Algorithms and Probability

Week 1

Georg Hasebe — 22/02/2024

Zusammenhang

Definition 1.23. A graph $G = (V, E)$ is said to be k-connected (or k-vertex-connected) if it remains connected whenever fewer than k vertices are removed.

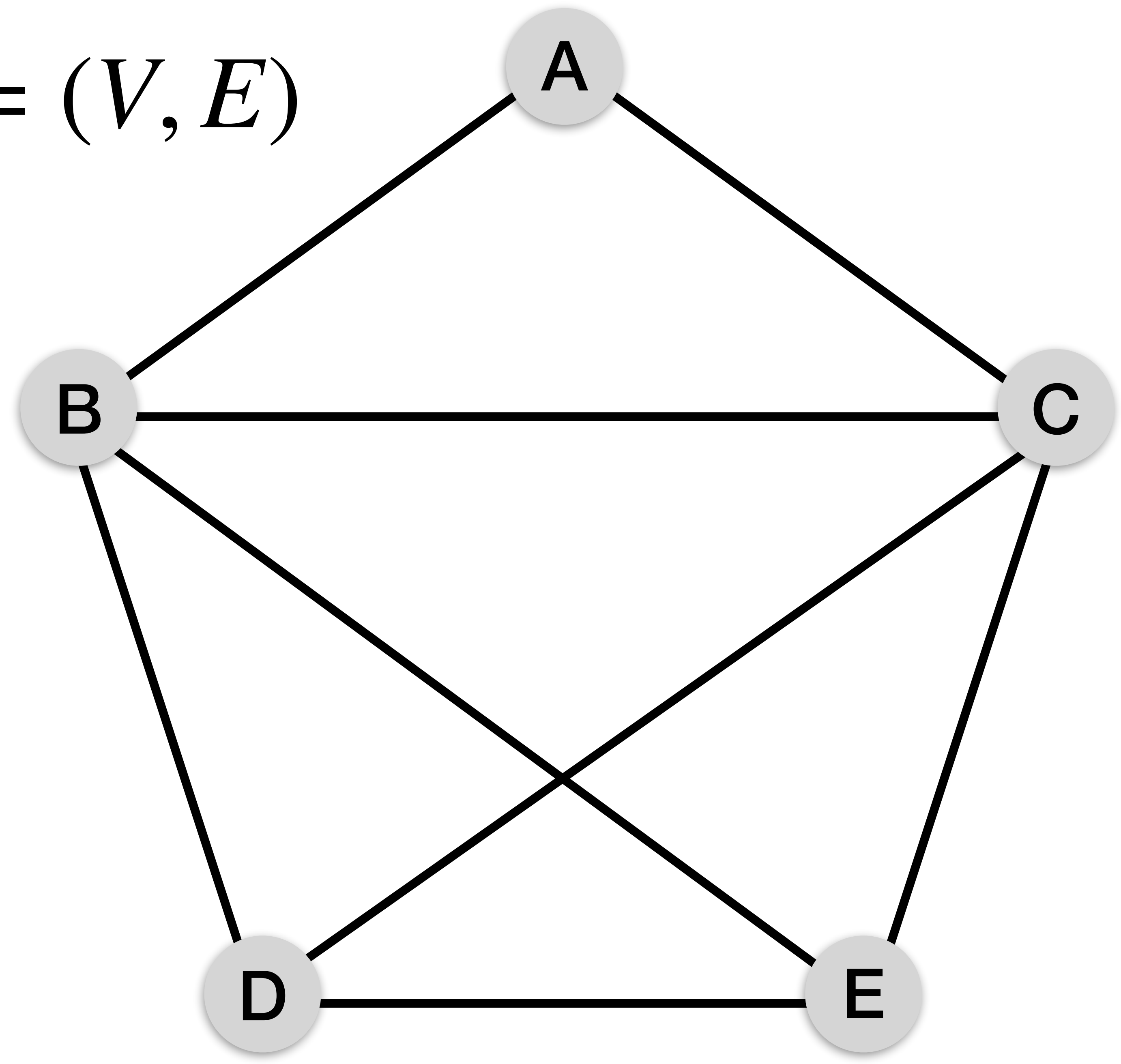
Definition 1.24. A graph $G = (V, E)$ is said to be k-edge-connected if it remains connected whenever fewer than k edges are removed.

Zusammenhang

- If the induced subgraph $G[V \backslash X]$, for $X \subseteq V$, is not connected anymore, then we call X a *vertex separator* (Knotenseparator).
- If $u, v \in V$ are in different connected components in $G[V \backslash X]$, then we call X a *u-v vertex separator* (u-v-Knotenseparator).
- Similarly, we can define *edge separator* (Kantenseparator) and *u-v edge separator* (u-v-Kantenseparator).

Examples

$G = (V, E)$



$X = \{B, C\}$

$G[V \backslash X]$

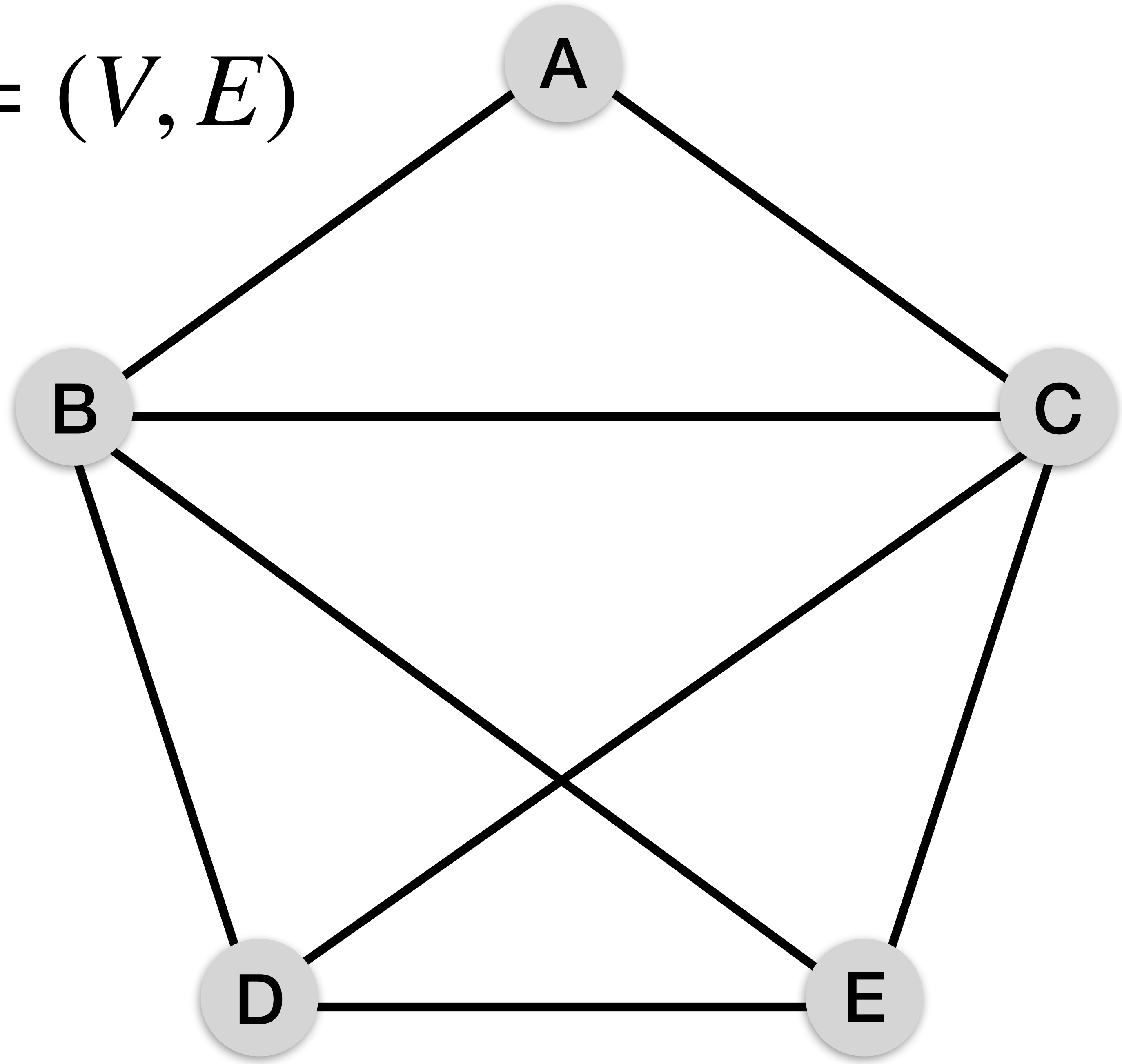
(*) G is 2-*connected*. Taking away any single vertex does not change G's connectivity.

(**) The induced subgraph $G[V \backslash X]$ is not connected. X is a *vertex separator*.

(***) X is a A-D *vertex separator*, because A and D are in different connected components after removing the vertices in X.

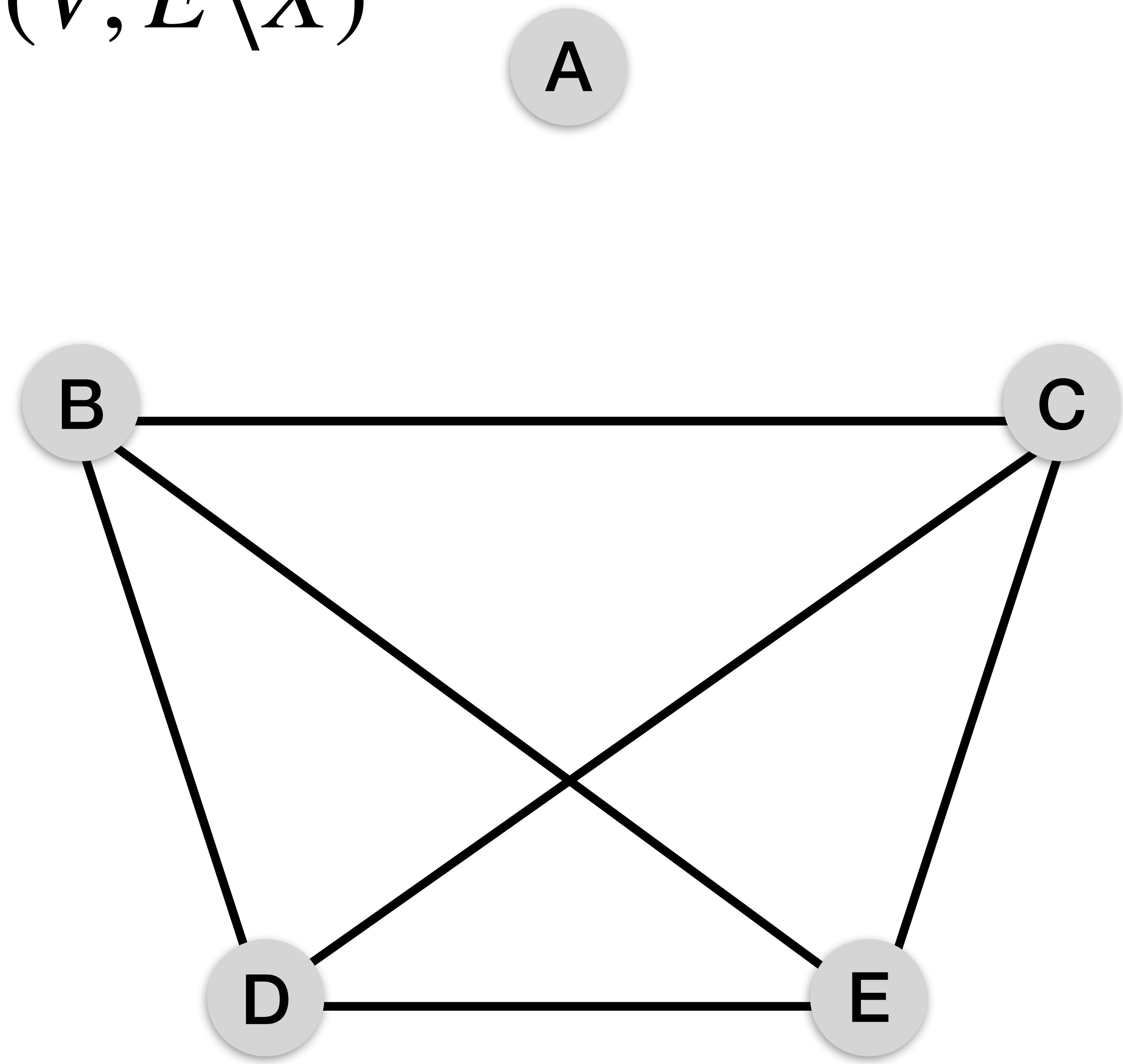
Examples

$$G = (V, E)$$



$$X = \{\{B, A\}, \{A, C\}\}$$

$$\tilde{G} = (V, E \backslash X)$$



(*) G is 2-*edge-connected*. Taking away any single edge does not change G's connectivity.

