Graph Stories in Small Area

27th Int. Symposium on Graph Drawing and Network Visualization (GD 2019)

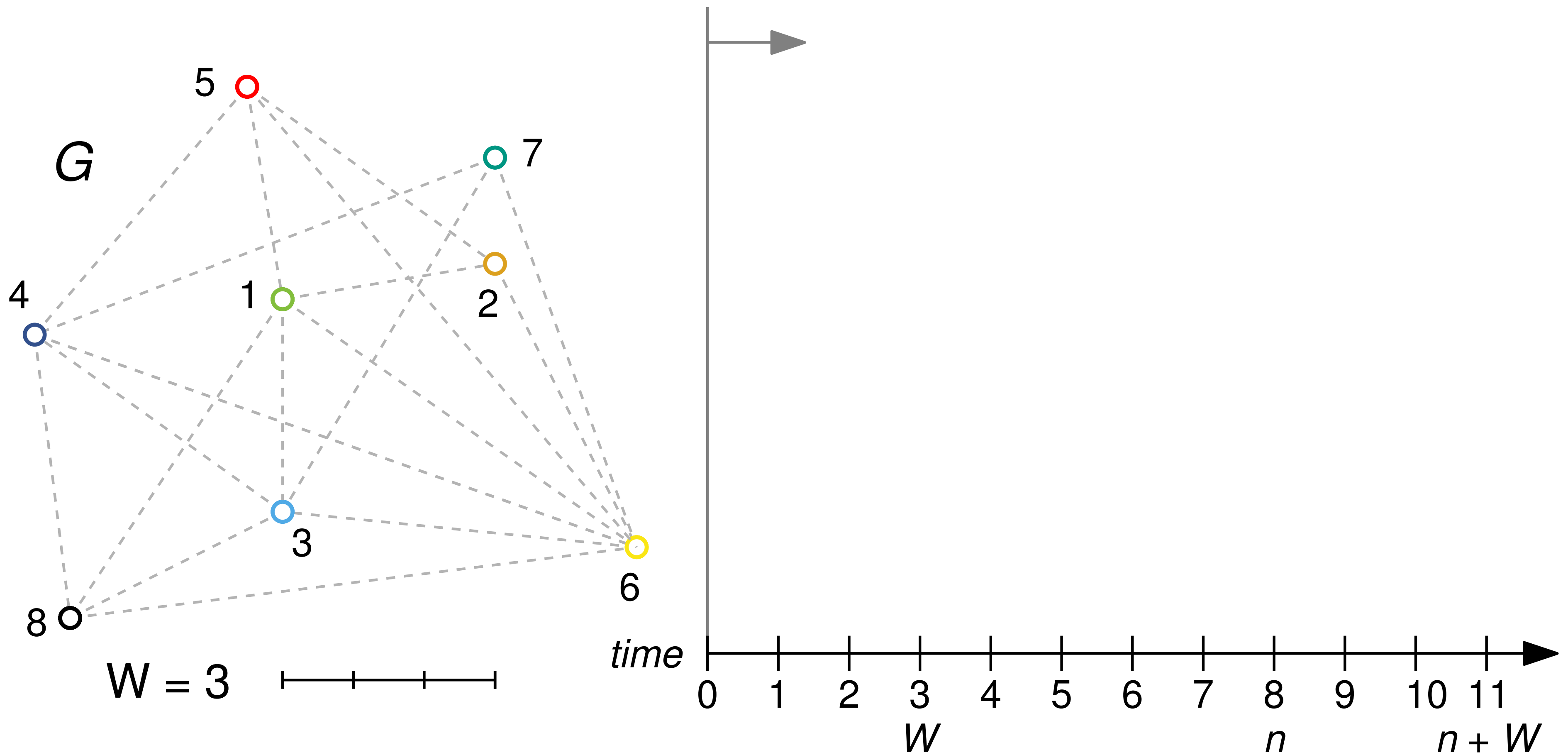
Manuel Borrazzo, **Giordano Da Lozzo**,
Fabrizio Frati, Maurizio Patrignani

DEPARTMENT OF ENGINEERING · ROMA TRE UNIVERSITY

Graph stories

Graphs that change over time:

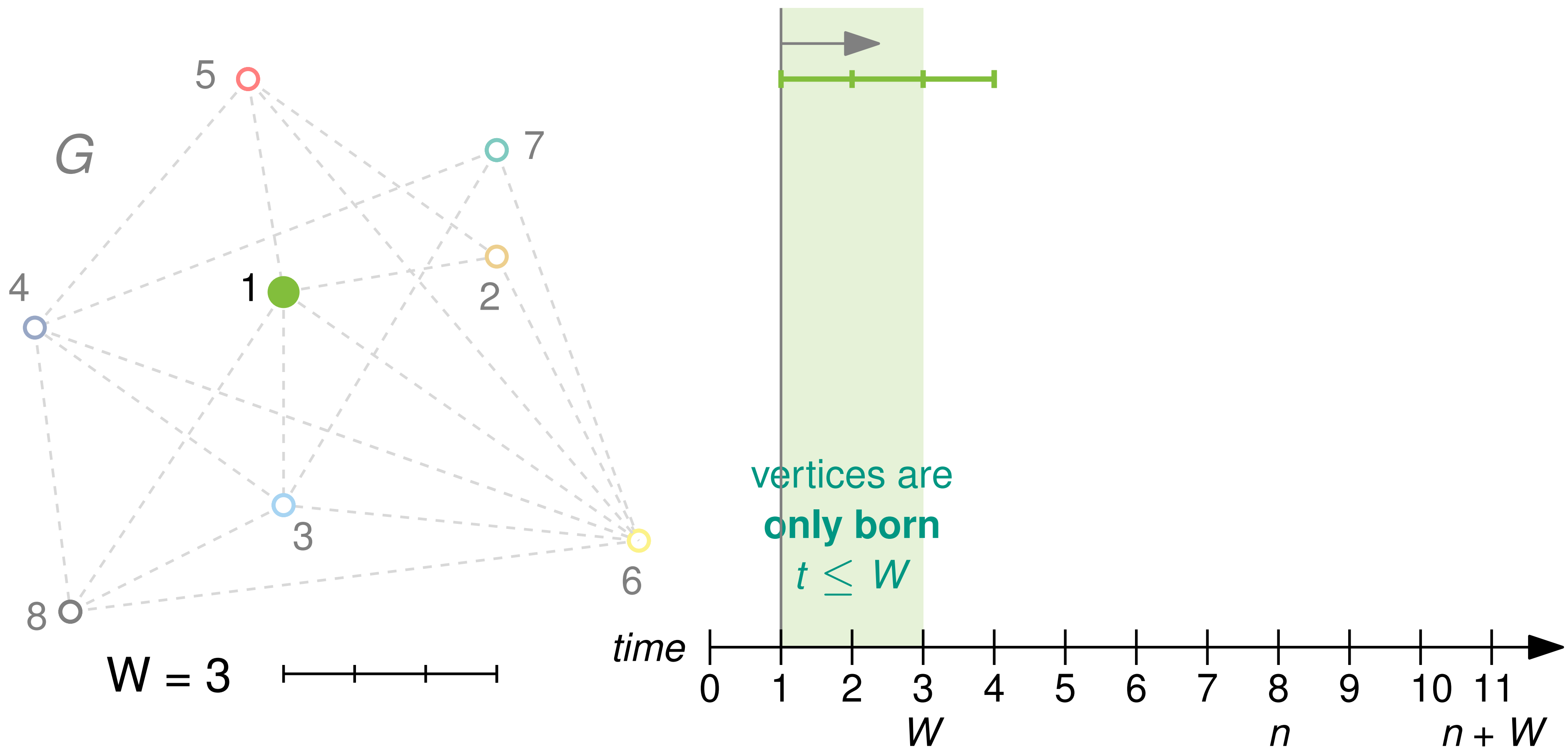
- vertices **enter** the graph one after the other, at discrete time instants
- each **persists** in the graph for a fixed amount of time W (**window size**)



Graph stories

Graphs that change over time:

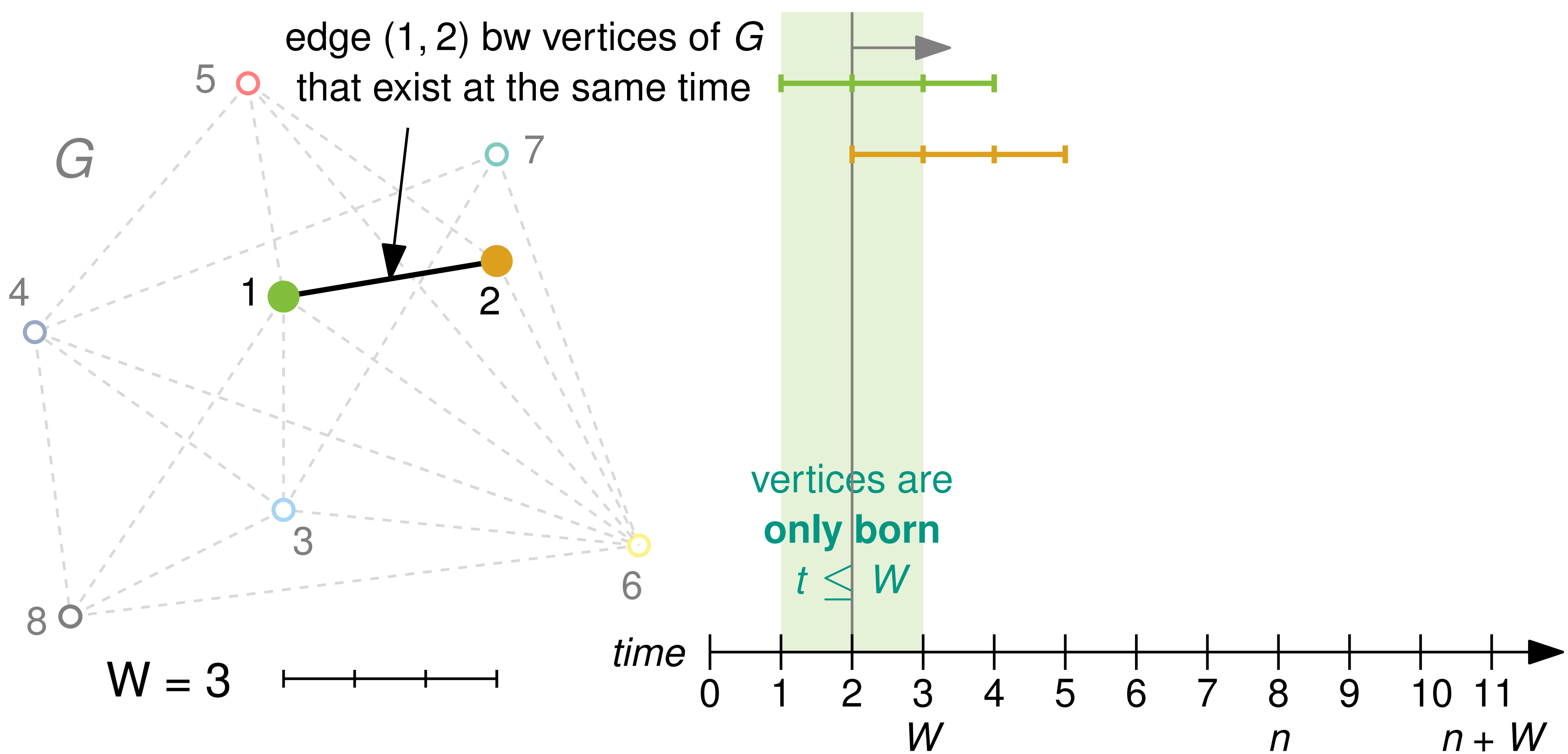
- vertices **enter** the graph one after the other, at discrete time instants
- each **persists** in the graph for a fixed amount of time W (**window size**)



Graph stories

Graphs that change over time:

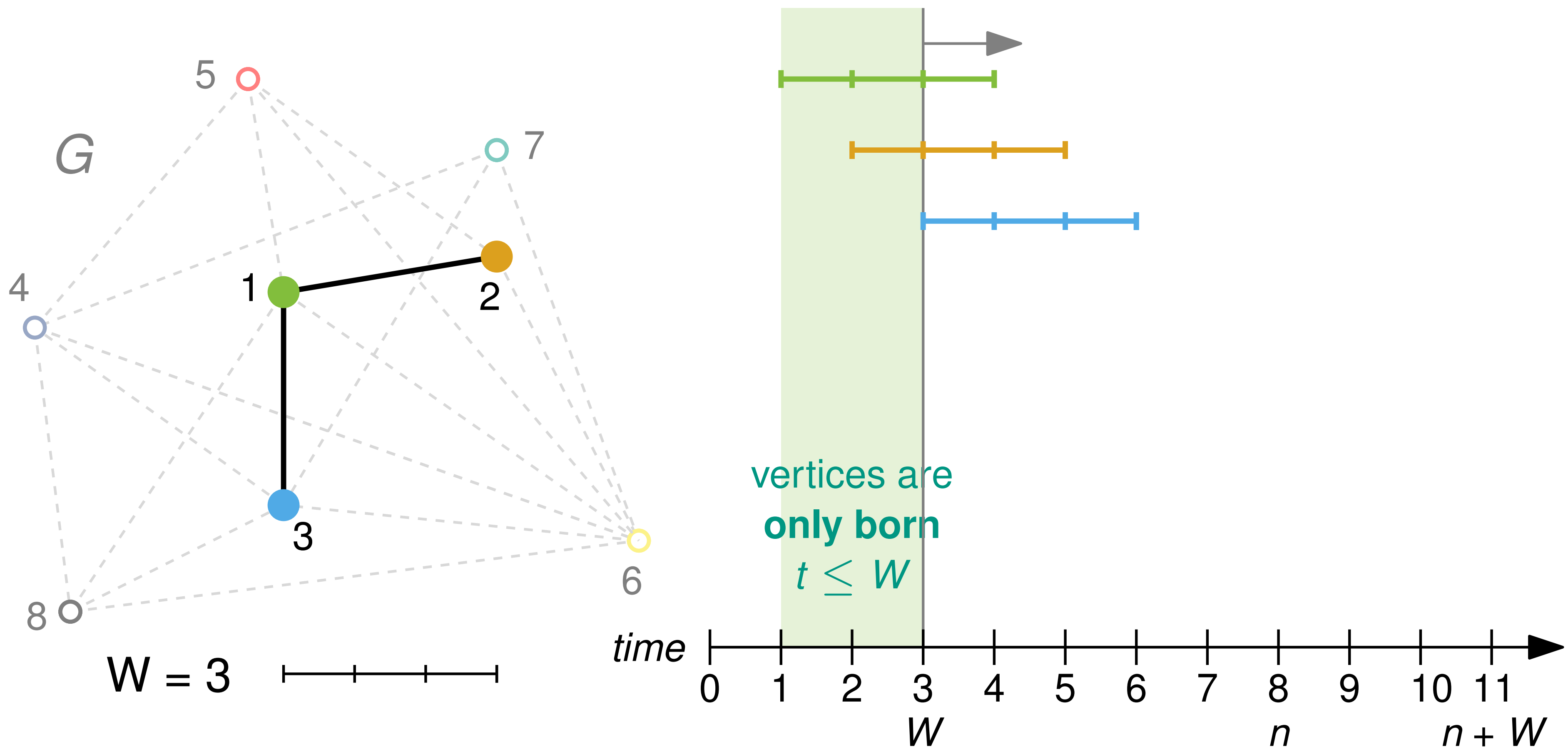
- vertices **enter** the graph one after the other, at discrete time instants
- each **persists** in the graph for a fixed amount of time W (**window size**)



Graph stories

Graphs that change over time:

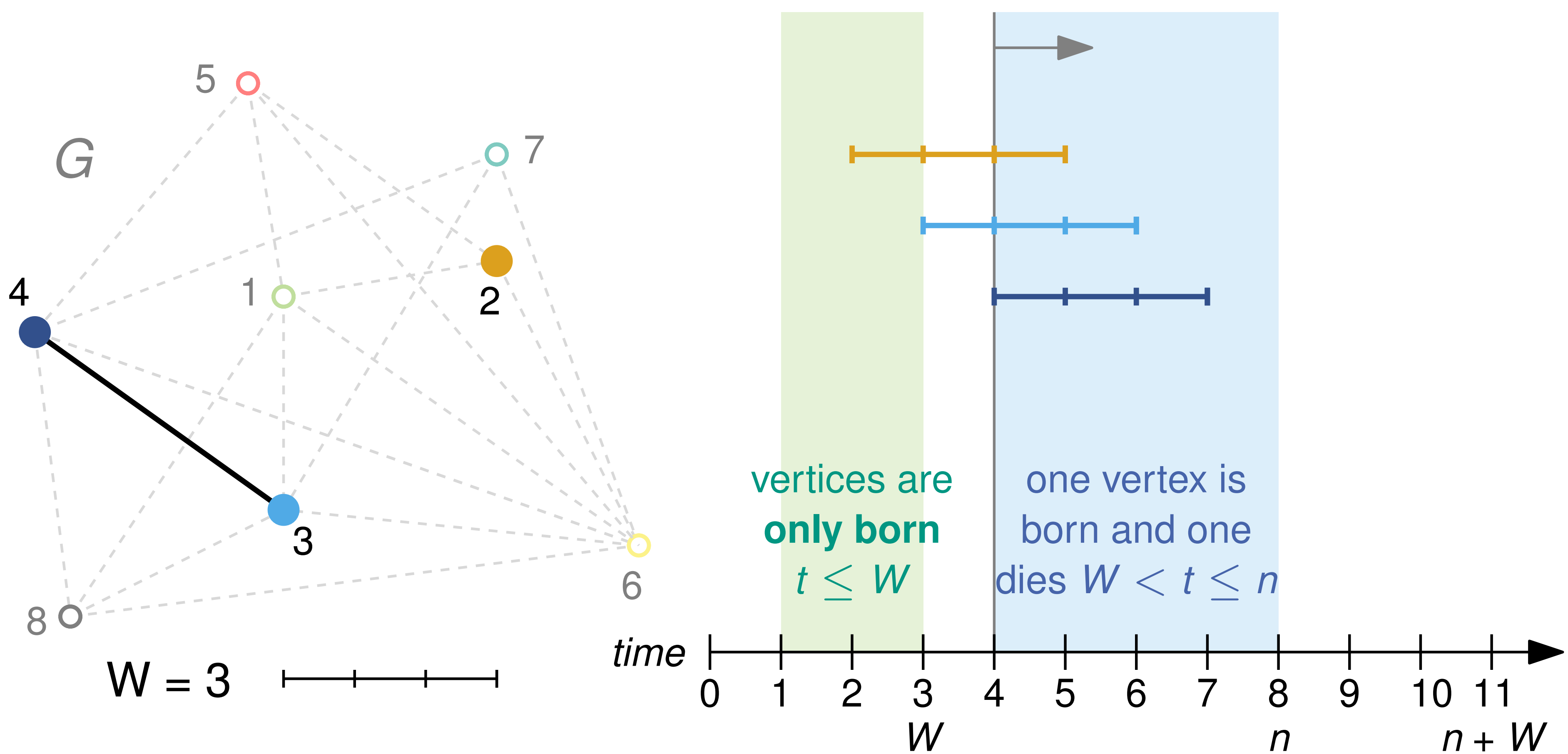
- vertices **enter** the graph one after the other, at discrete time instants
- each **persists** in the graph for a fixed amount of time W (**window size**)



Graph stories

Graphs that change over time:

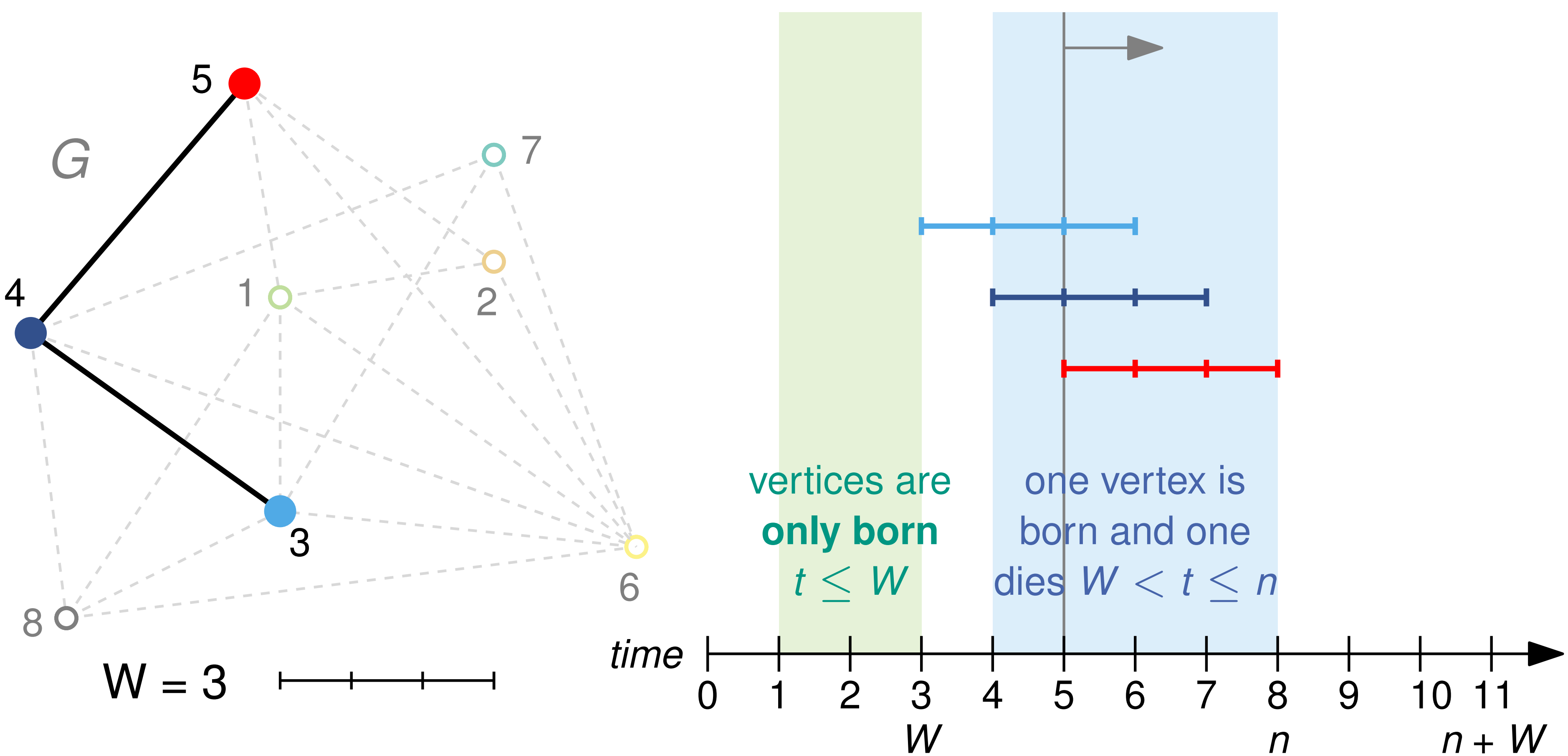
- vertices **enter** the graph one after the other, at discrete time instants
- each **persists** in the graph for a fixed amount of time W (**window size**)



Graph stories

Graphs that change over time:

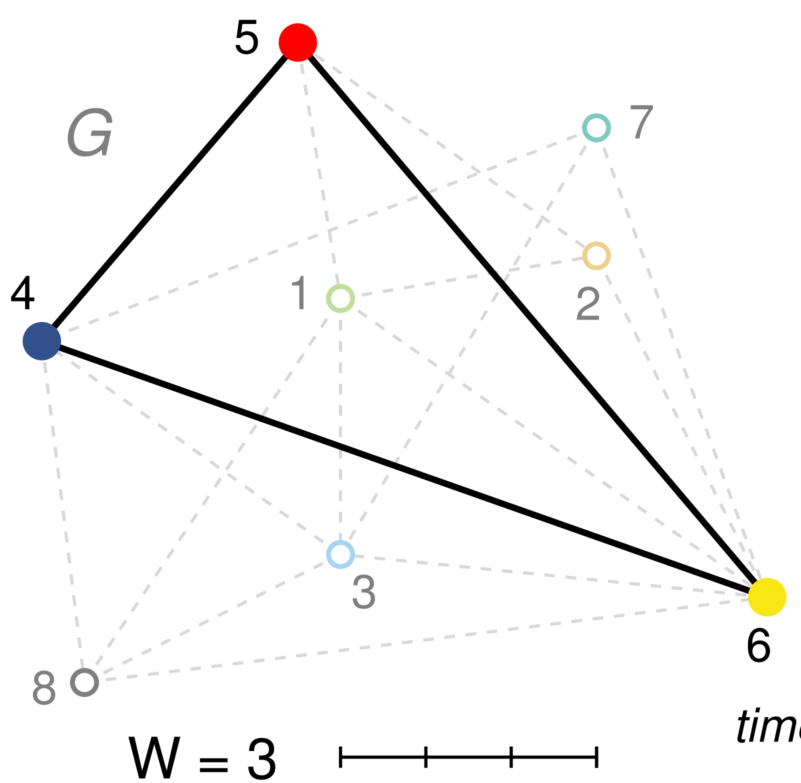
- vertices **enter** the graph one after the other, at discrete time instants
- each **persists** in the graph for a fixed amount of time W (**window size**)

