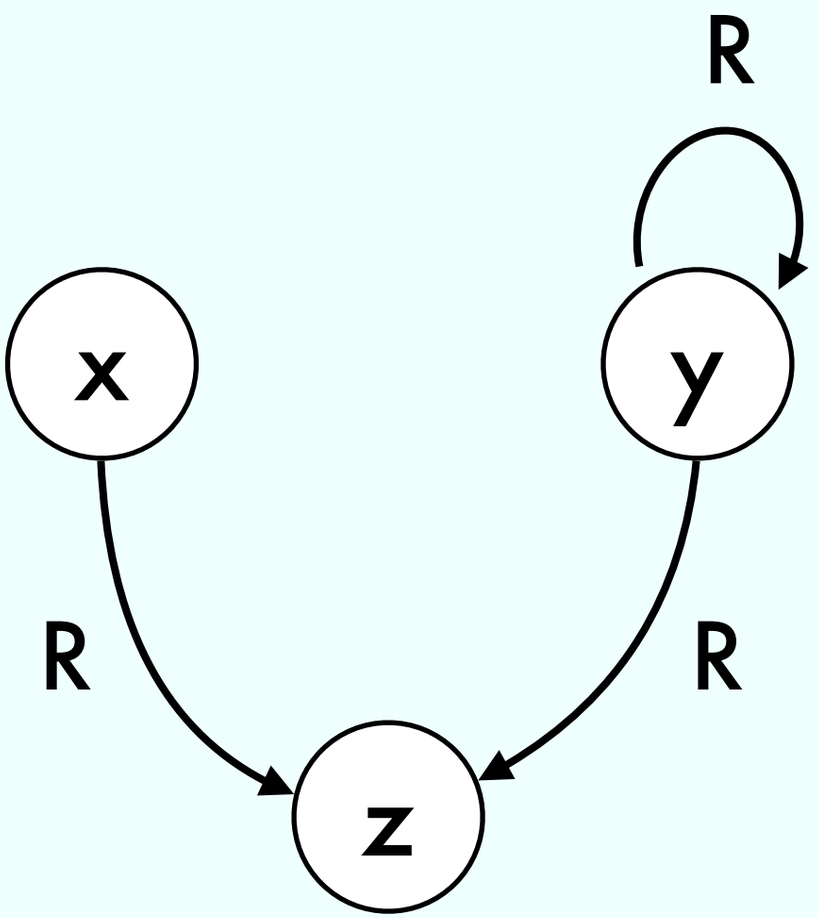
$\forall\ x\ y\ z$

reflexive	$x\ R\ x$
symmetric	$x\ R\ y \Rightarrow y\ R\ x$
transitive	$x\ R\ y, y\ R\ z \Rightarrow x\ R\ z$

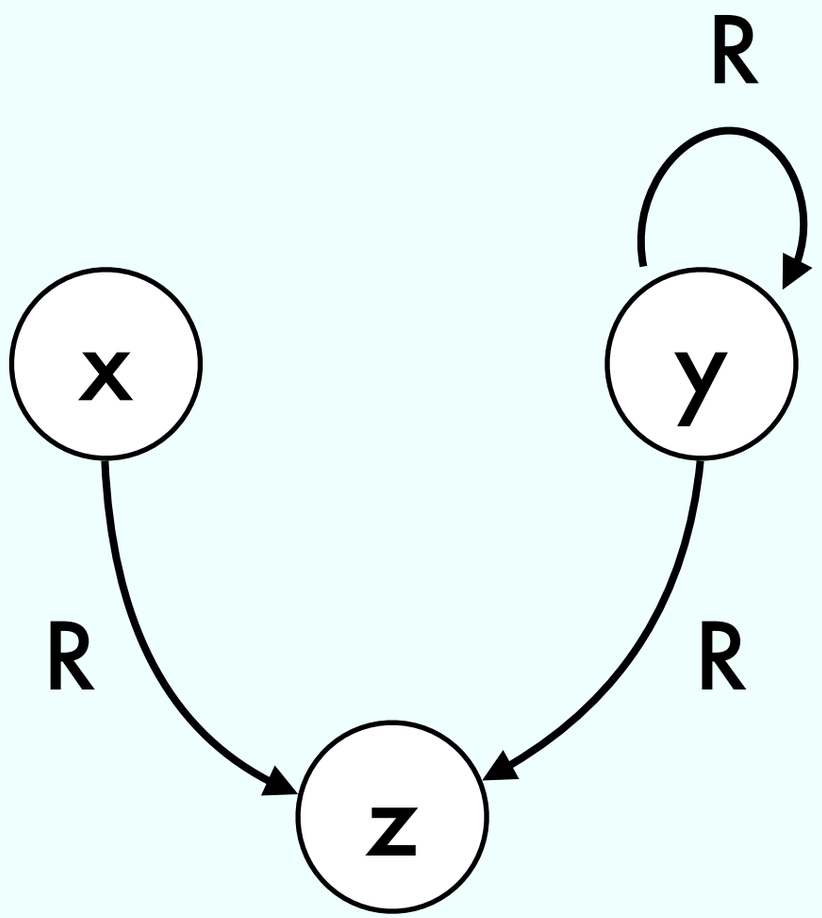
A graph G = (V, E) represents a relation R



A graph G = (V, E) represents a relation R



A graph G = (V, E) represents a relation R



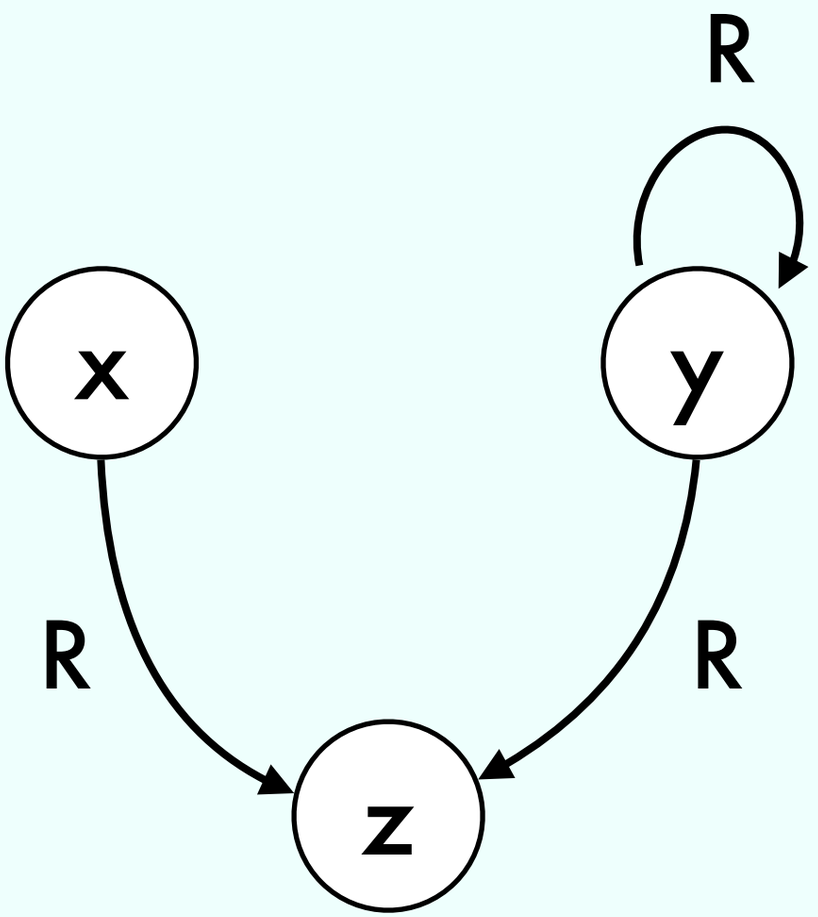
⇔

x R z

y R z

y R y

A graph G = (V, E) represents a relation R



⇔

x R z

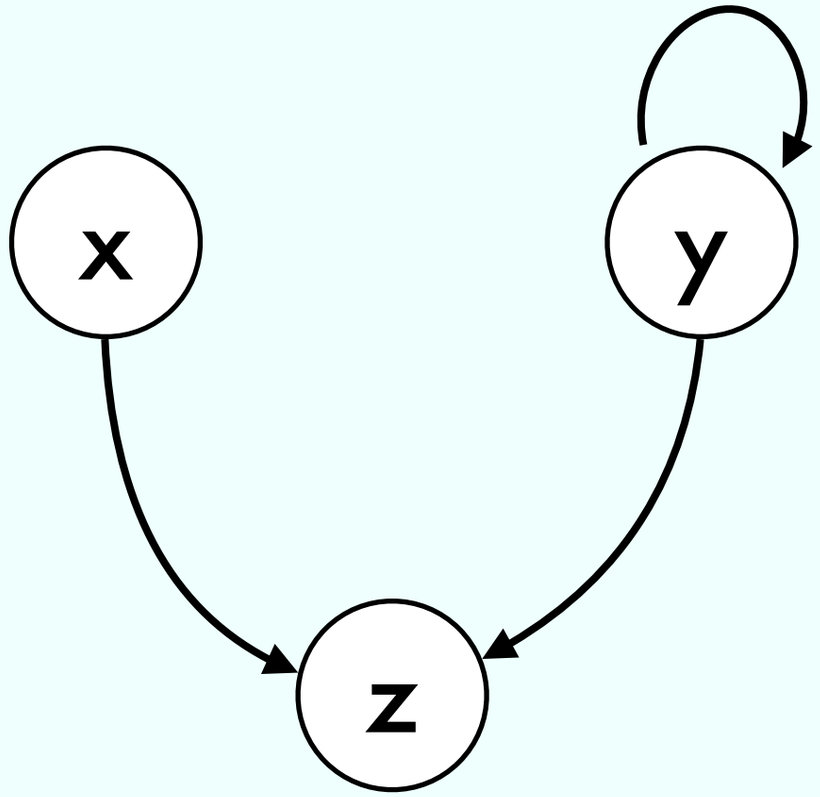
y R z

y R y

$(x,y) \in E \Leftrightarrow x\ R\ y$

Problem 6

Does G represent
an equivalence relation?