(**) $\tilde{G}$ is not connected. X is a *edge separator*.

(***) X is a A-B *edge separator*, because A and B are in different connected components after removing the edges in X.

Zusammenhang

- It is easy to see that any k-connected graph is also l-connected for $l < k$ (this also applies to k-edge-connected).
- We define the *vertex/edge connectivity* of G to be the biggest k such that G is k-connected/k-edge-connected.
- We have:

vertex connectivity $\leq$ edge connectivity $\leq$ minimal degree

Zusammenhang

Satz 1.25 (Menger). Sei $G = (V, E)$ ein Graph und $u, v \in V, u \neq v$. Dann gilt:

a) Jeder u-v-Knotenseparator hat Grösse mindestens $k \iff$ Es gibt mindestens k intern-knotendisjunkte u-v-Pfade.

b) Jeder u-v-Kantenseparator hat Grösse mindestens $k \iff$ Es gibt mindestens k kantendisjunkte u-v-Pfade.

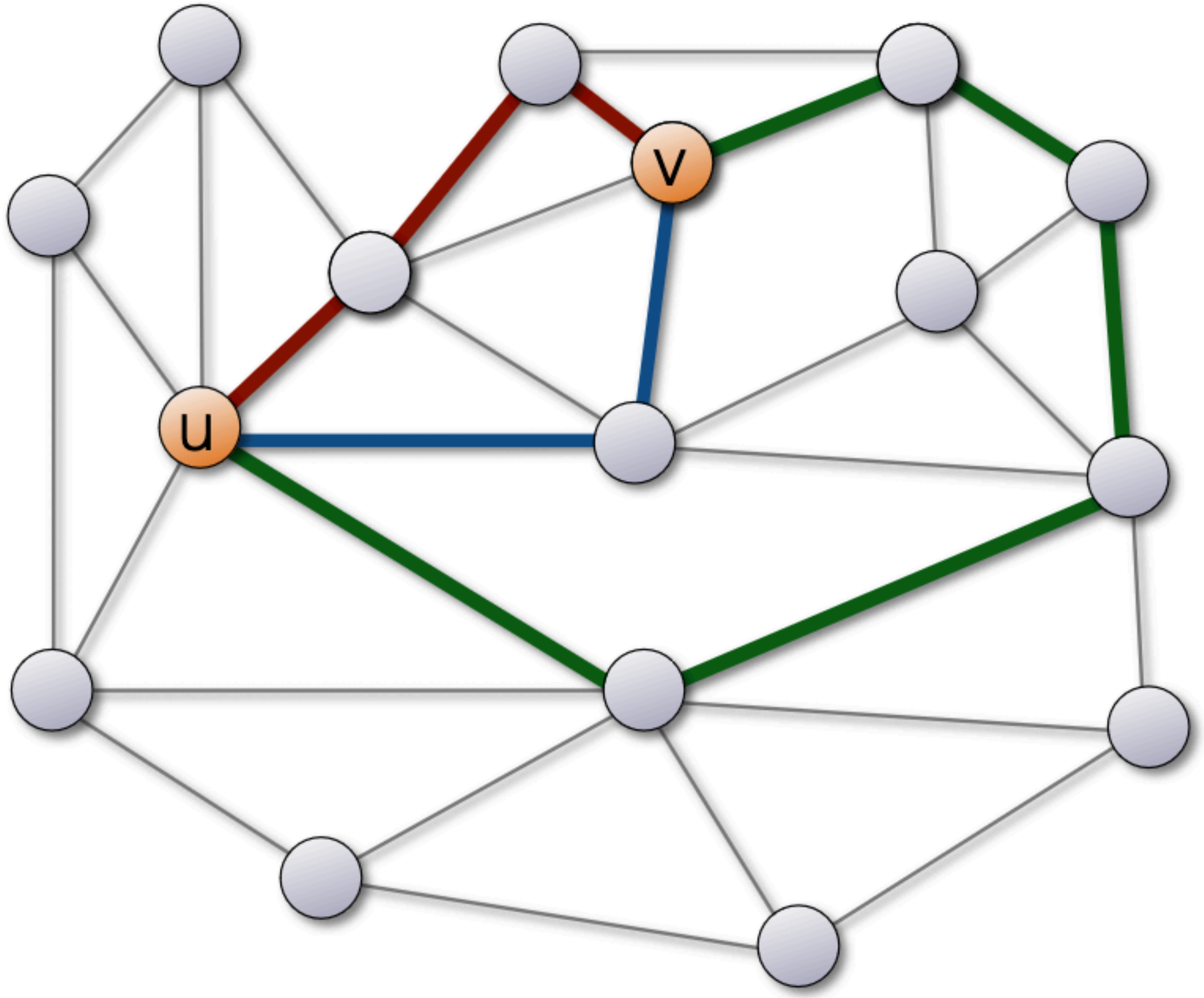
Without proof (for now).

Zusammenhang

Satz 1.25 (Menger). Sei $G = (V, E)$ ein Graph und $u, v \in V, u \neq v$. Dann gilt:

a) Jeder u-v-Knotenseparator hat Grösse mindestens $k \iff$ Es gibt mindestens k intern-knotendisjunkte u-v-Pfade.

b) Jeder u-v-Kantenseparator hat Grösse mindestens $k \iff$ Es gibt mindestens k kantendisjunkte u-v-Pfade.

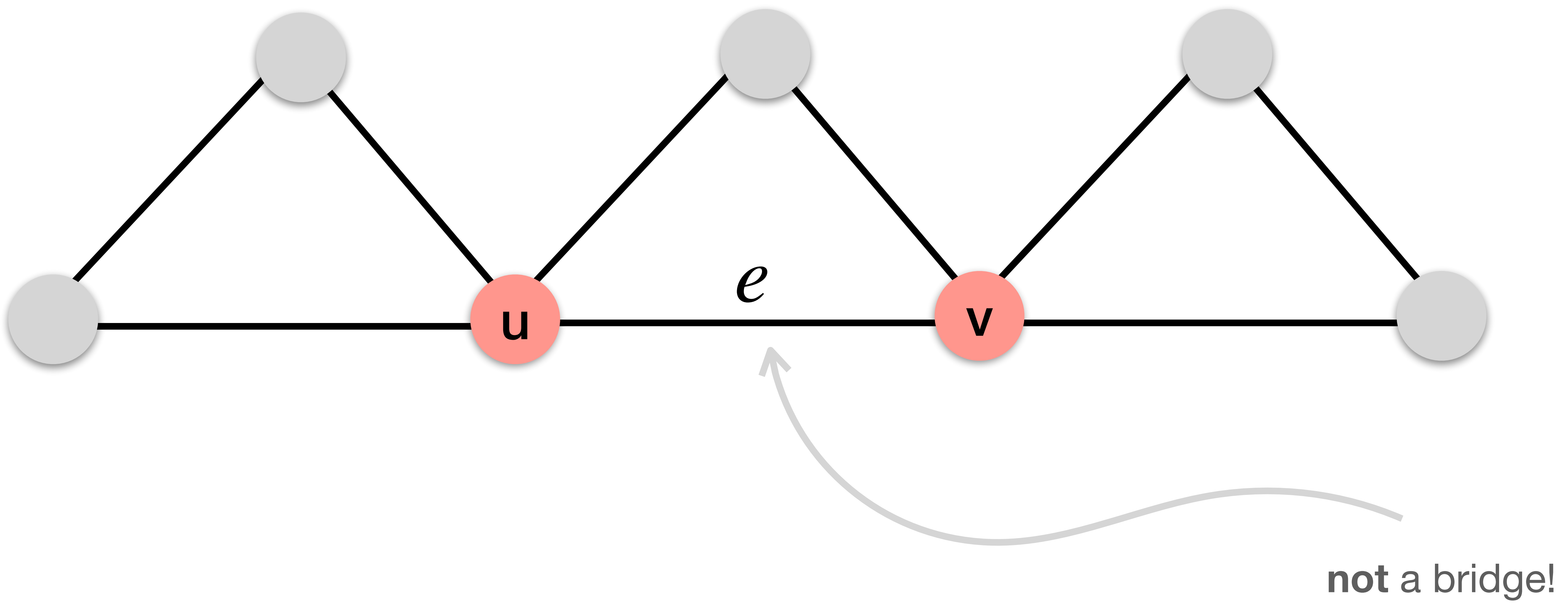


Zusammenhang

Lemma: Let $G = (V, E)$ be a connected graph. If $\{u, v\} \in E$ is a bridge, then u and v are cut vertices unless they have degree 1.

What about the other direction? Let $e = \{u, v\} \in E$ with u, v cut vertices. Is e a bridge?

Counterexample:



Blöcke

Definition: Sei $G = (V, E)$. Wir definieren eine Äquivalenzrelation auf E durch

$$e \sim f :\Longleftrightarrow \begin{cases} e = f, & \text{oder} \\ \exists \text{ Kreis durch } e \text{ und } f \end{cases}$$

Die Äquivalenzklassen nennen wir Blöcke.