$\Leftrightarrow$

$x \; R \; z$

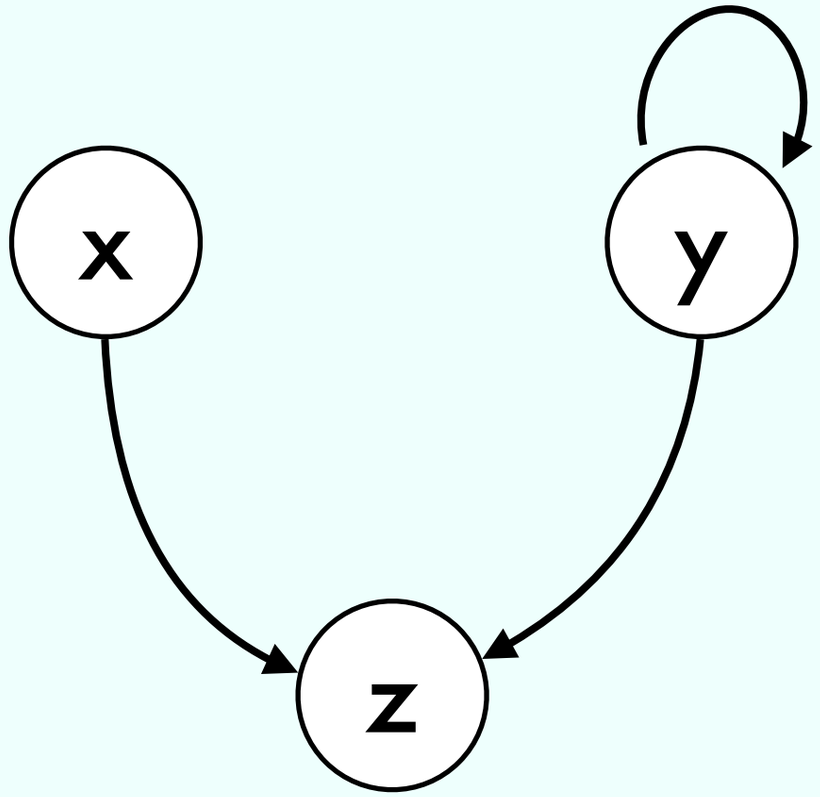
$y \; R \; z$

$y \; R \; y$

Problem 6

Does G represent
an equivalence relation?