DiscMath Recap

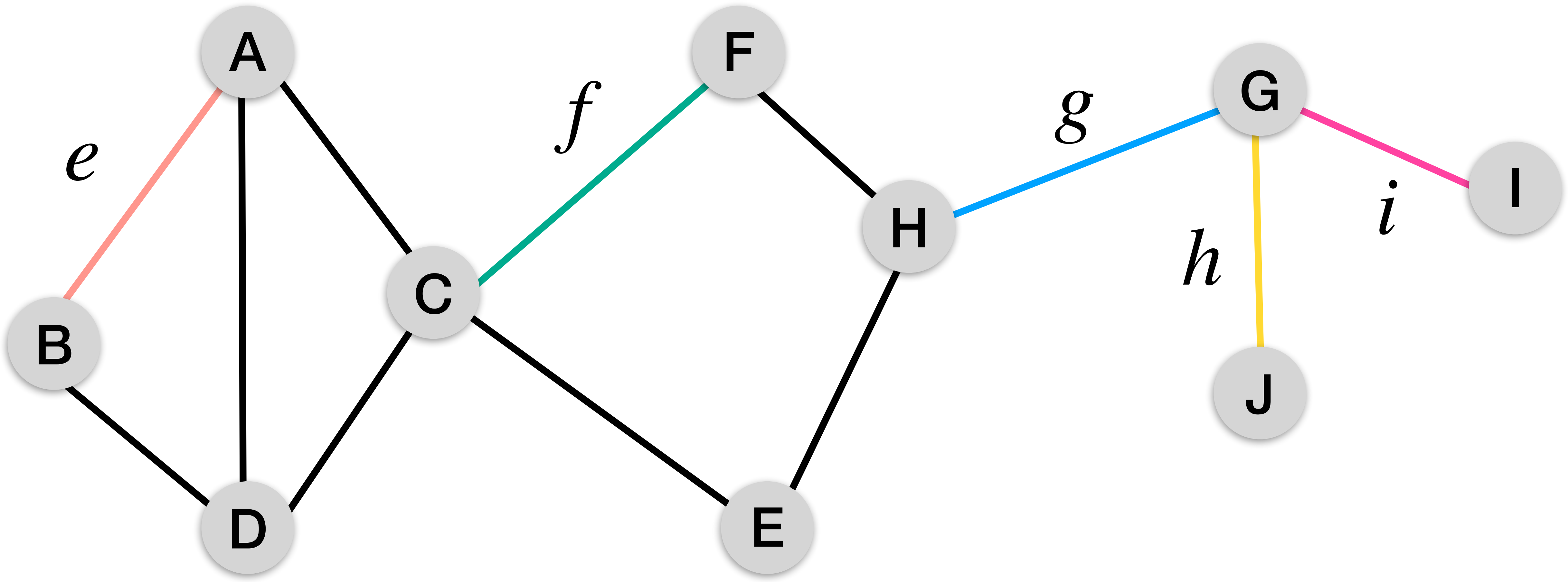
A binary relation $\sim$ on a set X is said to be an *equivalence relation* if and only if it is:

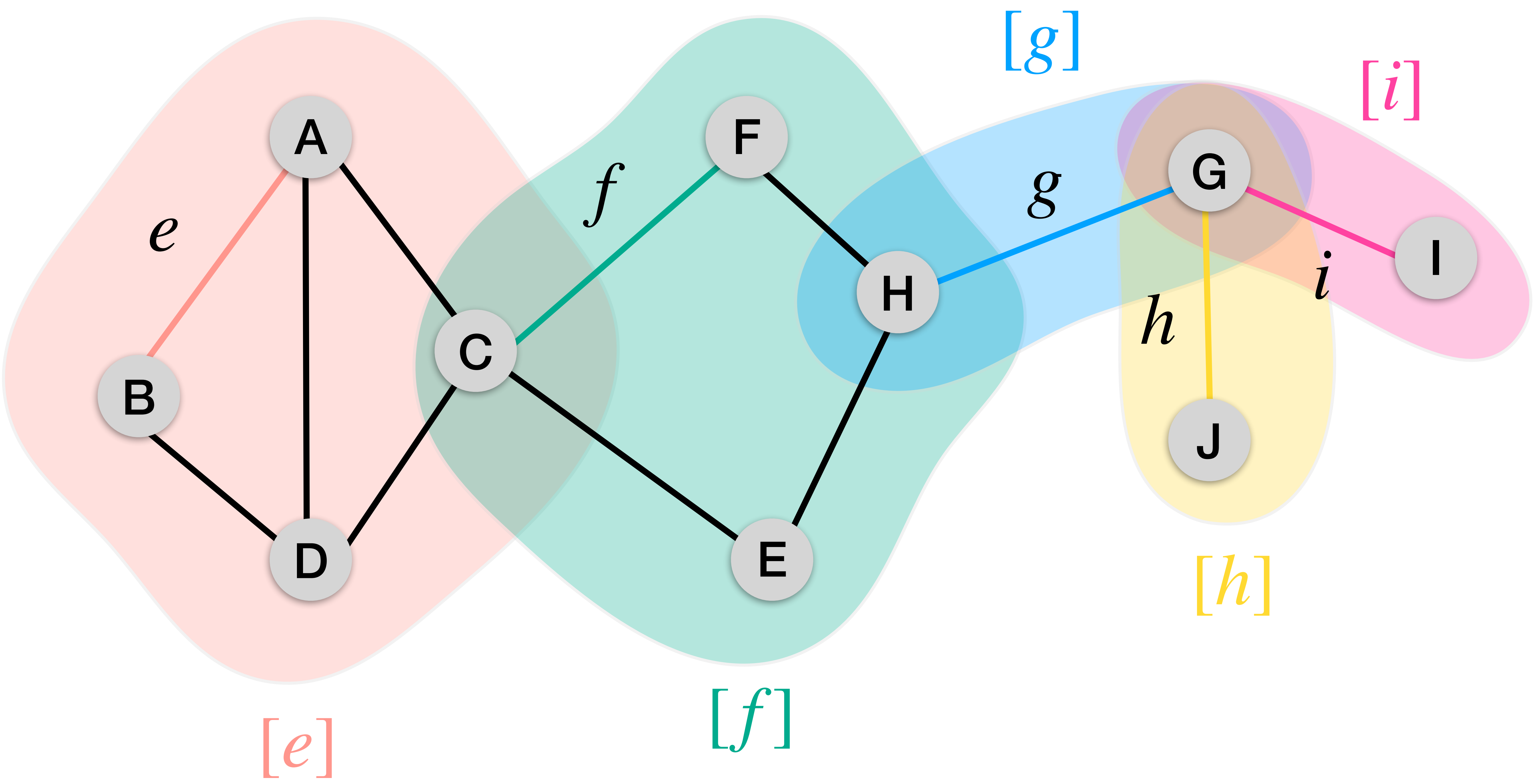
- Reflexive: $a \sim a$
- Symmetric: $a \sim b$ if and only if $b \sim a$
- Transitive: if $a \sim b$ and $b \sim c$ then $a \sim c$

For all $a, b, c \in X$.

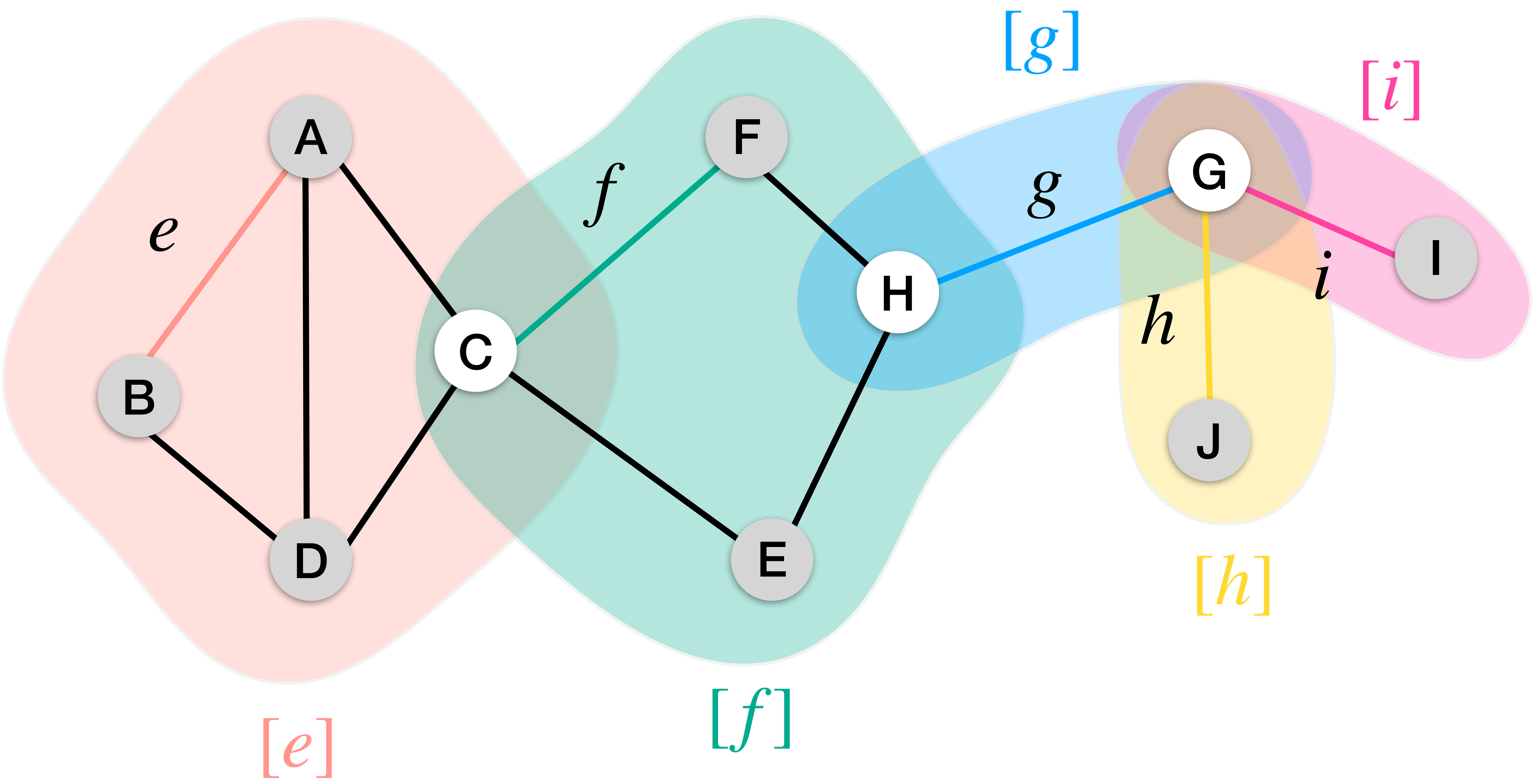
The *equivalence class* of a under $\sim$ is defined as $[a] := \{x \in X\colon x \sim a\}$.

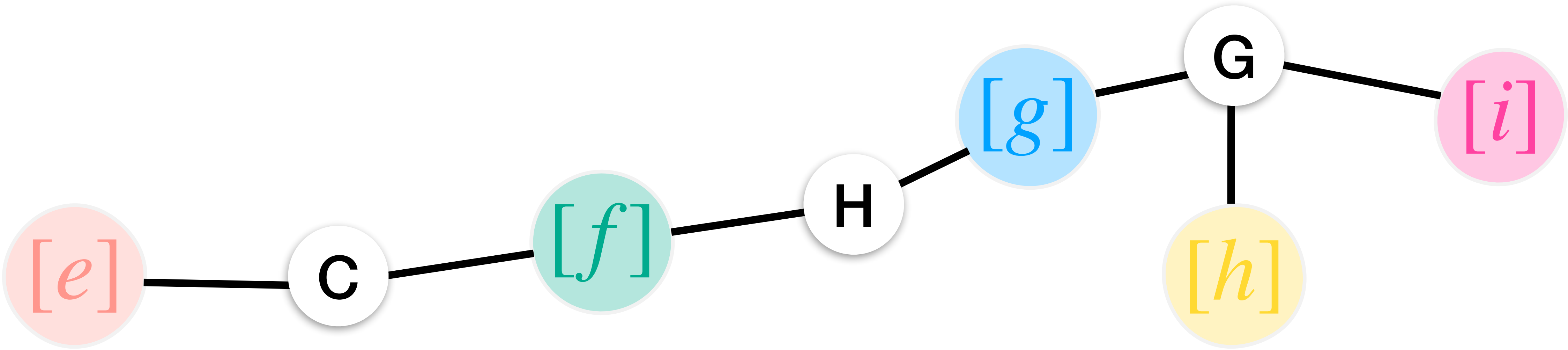
(*) Example: modulo 2 equivalence relation on $\mathbb{Z}$, then the two equivalence classes are even and odd numbers.