False

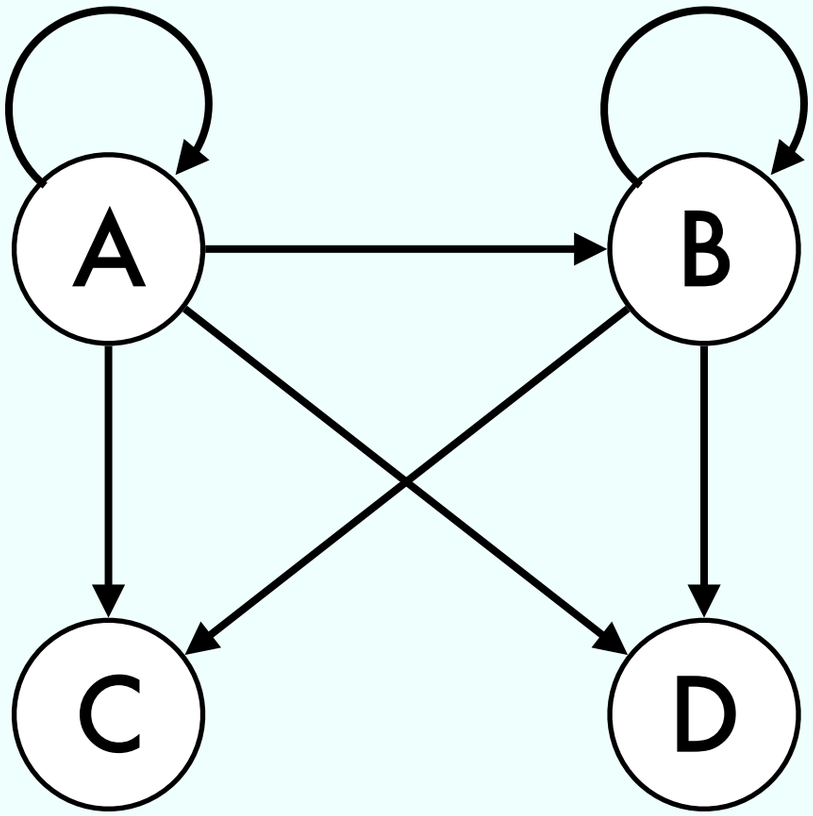


$\Leftrightarrow$

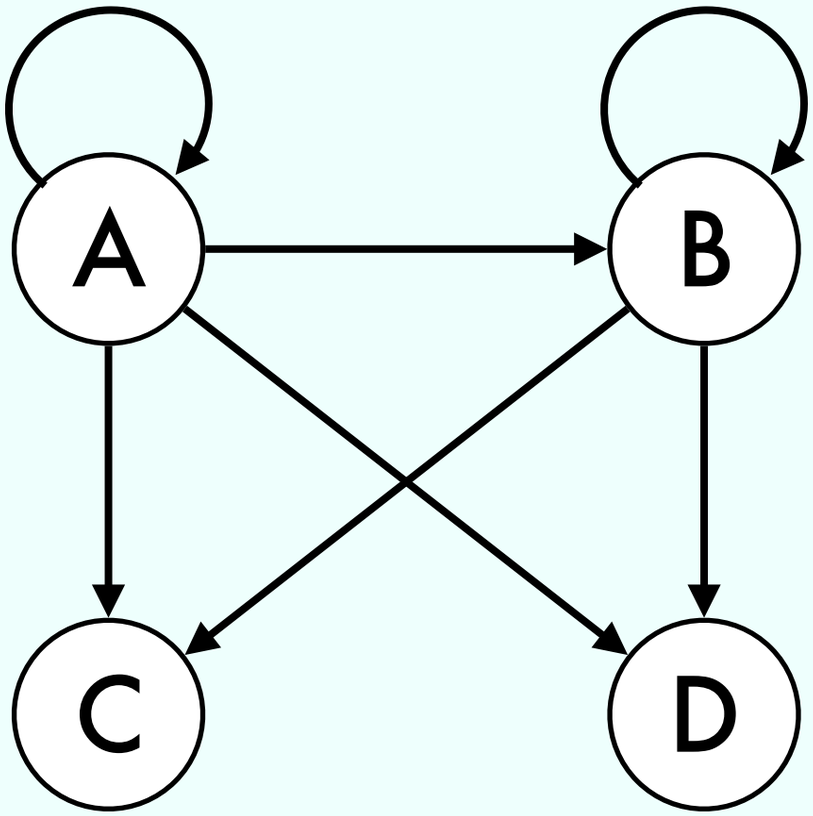
x R z

y R z

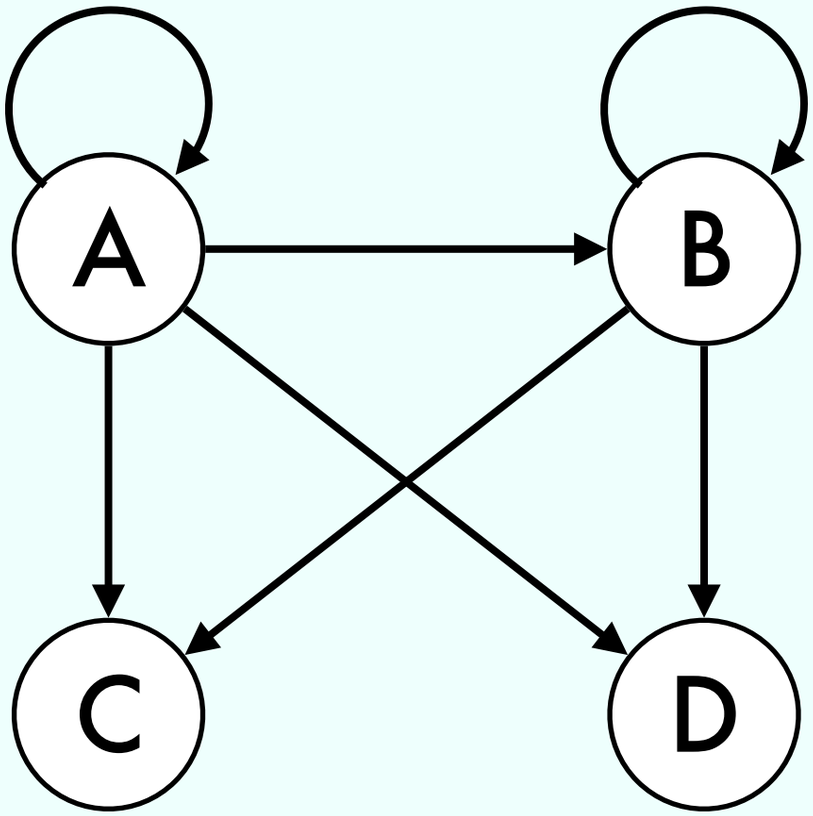
y R y



Path	Length



Path	Length
A - B	
B - C	
A - C	
A - A	
A - D	
B - D	
B - B	



Path	Length
A - B	1
B - C	1
A - C	1
A - A	0
A - D	1
B - D	1
B - B	0

Theorem

G reflexive

G transitive

⇔

∀ u v ∈ V

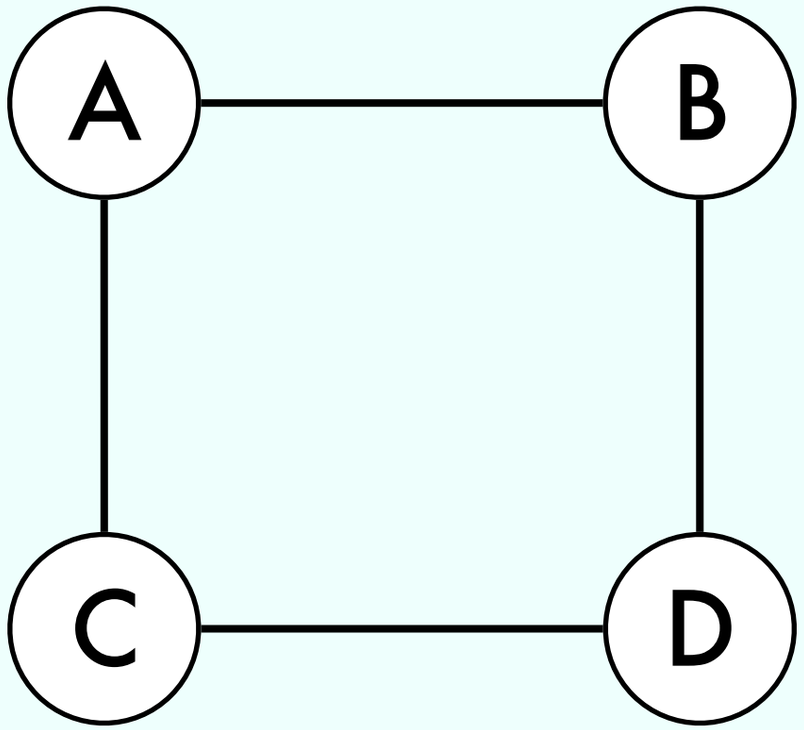
shortestPath(u, v) ≤ 1

Problem 5

Hamiltonian Path
Path that visits all nodes exactly once

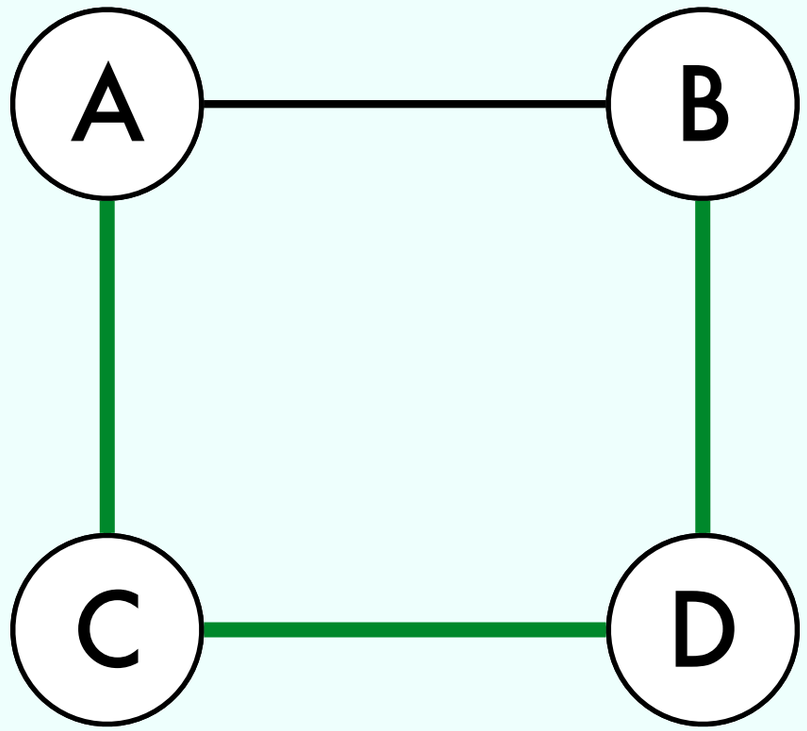
Problem 5

Hamiltonian Path
Path that visits all nodes exactly once



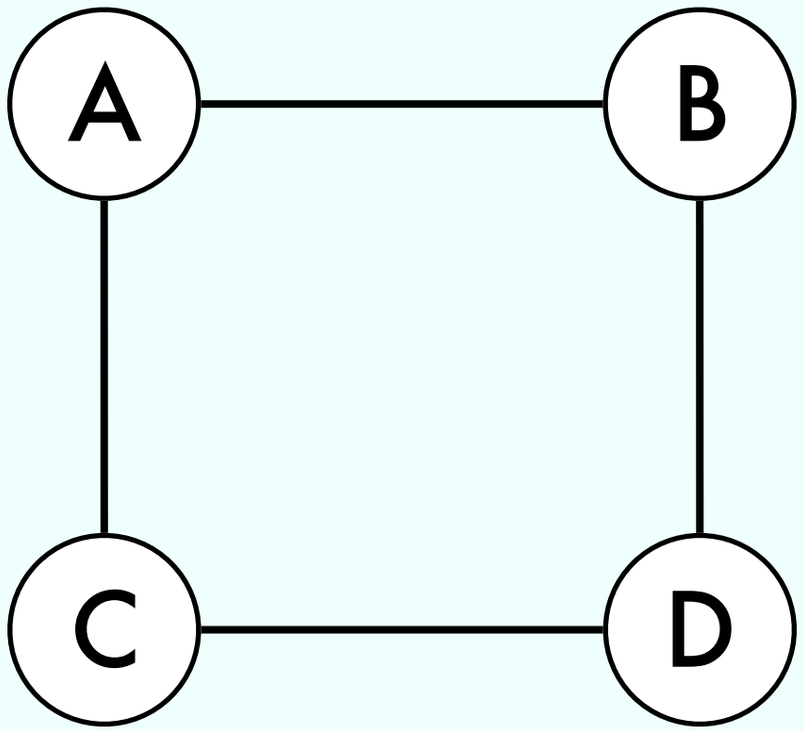
Problem 5

Hamiltonian Path
Path that visits all nodes exactly once



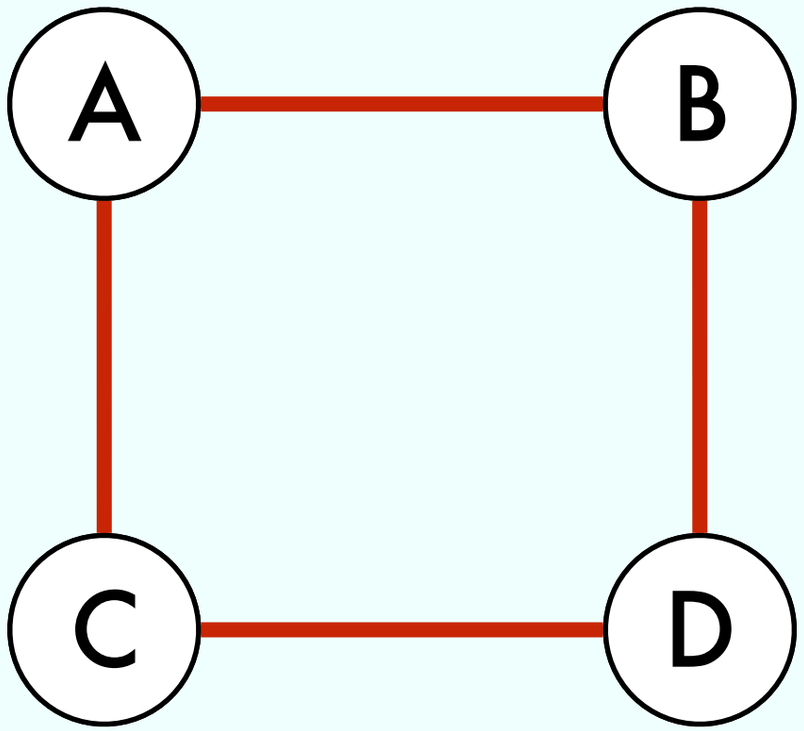
Problem 5

Hamiltonian Cycle
Hamiltonian path that is a cycle



Problem 5

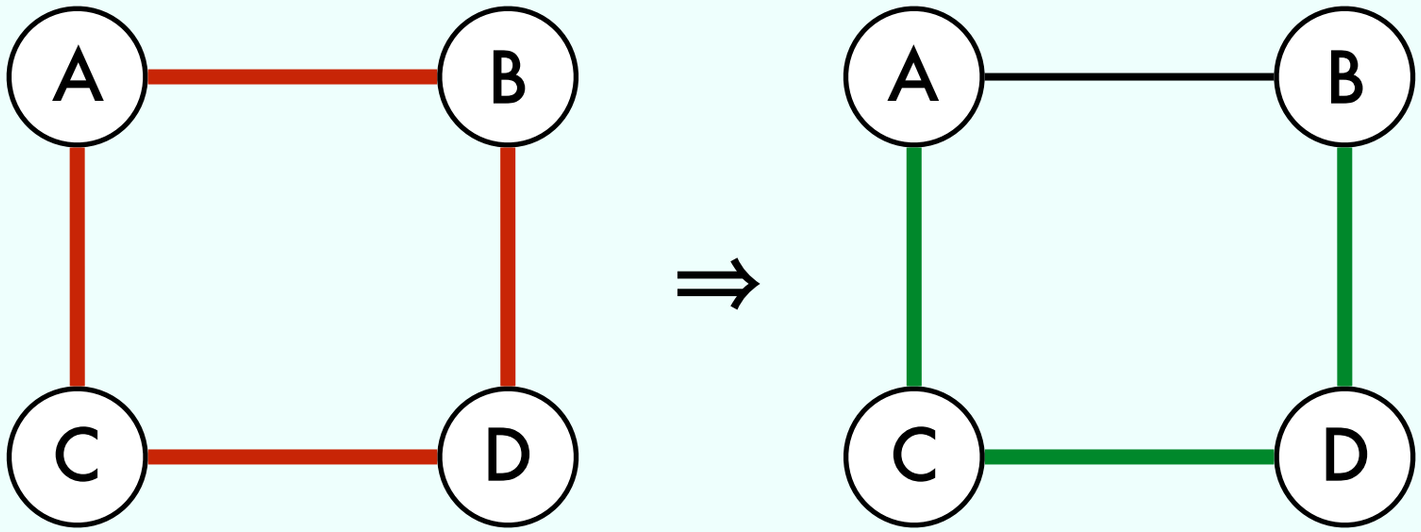
Hamiltonian Cycle
Hamiltonian path that is a cycle