A
B
C
D
E
F
G
H
I
J
e
f
g
h
i



A
B
C
D
E
F
G
H
I
J
e
f
g
h
i
[e]
[f]
[g]
[h]
[i]

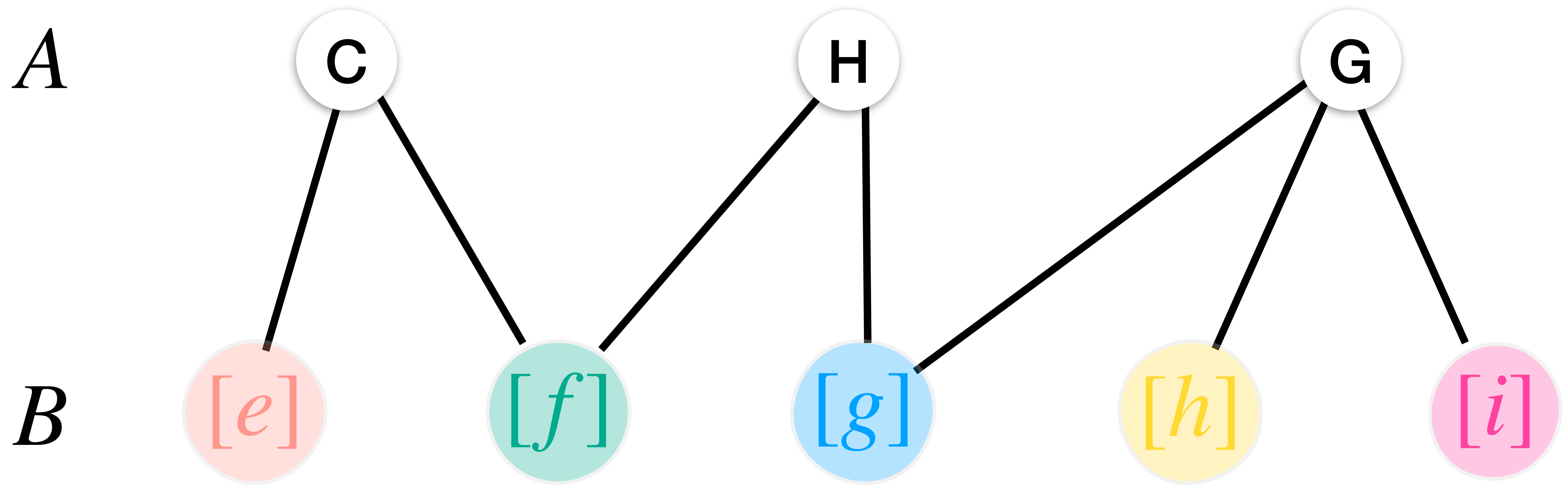

[g]
[i]
A
F
G
e
f
g
H
I
i
B
C
h
J
D
E
[h]
[f]
[e]


[e]
C
[f]
H
[g]
G
[h]
[i]

Definition: Sei $G = (V, E)$ ein zusammenhängender Graph.

Der Block-Graph von G ist der bipartite Graph $T = (A \uplus B, E_T)$ mit

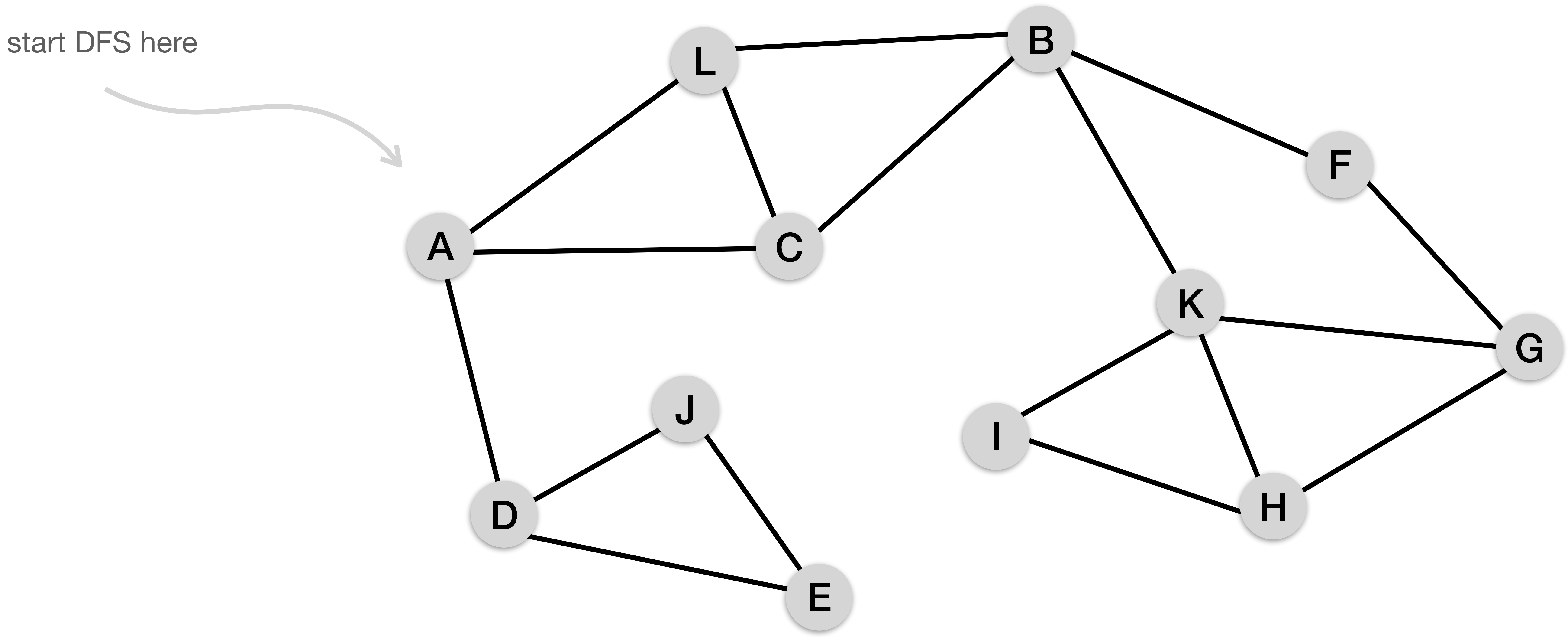
- $\mathrm{A} = \{\text{Artikulationsknoten von } G\}$.
- $\mathrm{B} = \{\text{Blöcke von } G\}$.
- $\forall a \in A, b \in B: \quad \{a, b\} \in E_T \iff a \text{ inzident zu einer Kante in } b$.



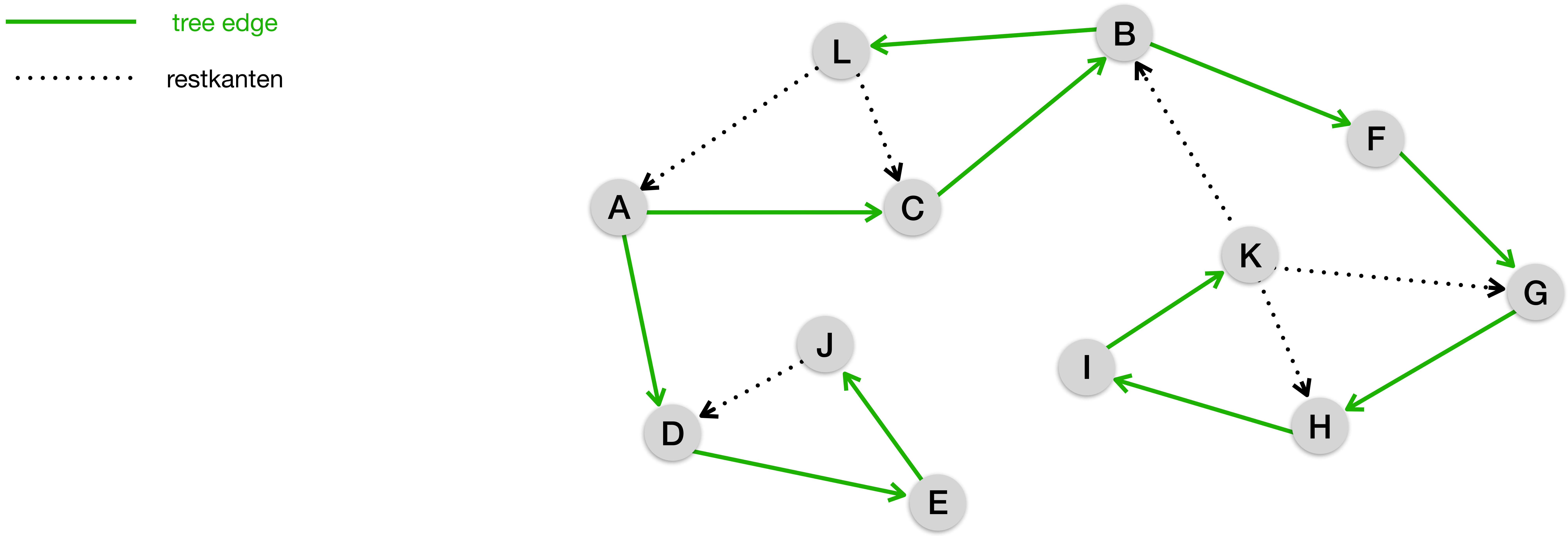
DFS for finding bridges/cut vertices

- Given an undirected graph $G = (V, E)$, find the bridges and cut vertices.
- Naive: remove edges/vertices and check for connectedness.
- As we will see, there is a more efficient approach using DFS.