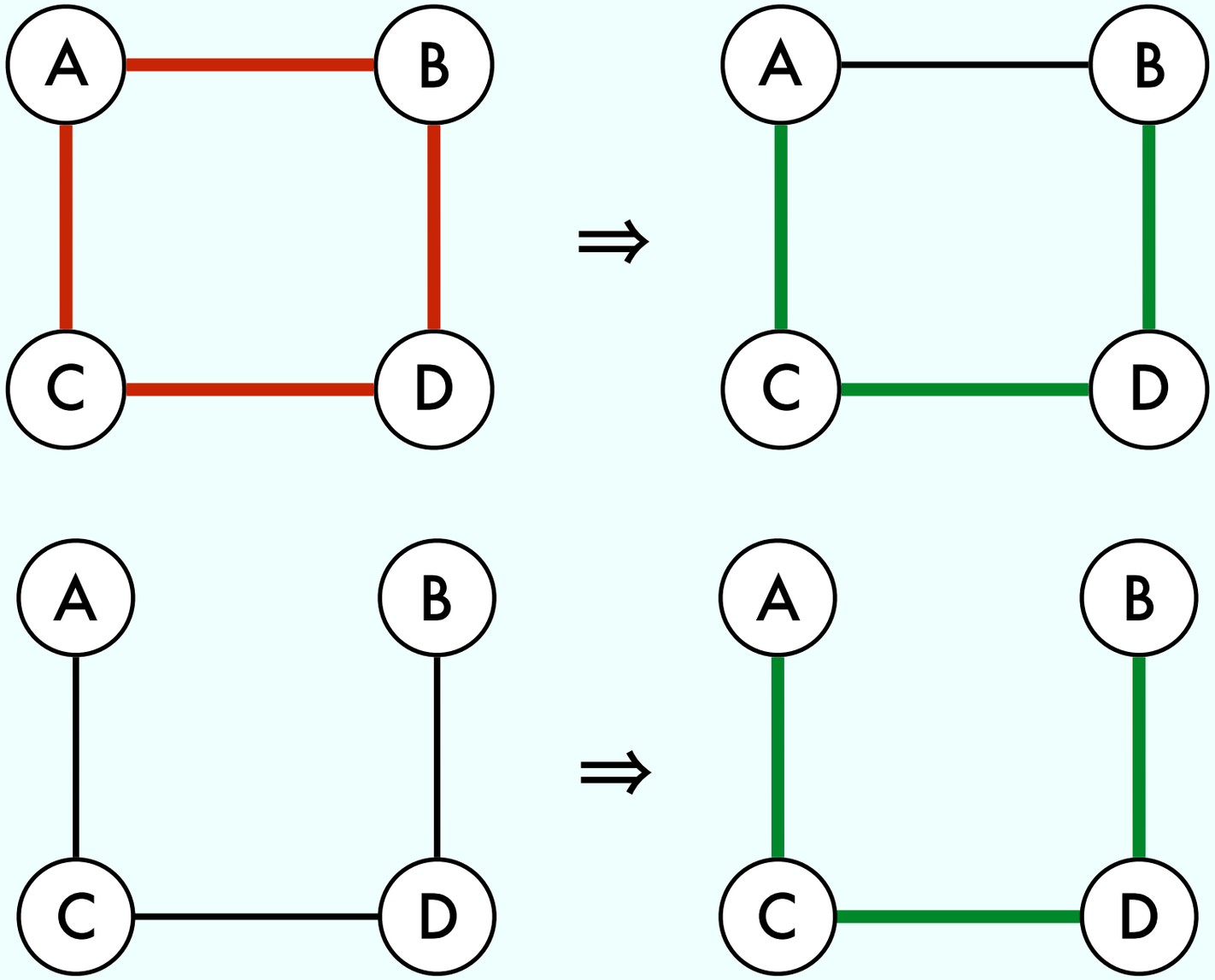


Problem 5

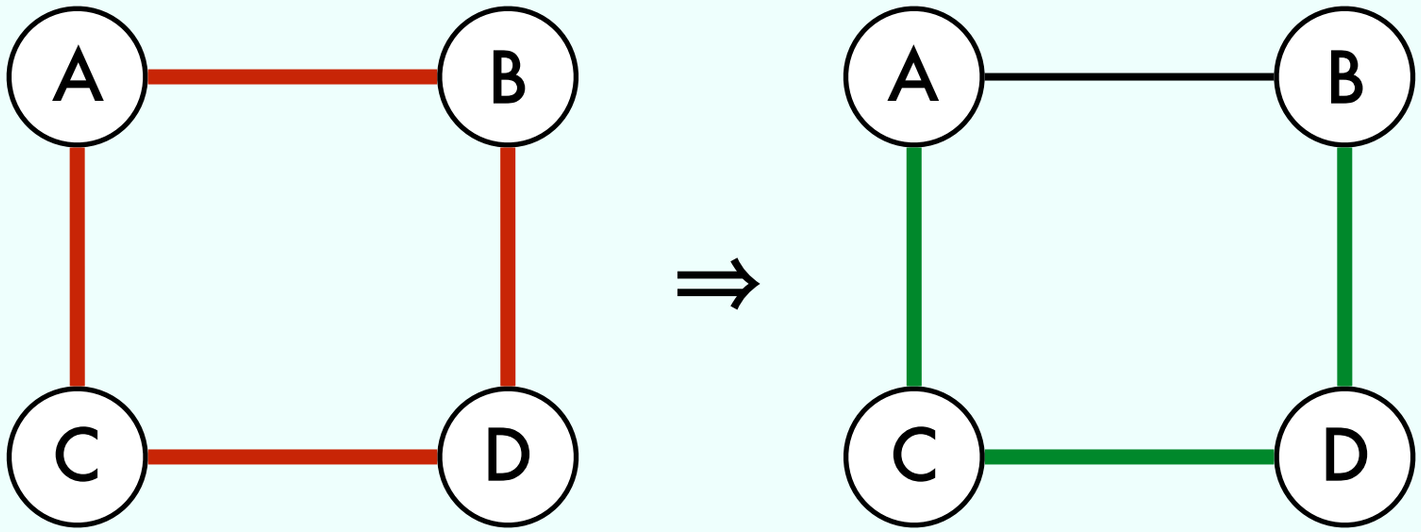
Assume you have hasHCycle(G)

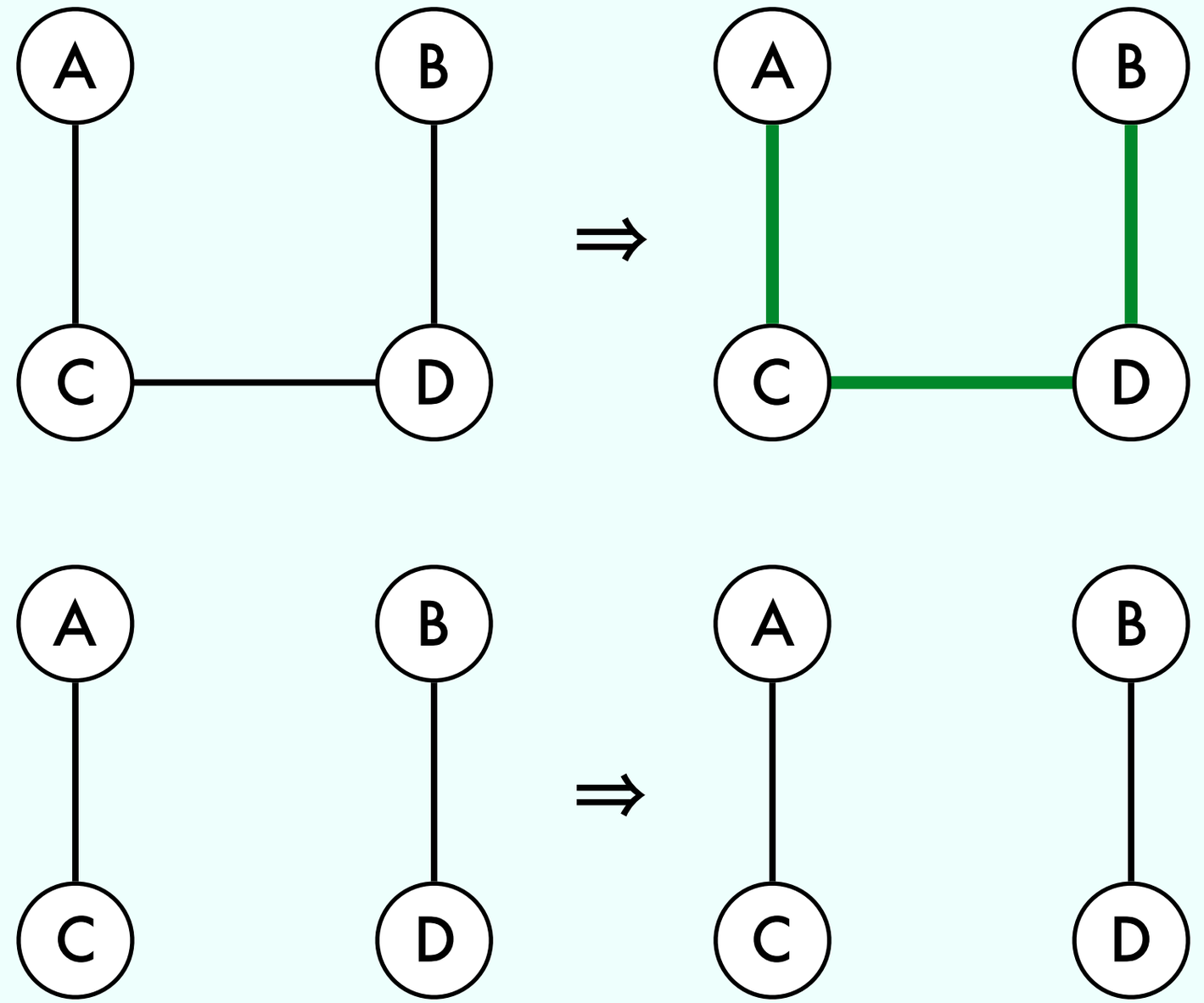
Implement hasHPath(G)


A
B
C
D
⇒
A
B
C
D

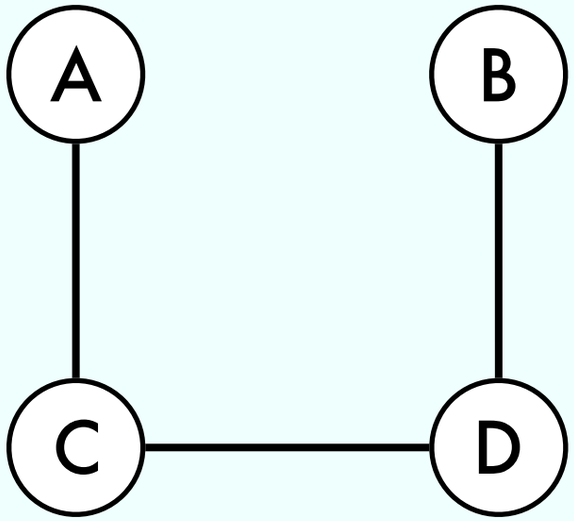

A
B
C
D
⇒
A
B
C
D

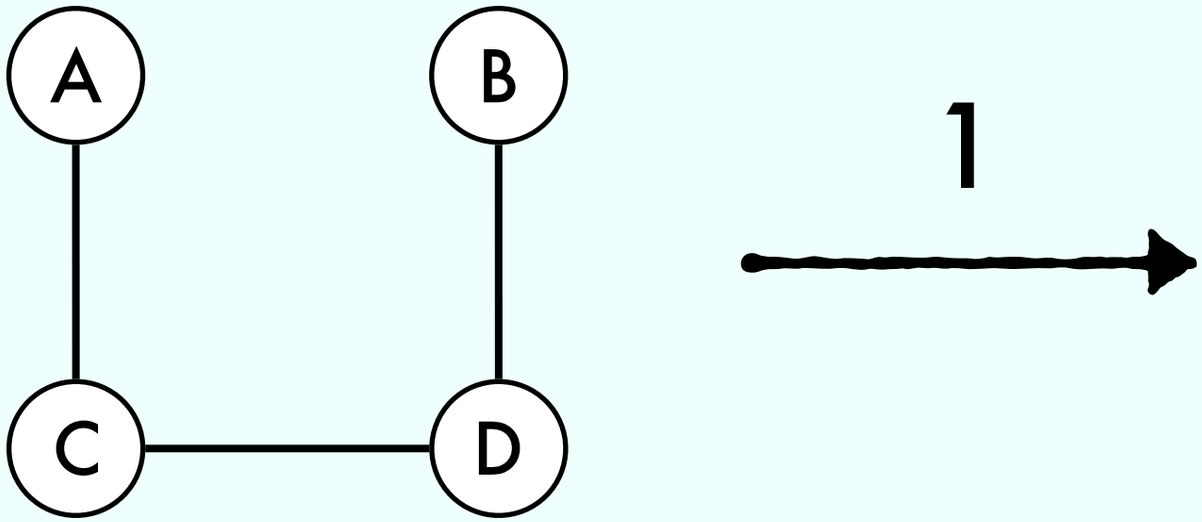
A
B
C
D
⇒
A
B
C
D

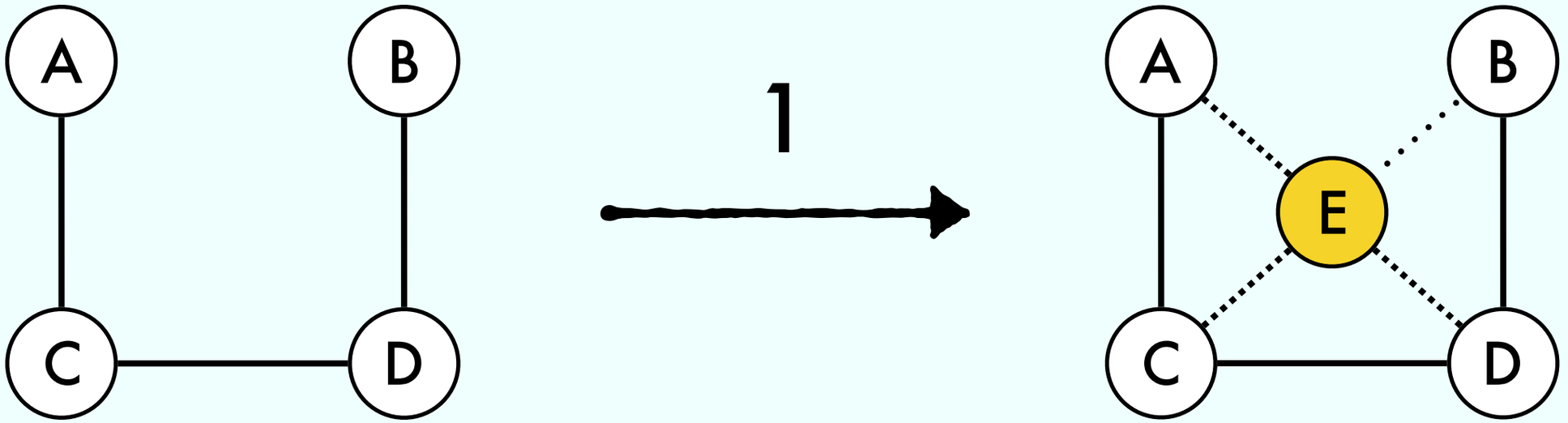

A
B
C
D
⇒
A
B
C
D

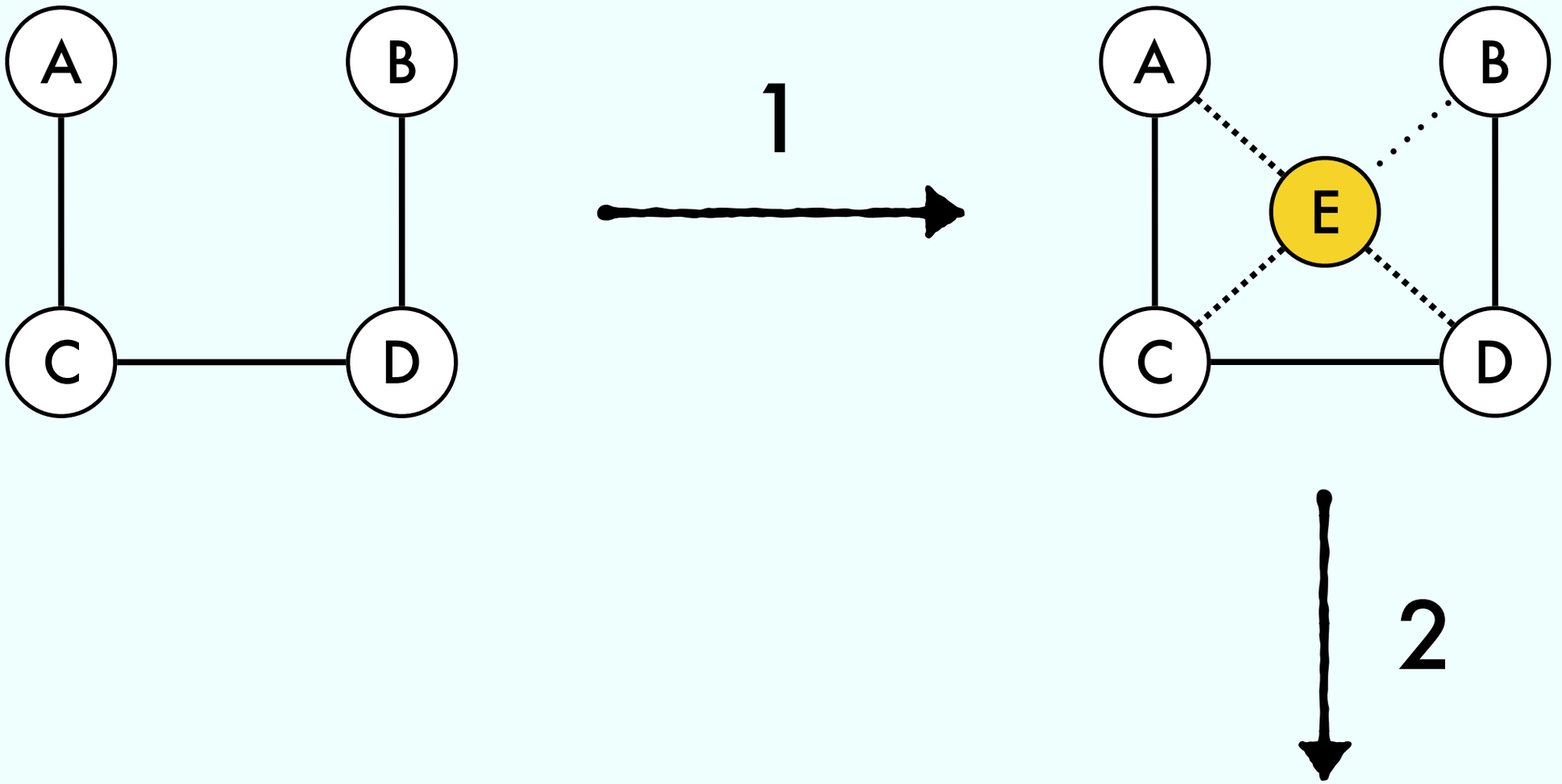

A
B
C
D
⇒
A
B
C
D