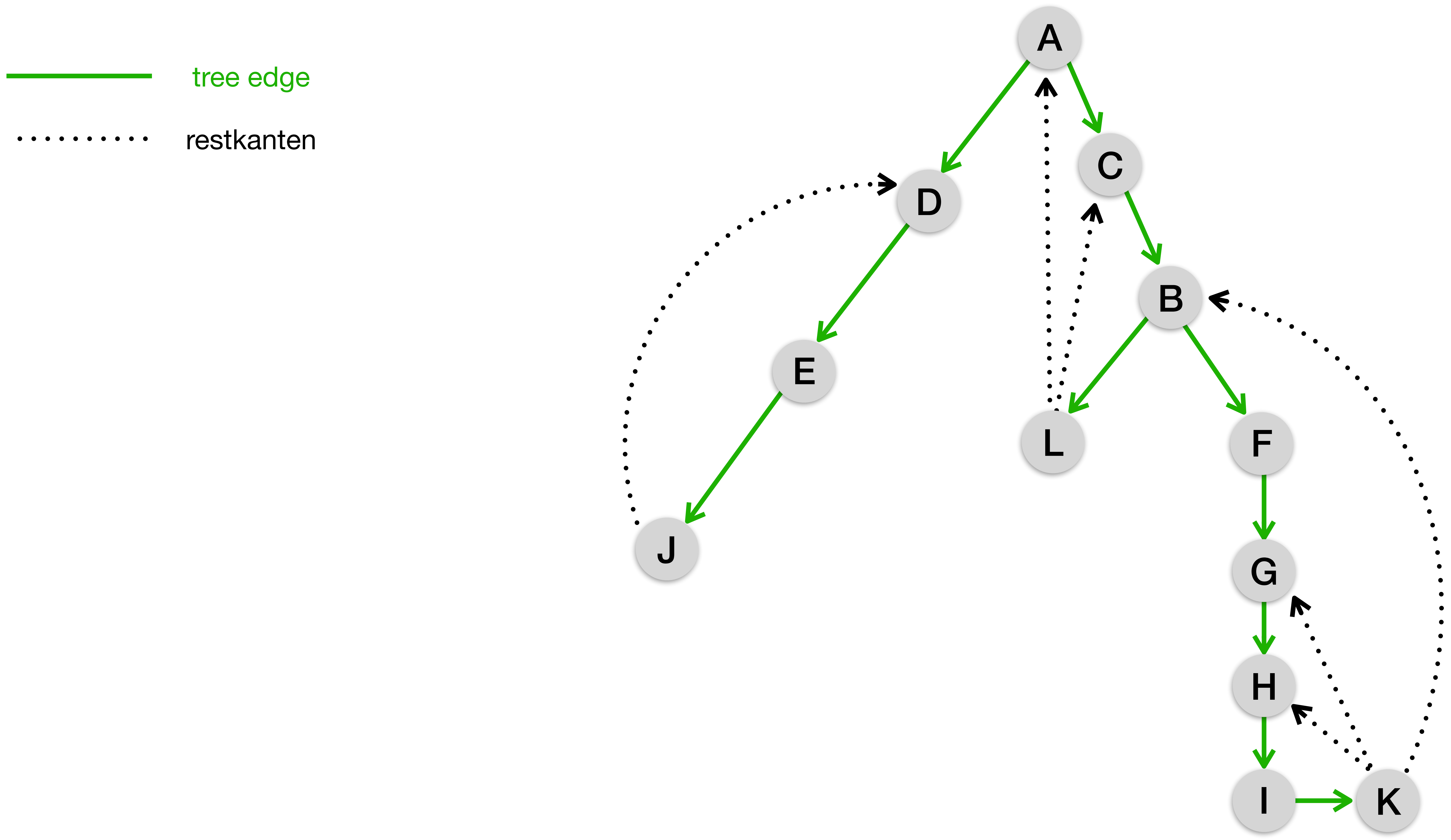
DFS recap



DFS recap

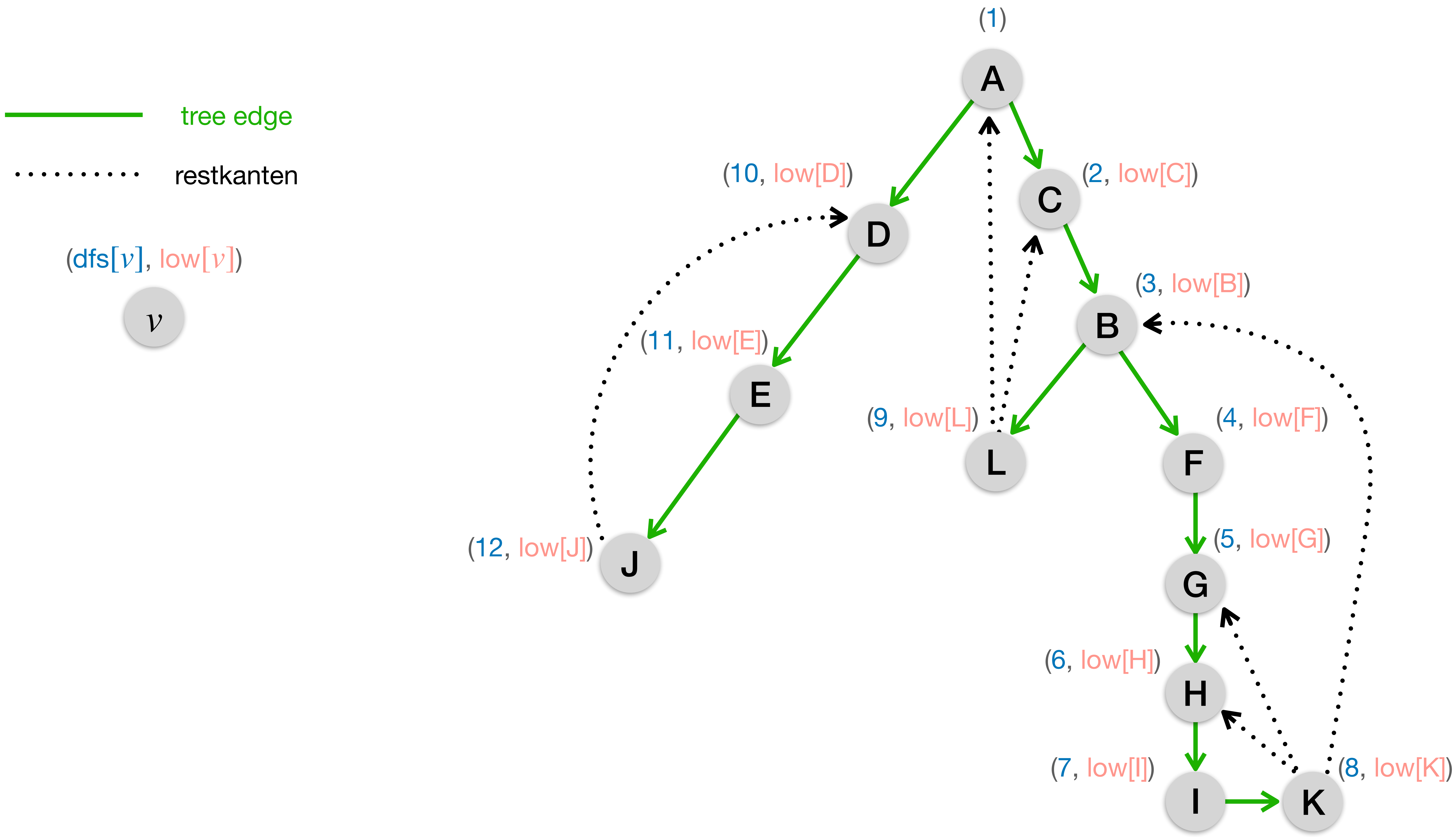


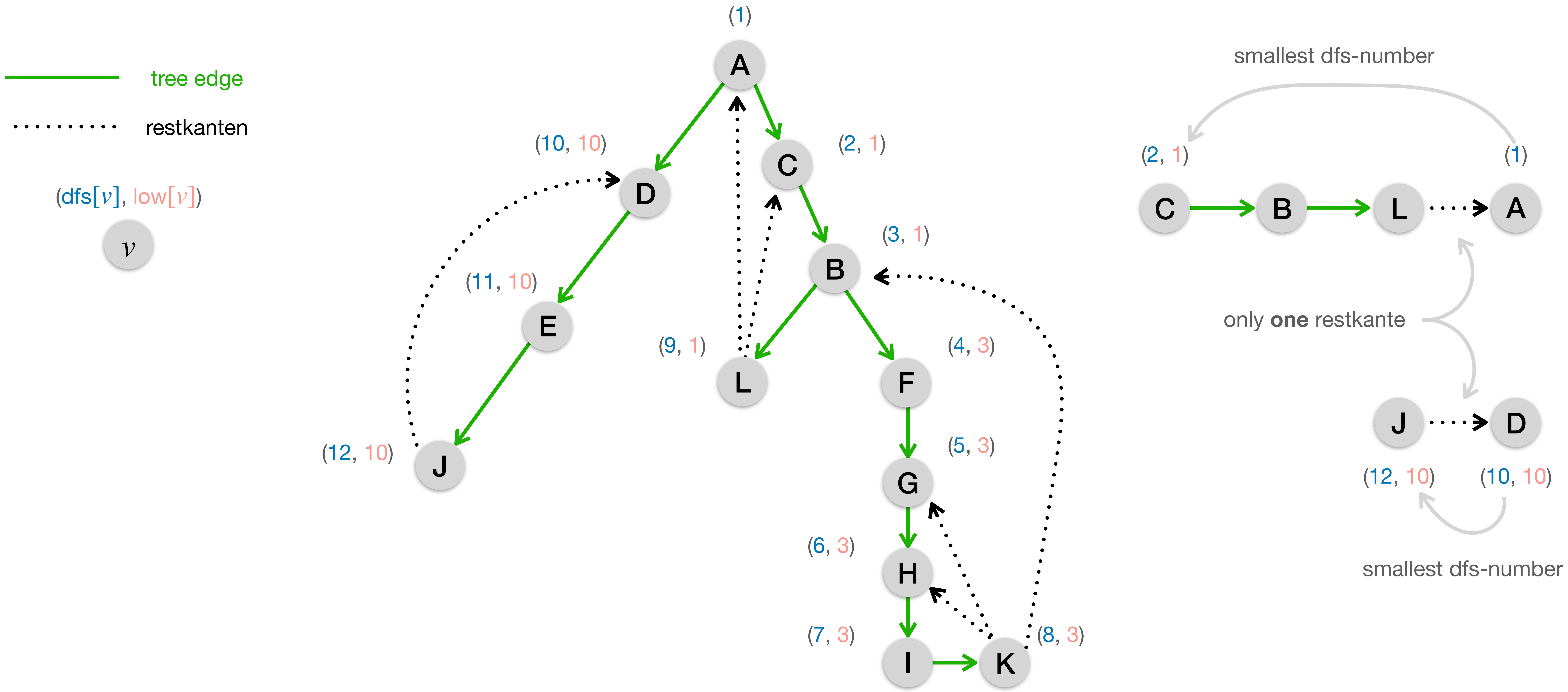
DFS recap



DFS for finding bridges/cut vertices

- We extend traditional DFS by maintaining the following information throughout iteration:
  - dfs$[v]$ … the time DFS “entered” vertex v (dfs$[r] = 1$, where r is the root of the DFS tree).
  - low$[v]$ … the lowest entry time dfs$[w]$ we can reach from v through a directed path consisting of an arbitrary number of tree edges and a single restkante.
- The root will be treated separately, so we don’t define low$[r]$, where r is the root of the DFS tree.


tree edge
restkanten
(dfs[v], low[v])
v
(1)
A
(10, low[D])
D
(2, low[C])
C
(3, low[B])
B
(11, low[E])
E
(9, low[L])
L
(4, low[F])
F
(12, low[J])
J
(5, low[G])
G
(6, low[H])
H
(7, low[I])
I
(8, low[K])
K



tree edge
restkanten
(dfs[v], low[v])
v
(1)
A
(10, 10)
D
(2, 1)
C
(3, 1)
B
(11, 10)
E
(9, 1)
L
(4, 3)
F
(12, 10)
J
(5, 3)
G
(6, 3)
H
(7, 3)
I
(8, 3)
K
smallest dfs-number
(2, 1)
(1)
C
B
L
A
only **one** restkante
J
D
(12, 10)
(10, 10)
smallest dfs-number

Proof

Let $v \in V$ such that v is **not the root** of the DFS tree.

We show that v is a cut vertex if and only if v has a neighbor u such that $\text{low}[u] \geq \text{dfs}[v]$.

$(\Rightarrow)$

Proof

Assume that v is a cut vertex. Then $G[V \backslash \{v\}]$ has at least 2 connected components Z_1 and Z_2. Without loss of generality, assume $s \in Z_1$.

Every path from s to a vertex in Z_2 must include v, we have
$1 = \text{dfs}[s] < \text{dfs}[v] < \text{dfs}[w] \quad \forall w \in Z_2$.