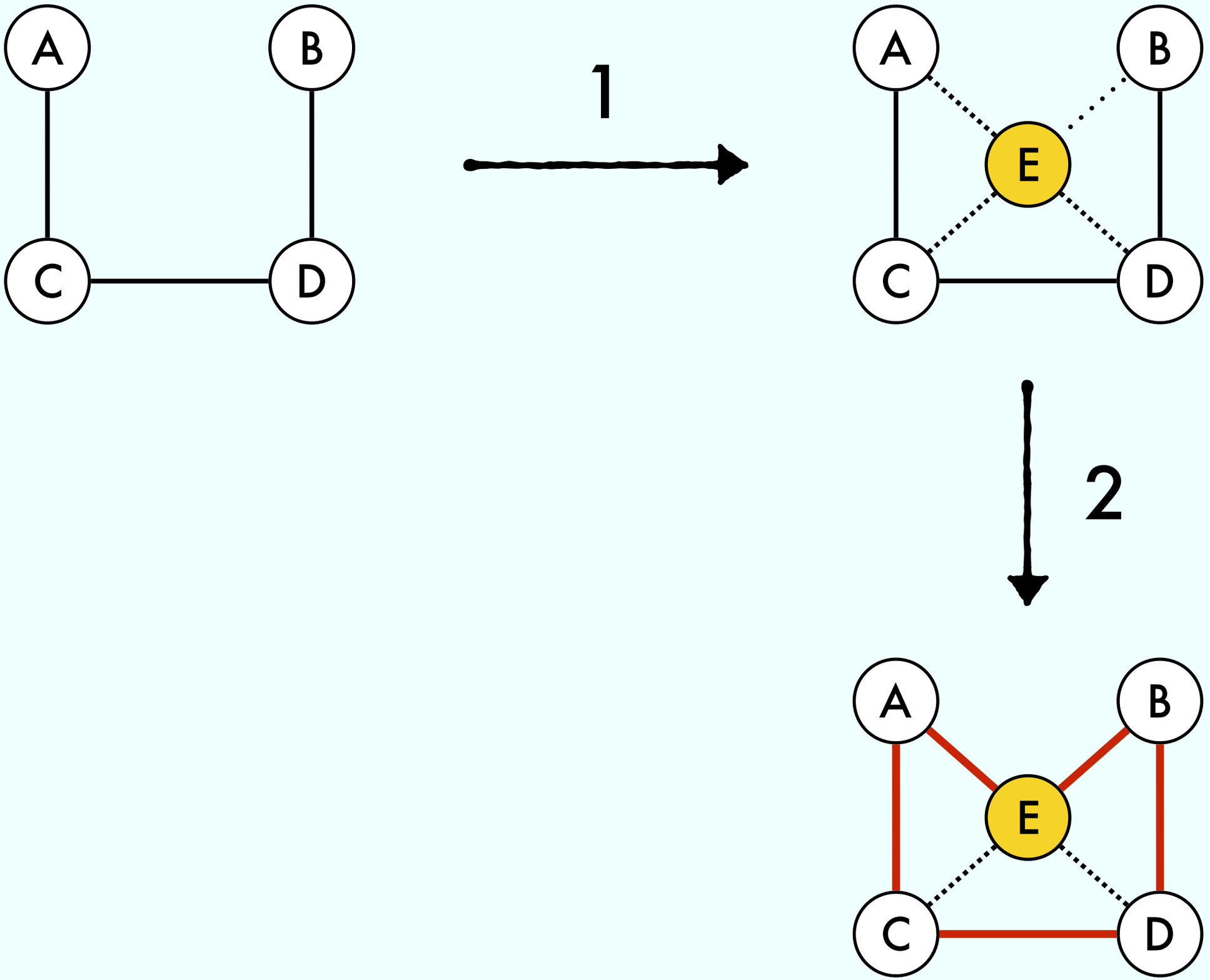
A
B
C
D
⇒
A
B
C
D

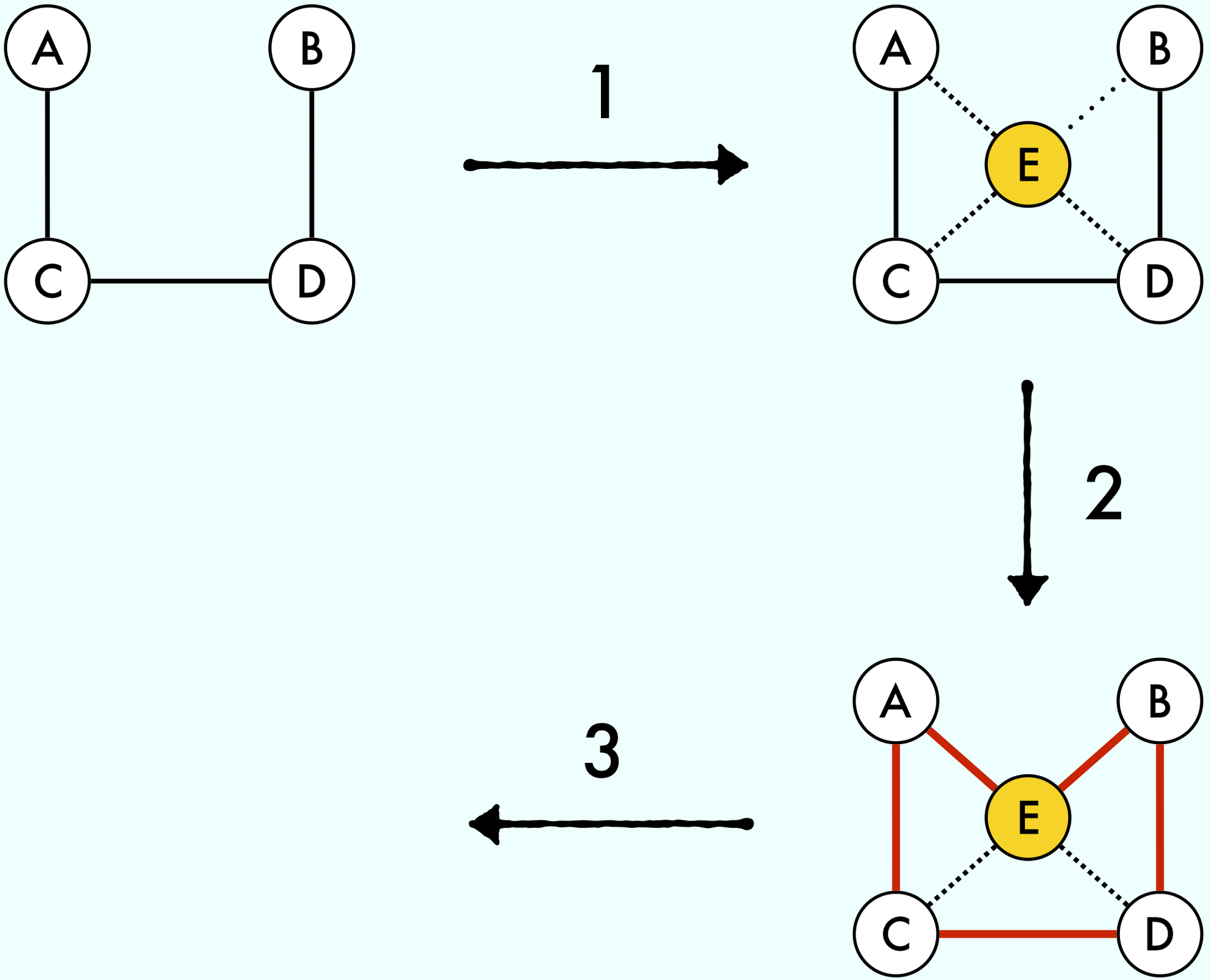

A
B
C
D

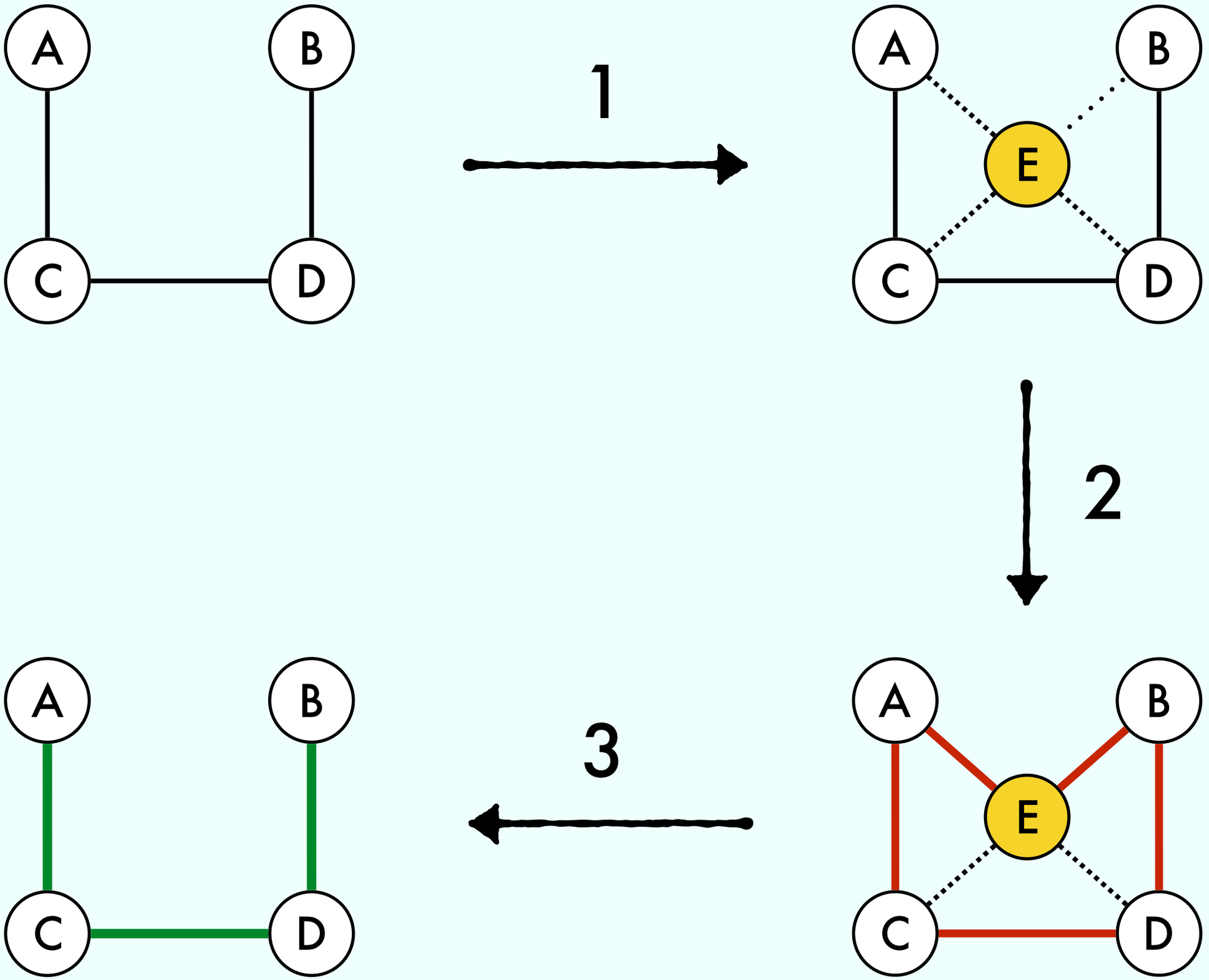

A
B
C
D
1

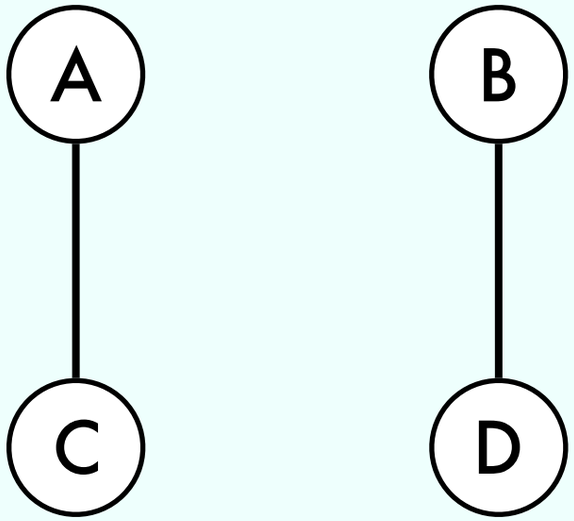

A
B
C
D
1
A
B
E
C
D

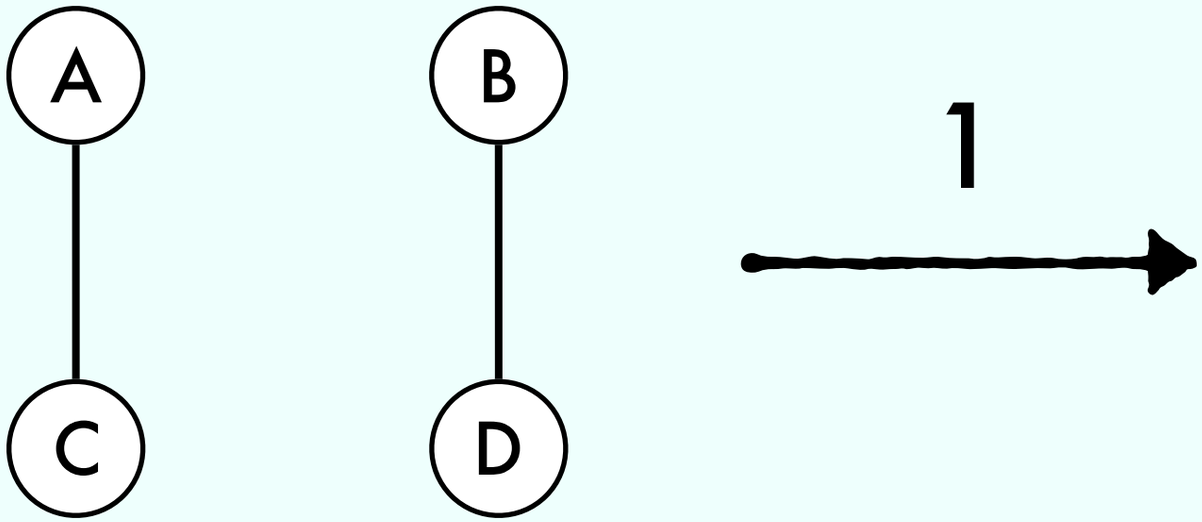

A
B
C
D
1
A
B
E
C
D
2