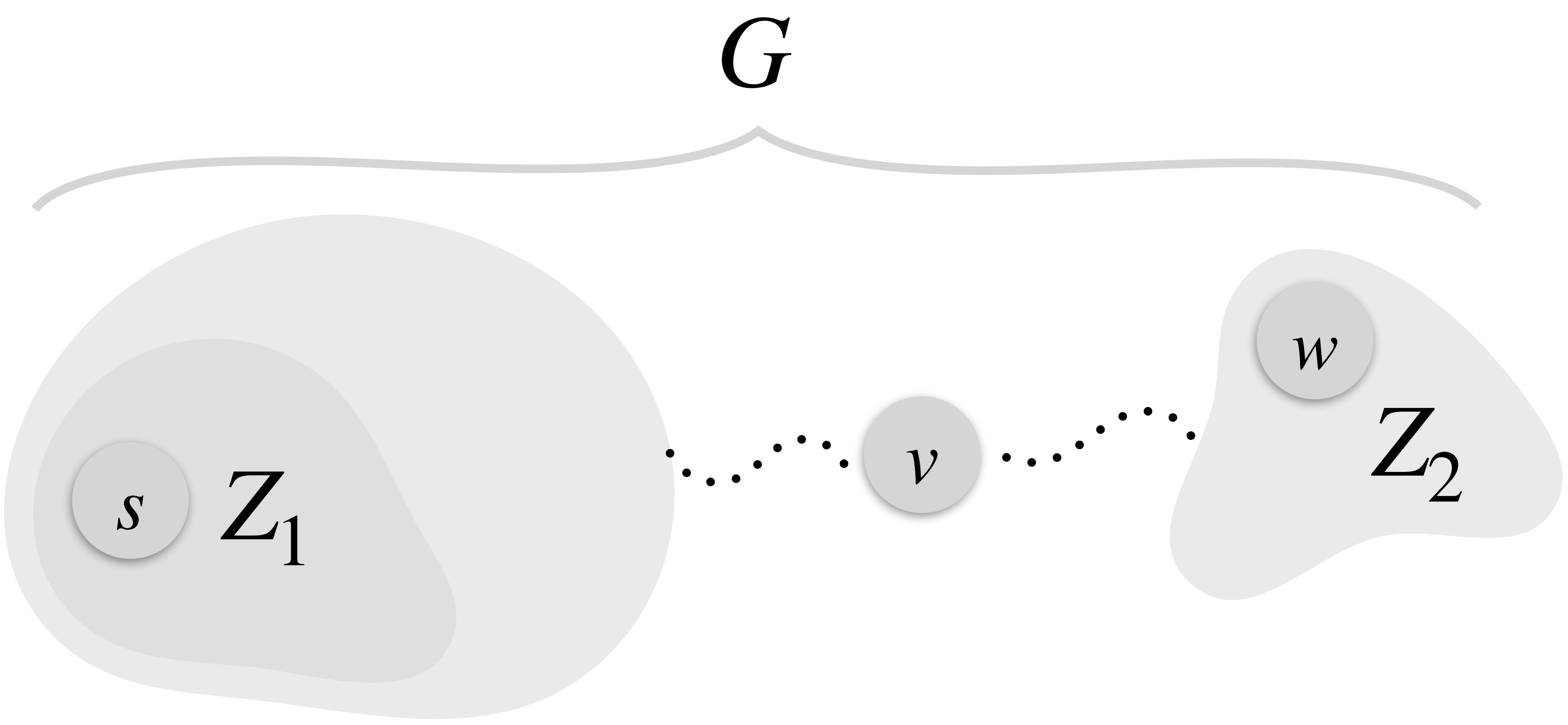
Since $G[Z_2]$ is a connected component in $G[V \backslash \{v\}]$, there cannot be an edge from $w \in Z_2$ to a vertex $u \in V \backslash (\{v\} \cup Z_2)$.

($\Rightarrow$)

Proof

…

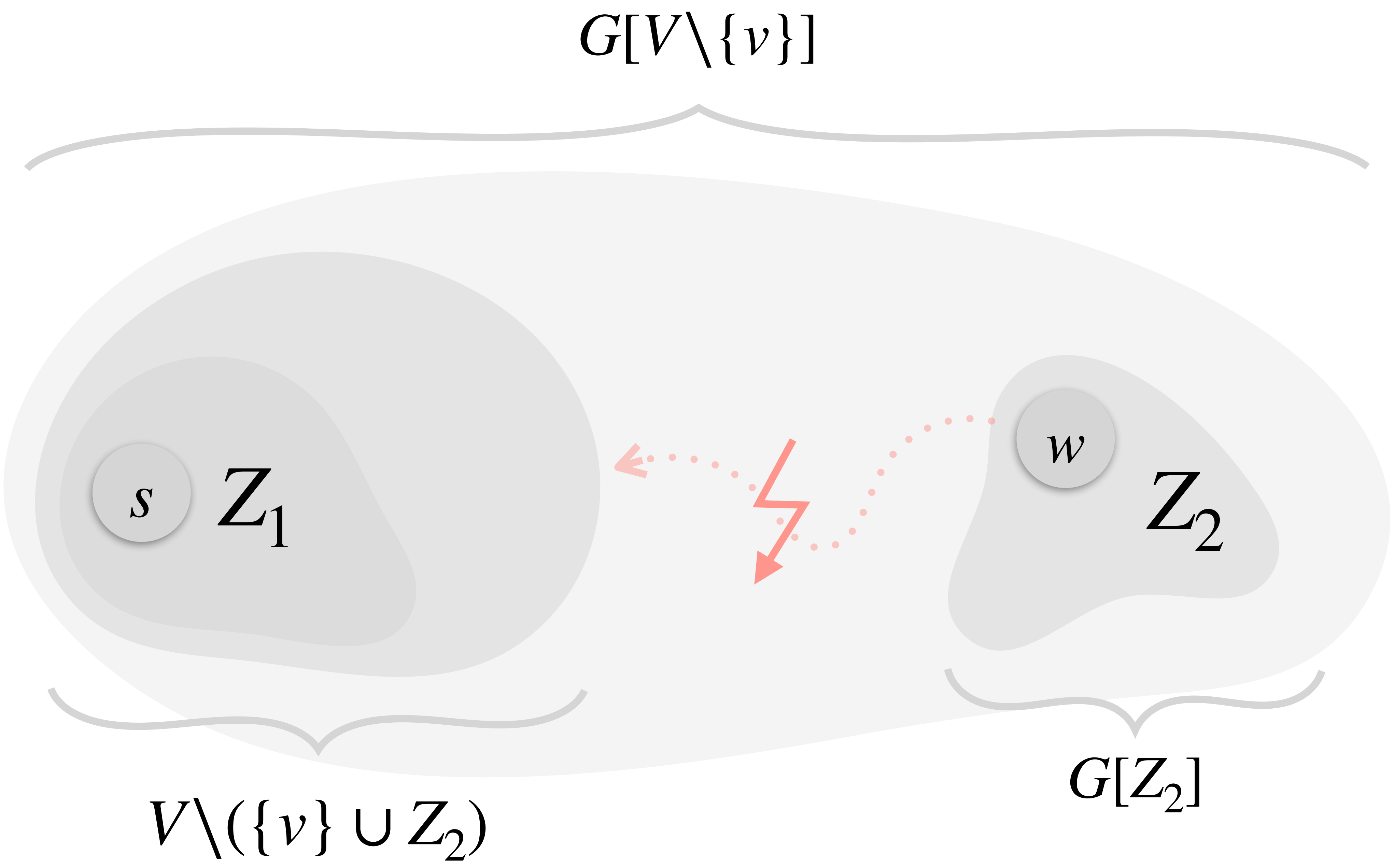
Since $G[Z_2]$ is a connected component in $G[V \backslash \{v\}]$, there cannot be an edge from $w \in Z_2$ to a vertex $u \in V \backslash (\{v\} \cup Z_2)$.



there could possibly be more than 2 connected components after removing v hence Z_1 is only part of the bigger shaded region which contains all vertices that are not in Z_2 or v itself.

remove v

$\Longrightarrow$



if there was such an edge, then u would be connected to Z_2 and therefore element of Z_2 but $V \backslash (\{v\} \cup Z_2)$ doesn't contain vertices from Z_2.

$(\Rightarrow)$

Proof

…

Since $G[Z_2]$ is a connected component in $G[V \setminus \{v\}]$, there cannot be an edge from $w \in Z_2$ to a vertex in $u \in V \setminus (\{v\} \cup Z_2)$.

Thus low$[w]$ is at least dfs$[v]$ for all $w \in Z_2$. Since v is connected to Z_2, it has at least one neighbor $w \in Z_2$ such that low$[w] \geq$ dfs$[v]$.

DiskMath Recap

Contraposition:

$P \rightarrow Q$ is equivalent to $\neg Q \rightarrow \neg P$

Here:

v is a cut vertex $\Leftarrow$ v has a neighbor u such that low$[u]$ $\geq$ dfs$[v]$

$\Leftrightarrow$

v is **not** a cut vertex $\Rightarrow$ v **has no** neighbor u such that low$[u]$ $\geq$ dfs$[v]$

using contraposition

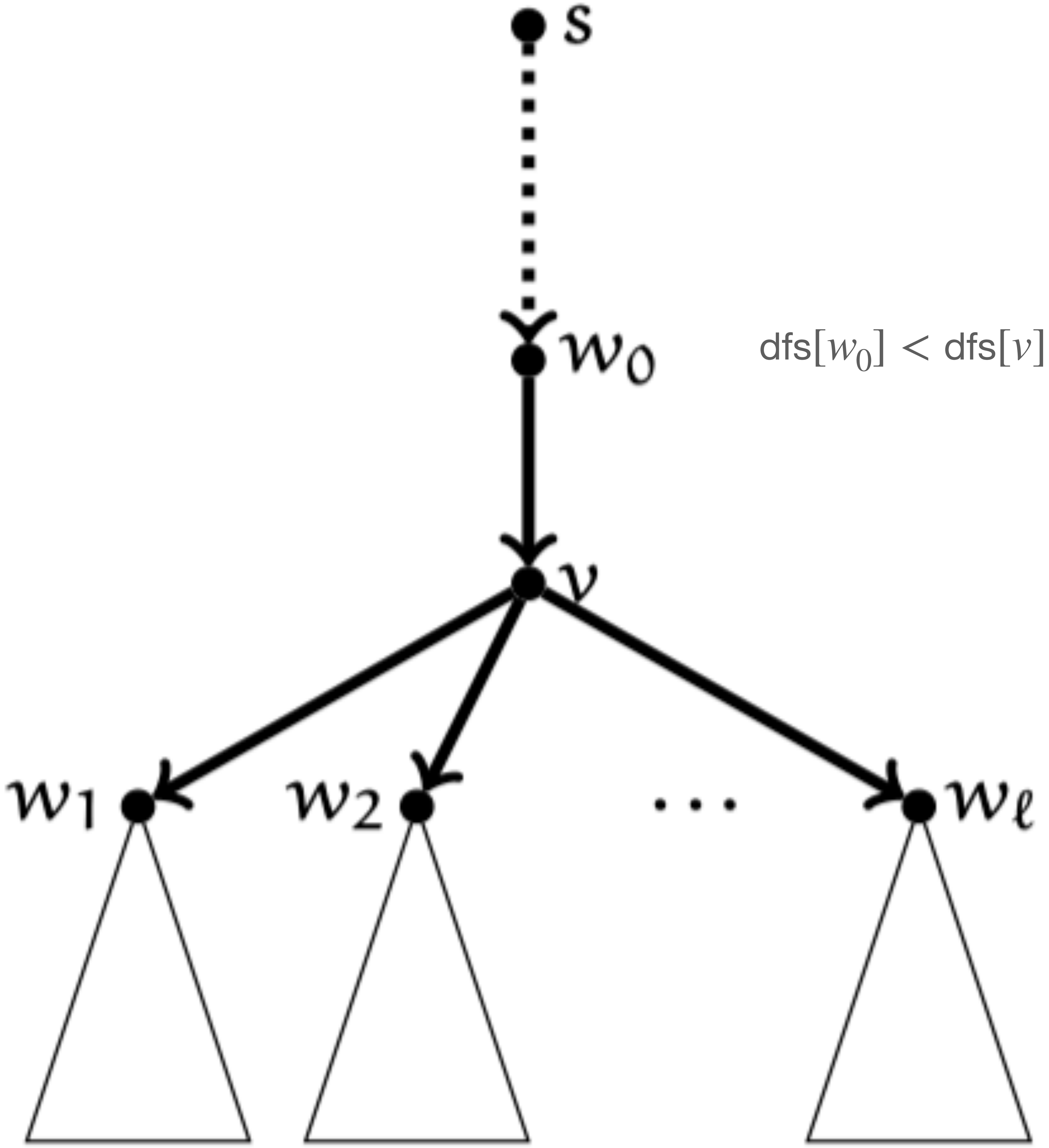
$(\Leftarrow)$

Proof

Assume v is not a cut vertex.