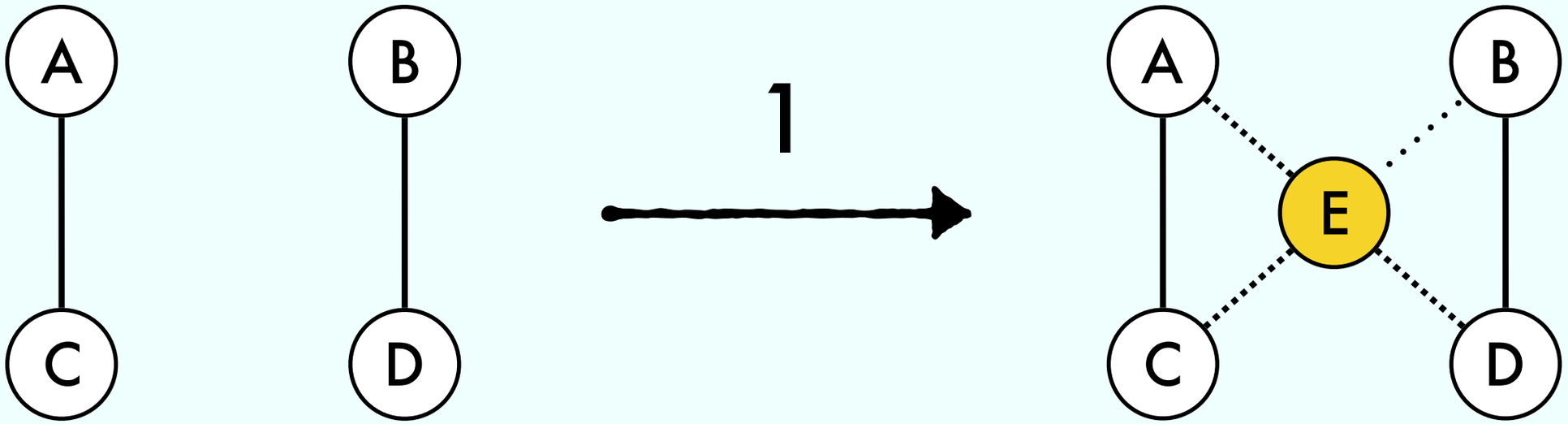

A
B
C
D
1
A
B
E
C
D
2
A
B
E
C
D

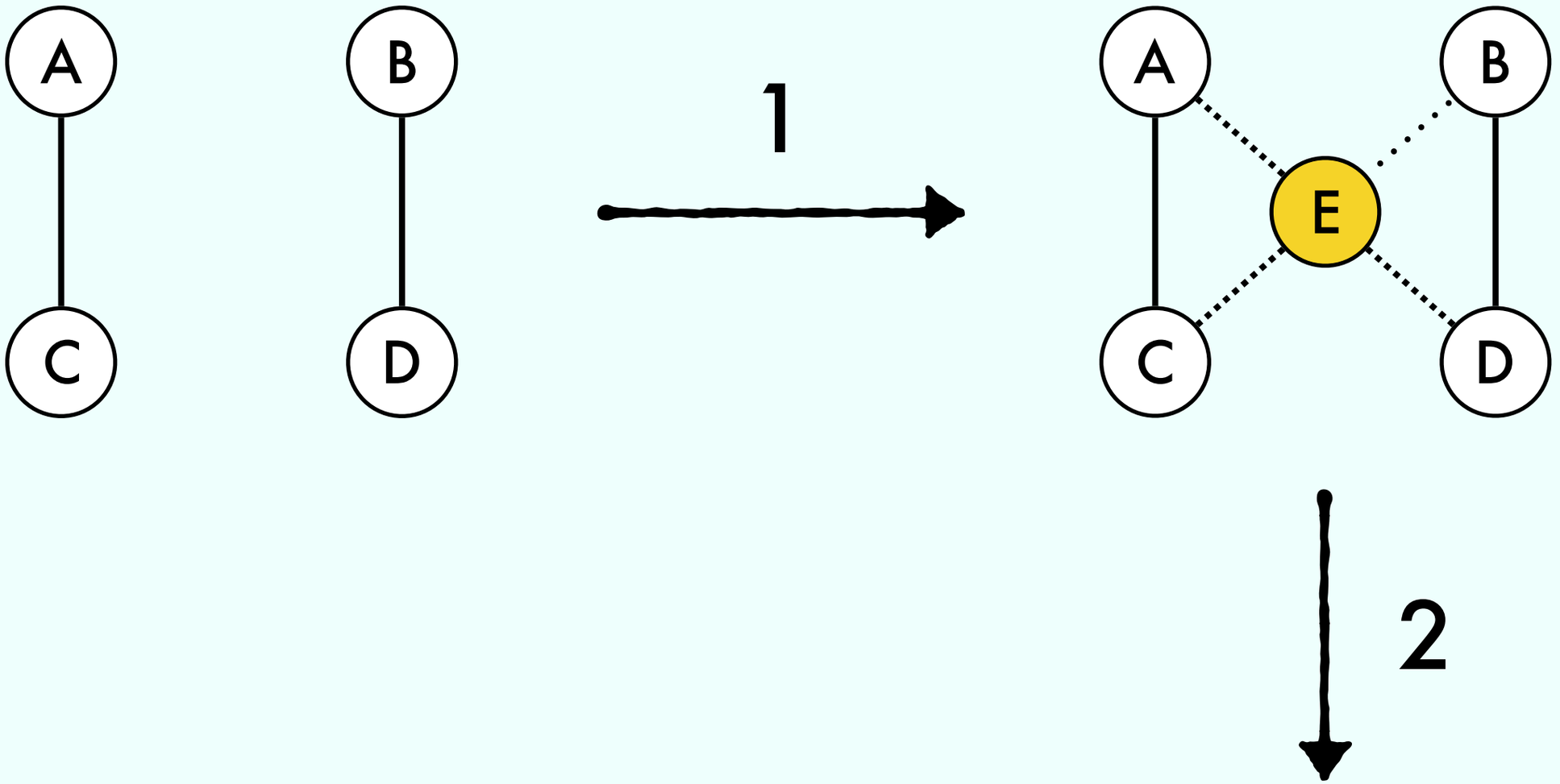

A
B
C
D
1
A
B
E
C
D
2
A
B
E
C
D
3

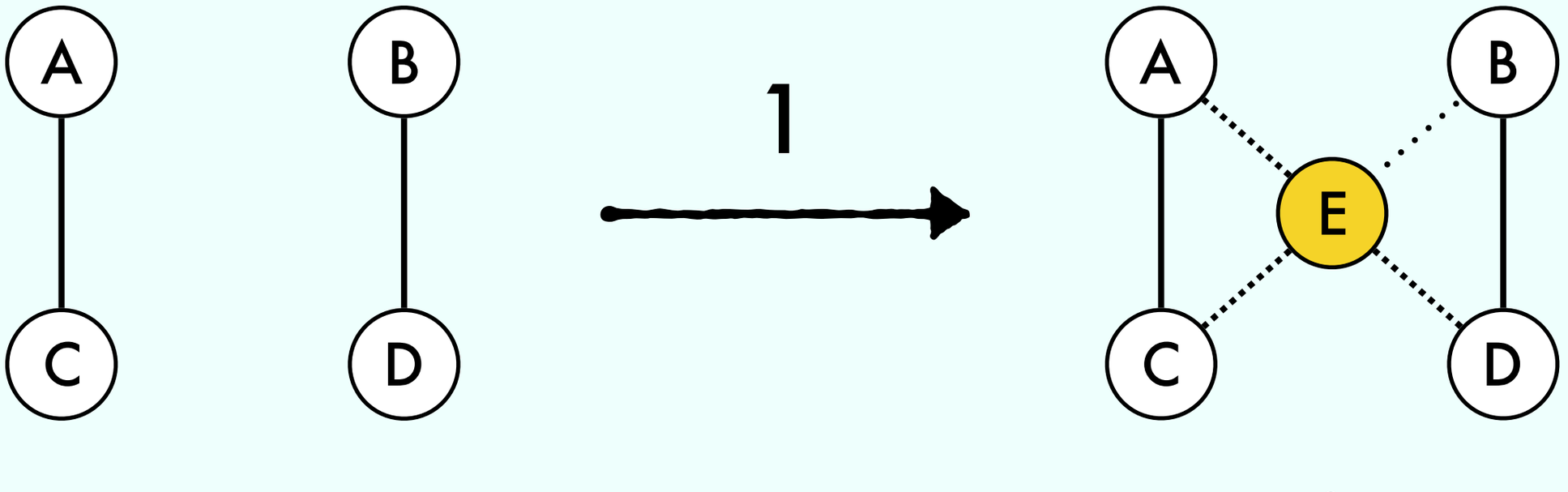

A
B
C
D
1
A
B
E
C
D
2
A
B
C
D
3
A
B
E
C
D

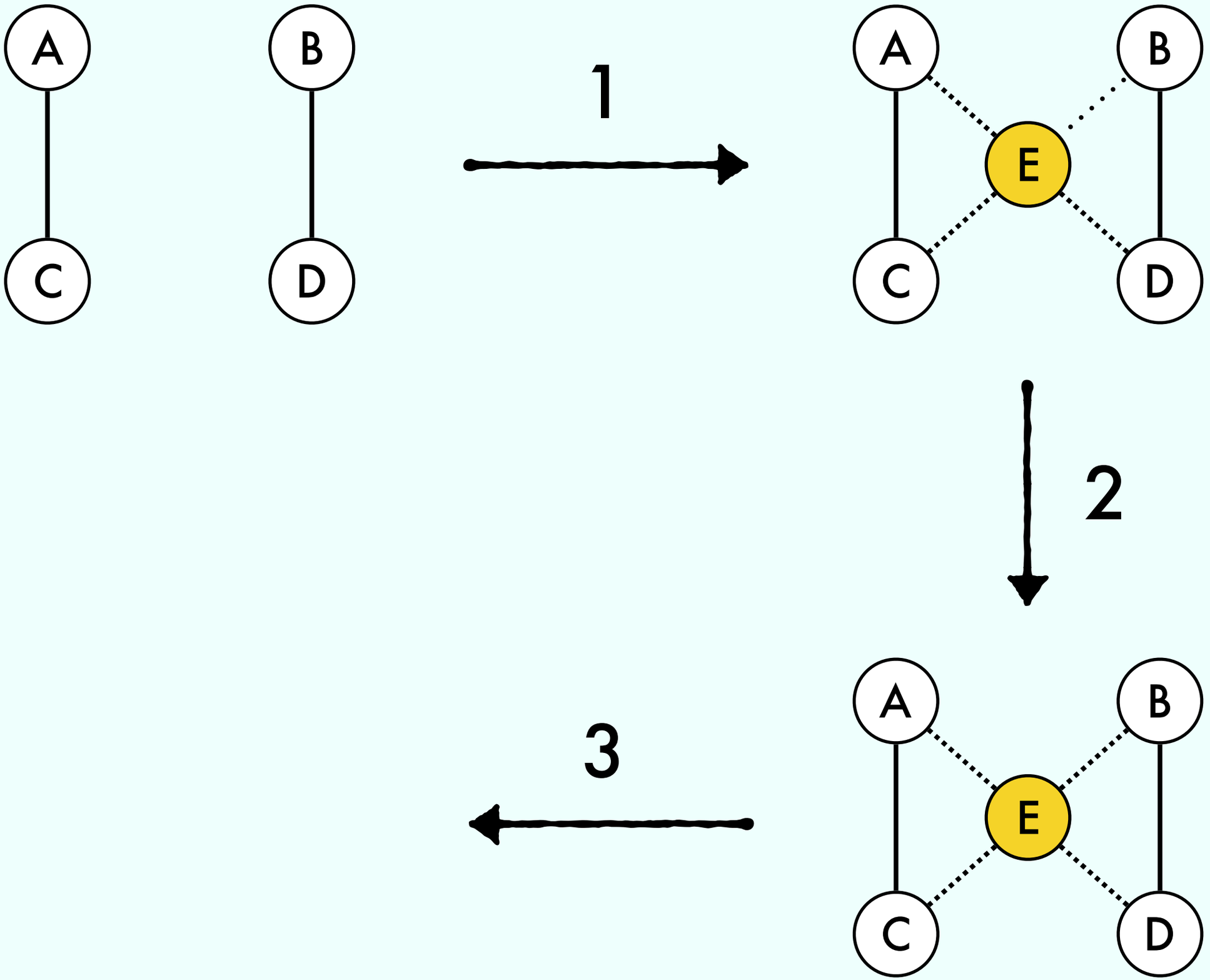

A
B
C
D

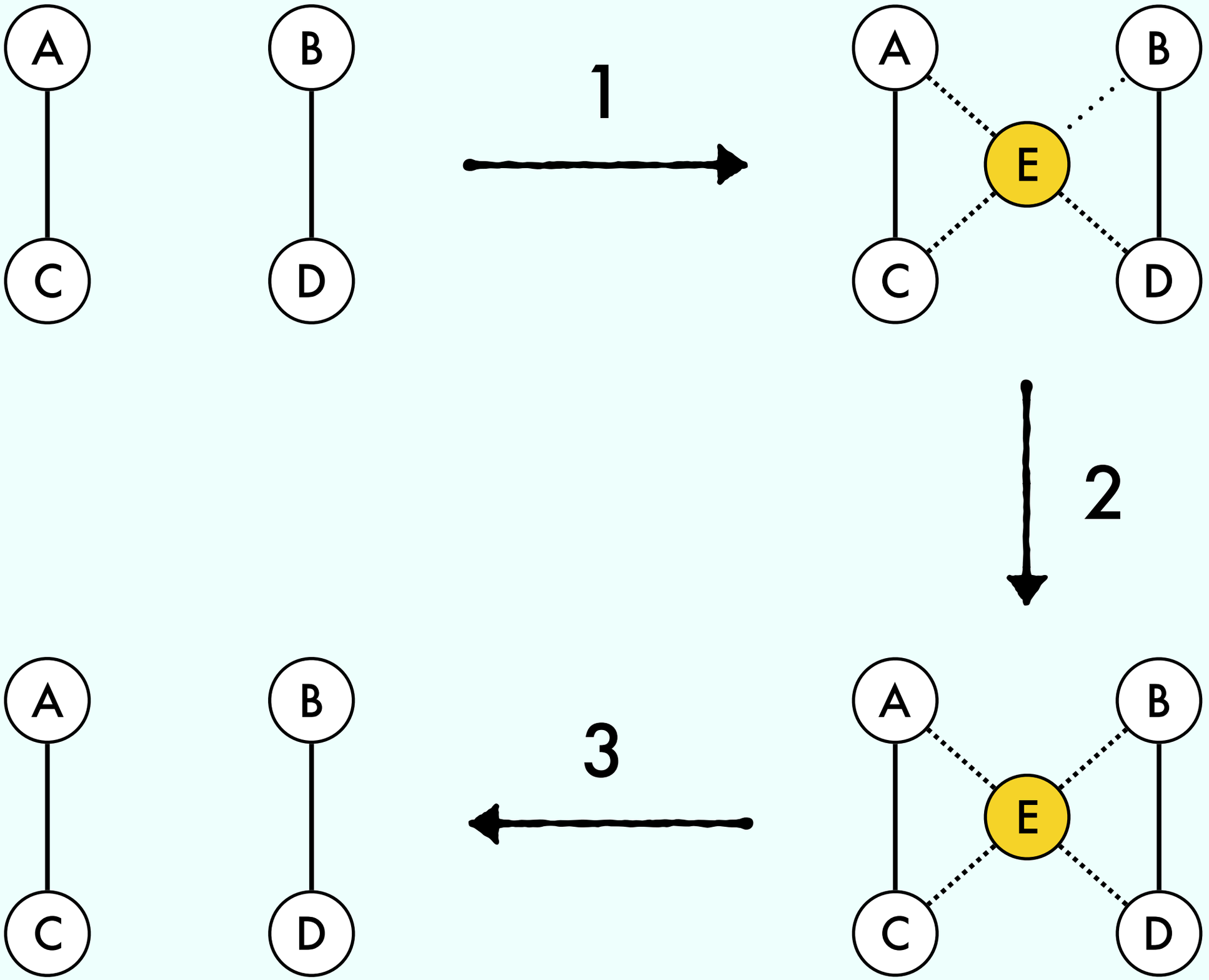

A
B
C
D
1


A
C
B
D
1
A
B
E
C
D


A
B
C
D
1
A
B
E
C
D
2


A
B
C
D
1
A
B
E
C
D
2
A
B
E
C
D


A
B
C
D
1
A
B
E
C
D
2
3
A
B
E
C
D


A
B
C
D
1
A
B
E
C
D
2
A
B
C
D
3
A
B
E
C
D

Theorem

H. Path in G

$\Leftrightarrow$

H. Cycle in G*