Let T be the DFS tree rooted at s and let $w_0, \ldots, w_l$ be all the neighbors of v in G. Without loss of generality, assume dfs$[w_0] <$ dfs$[v]$.

Proof



dfs$[w_0] <$ dfs$[v]$ since we reach w_0 earlier from s than v.

by construction of the DFS algorithm, the subtrees rooted at $w_1, \ldots, w_l$ cannot be connected.

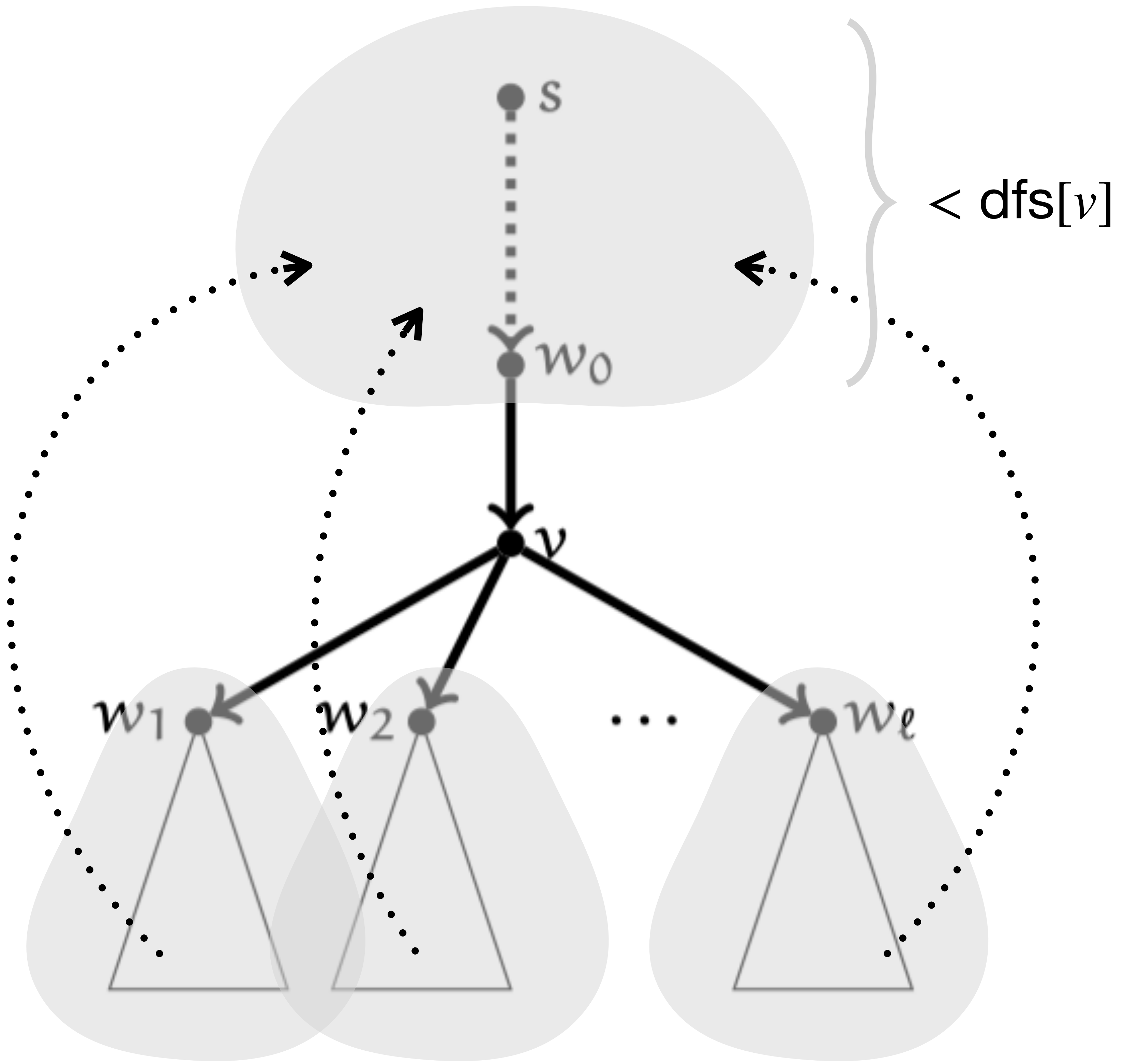
but they have to be connected in G, otherwise v would be a cut vertex.

$(\Leftarrow)$

Proof

…

For every neighbor $w_1, \ldots, w_l$ there exists a path using a restkante to a vertex with smaller dfs-number.



using contraposition

$(\Leftarrow)$

Proof

…

For every neighbor $w_1, \ldots, w_l$ there exists a path using a restkante to a vertex with smaller dfs-number.

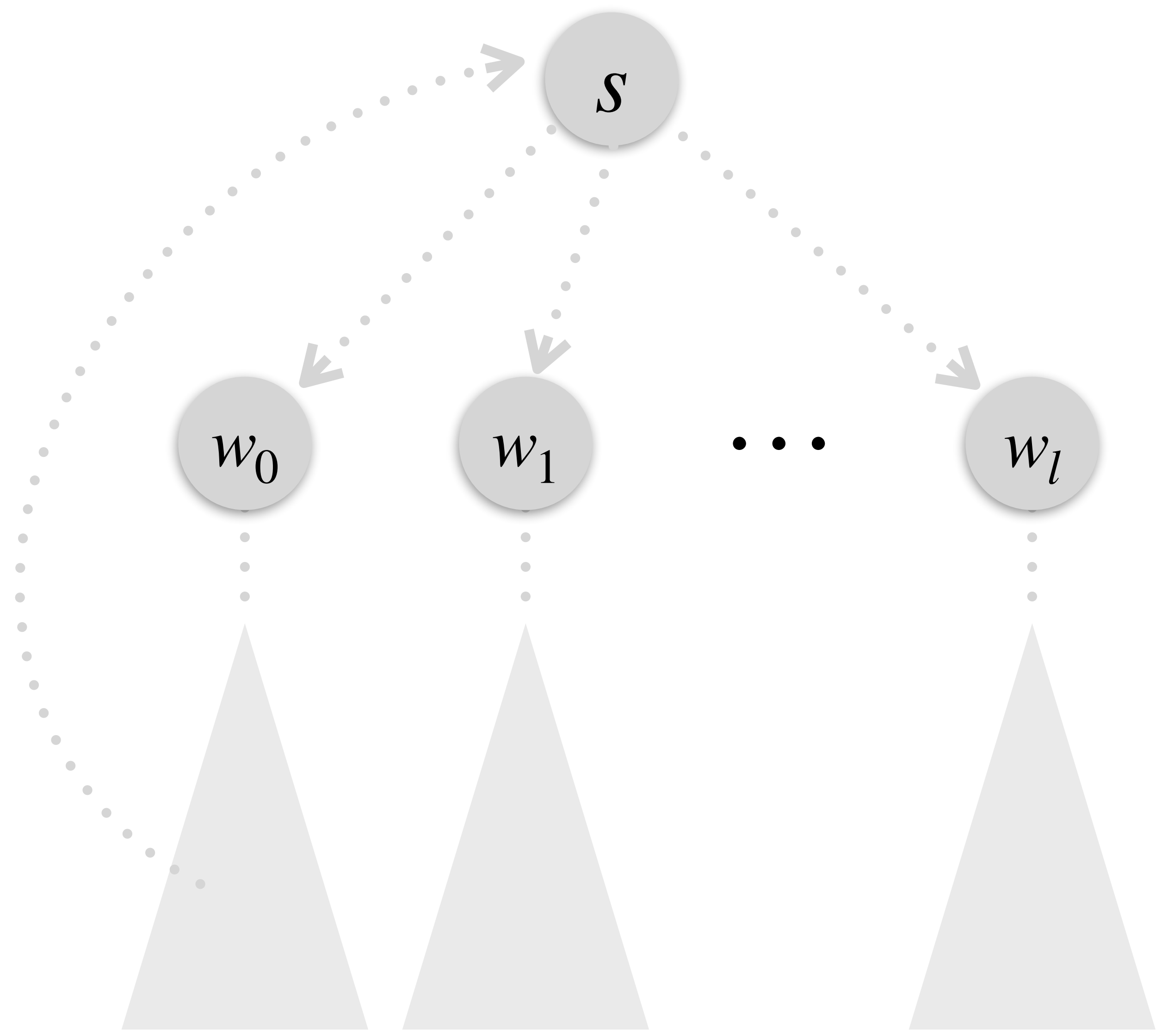
Thus low$[w]$ is greater than dfs$[v]$ for all neighbors w of v, meaning that there is no neighbor w of v such that low$[v] \geq$ dfs$[u]$.

What about the root?

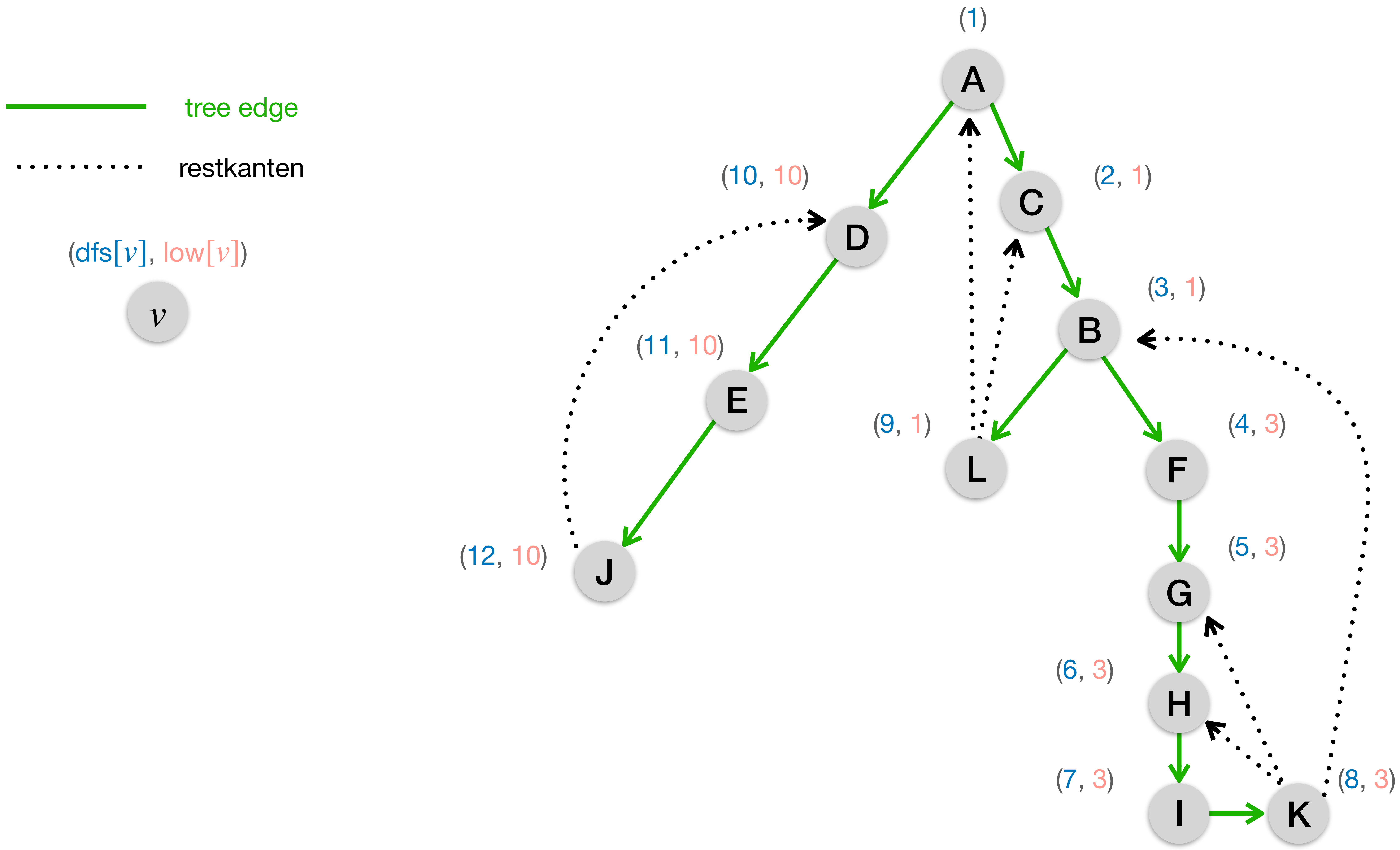
Let T be the DFS tree **rooted at** s. If $\deg(s) \geq 2$ then s is a cut vertex.

Proof. Assume $\deg(s) = l \geq 2$. By construction of the DFS algorithm, the subtrees rooted at $w_1, \ldots, w_l$ cannot be connected. Even if there was a restkante from a subtree to s, after removing s the vertices contained in the subtrees become disconnected in $G[V \backslash \{s\}]$.

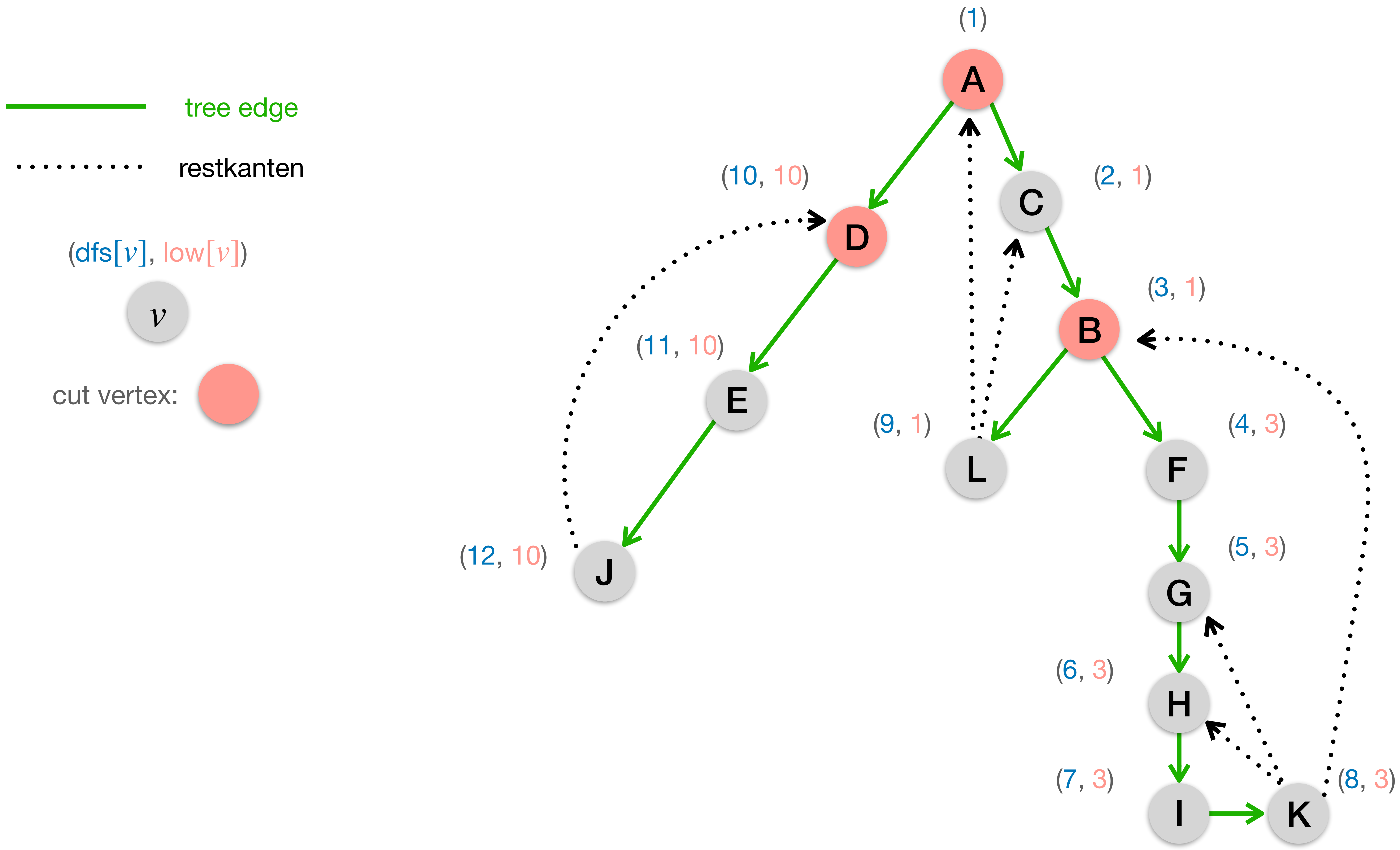
Thus s is a cut vertex.



Identify the cut vertices



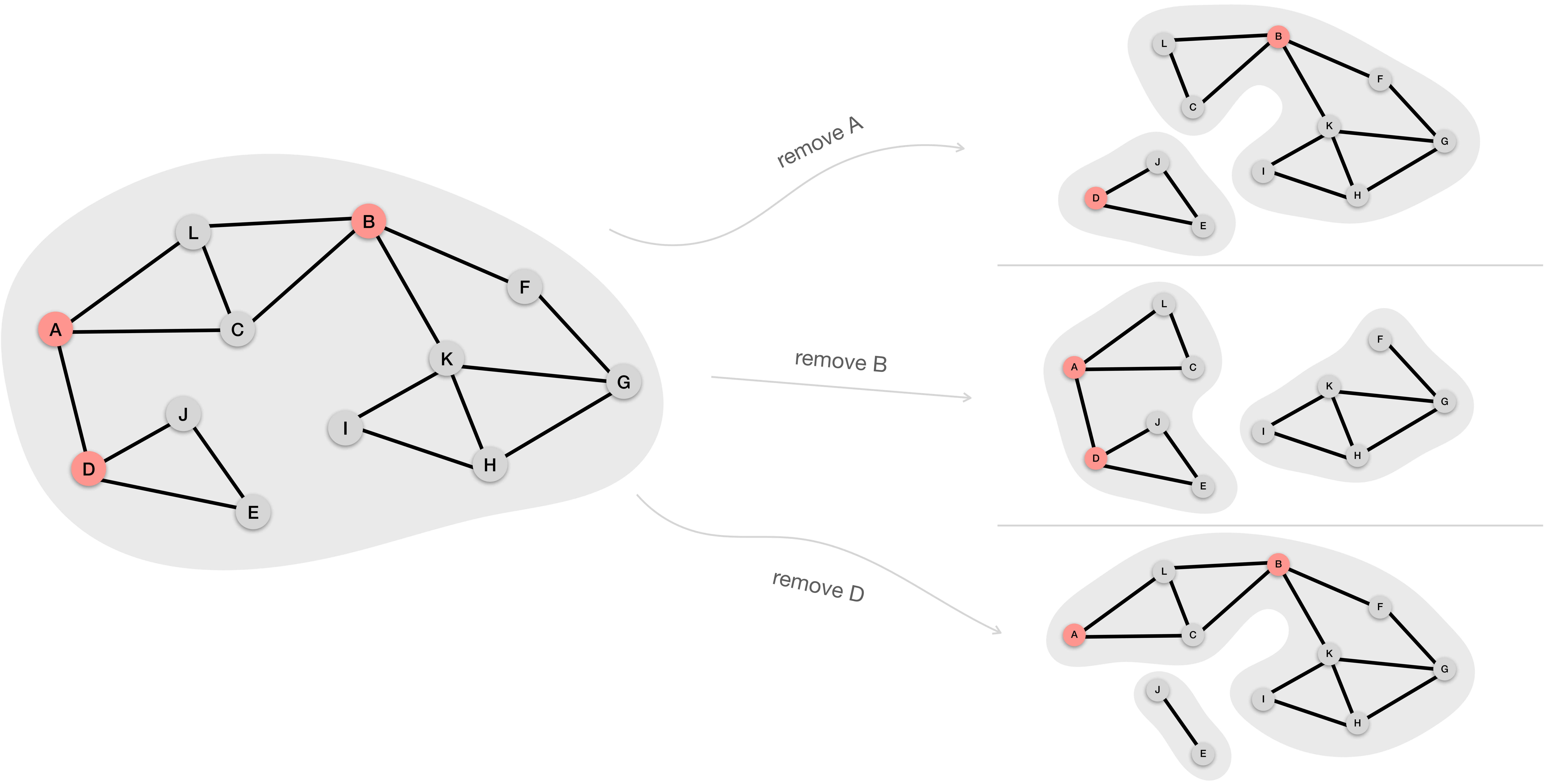
Identify the cut vertices



A since it is the root and $\deg(A) \geq 2$ in the DFS tree.

D since E is a neighbor of D and low$[E] = 10 \geq$ dfs$[D] = 10.$

B since F is a neighbor of B and low$[F] = 3 \geq$ dfs$[D] = 3.$



L
B
A
C
F
K
G
J
I
D
H
E
remove A
remove B
remove D

Pseudocode

```
DFS-Visit(G, v)

num ← num + 1
dfs[v] ← num
low[v] ← dfs[v]
isArtVert[v] ← FALSE
for all {v, w} ∈ E do
    if dfs[w] = 0 then
        T ← T + {v, w}
        val ← DFS-Visit(G, w)
        if val ≥ dfs[v] then
            isArtVert[v] ← TRUE
        low[v] ← min{low[v], val}
    else dfs[w] ≠ 0 and {v, w} ∉ T
        low[v] ← min{low[v], dfs[w]}
return low[v]
```

```
DFS(G, s)

∀v ∈ V :    dfs[v] ← 0
num ← 0
T ← ∅
DFS-Visit(G, s)
if s hat in T Grad mindestens zwei then
    isArtVert[s] ← TRUE
else
    isArtVert[s] ← FALSE
```

Result (cut vertices)

Satz 1.27. Für zusammenhängende Graphen $G = (V, E)$, die mit Adjazenzlisten gespeichert sind, kann man in Zeit $O(|E|)$ alle Artikulationsknoten berechnen.

Note that DFS normally runs in $O(|V| + |E|)$ but since we assume G is connected, we know that $|E| \geq |V| - 1$ thus $|V| + |E| \leq 2 \cdot |E| \leq O(|E|)$.

What about bridges?

- First, notice that if G (connected) contains a bridge $e \in E$, any spanning tree of G must contain e.
- Hence the DFS tree must contain e, as it is a spanning tree of G.
- We reuse our Lemma from earlier:

Lemma: Let $G = (V, E)$ be a connected graph. If $\{u, v\} \in E$ is a bridge, then u and v are cut vertices unless they have degree 1.

What about bridges?

Lemma: Let $G = (V, E)$ be a connected graph. If $\{u, v\} \in E$ is a bridge, then u and v are cut vertices unless they have degree 1.

Let $e = (v, w)$ be an edge in the DFS tree T, then e is a bridge if and only if low$[w] >$ dfs$[v]$.