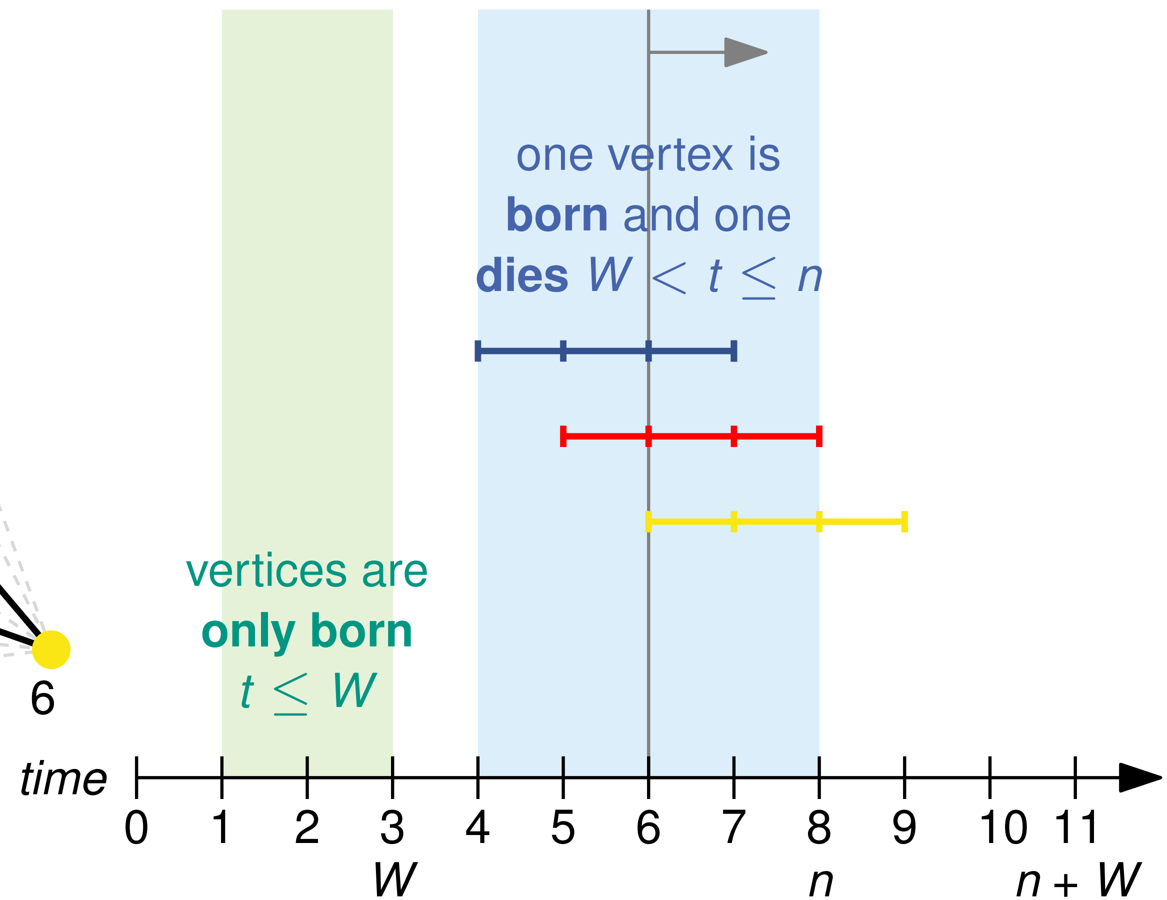


Graph stories

Graphs that change over time:

- vertices **enter** the graph one after the other, at discrete time instants
- each **persists** in the graph for a fixed amount of time W (**window size**)

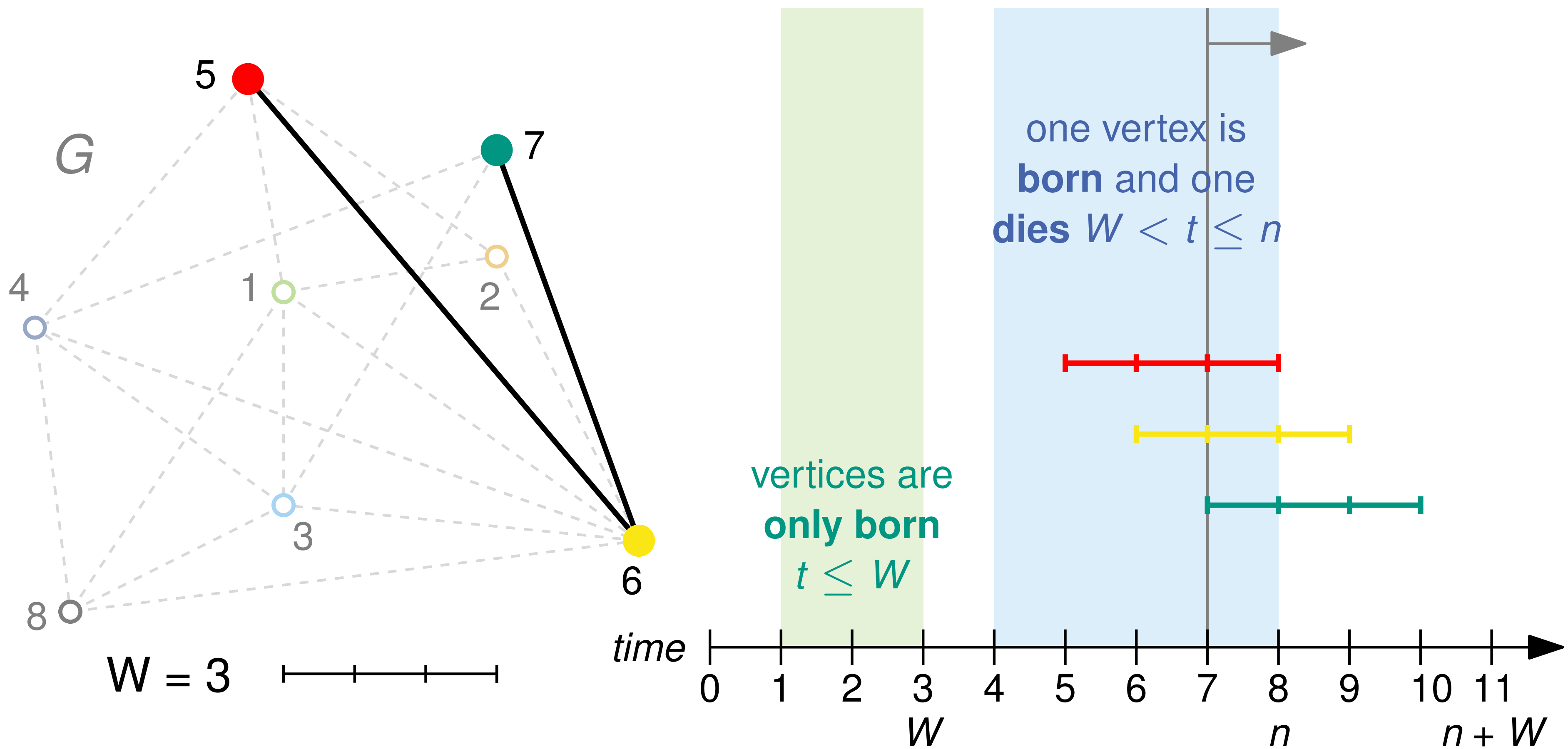




Graph stories

Graphs that change over time:

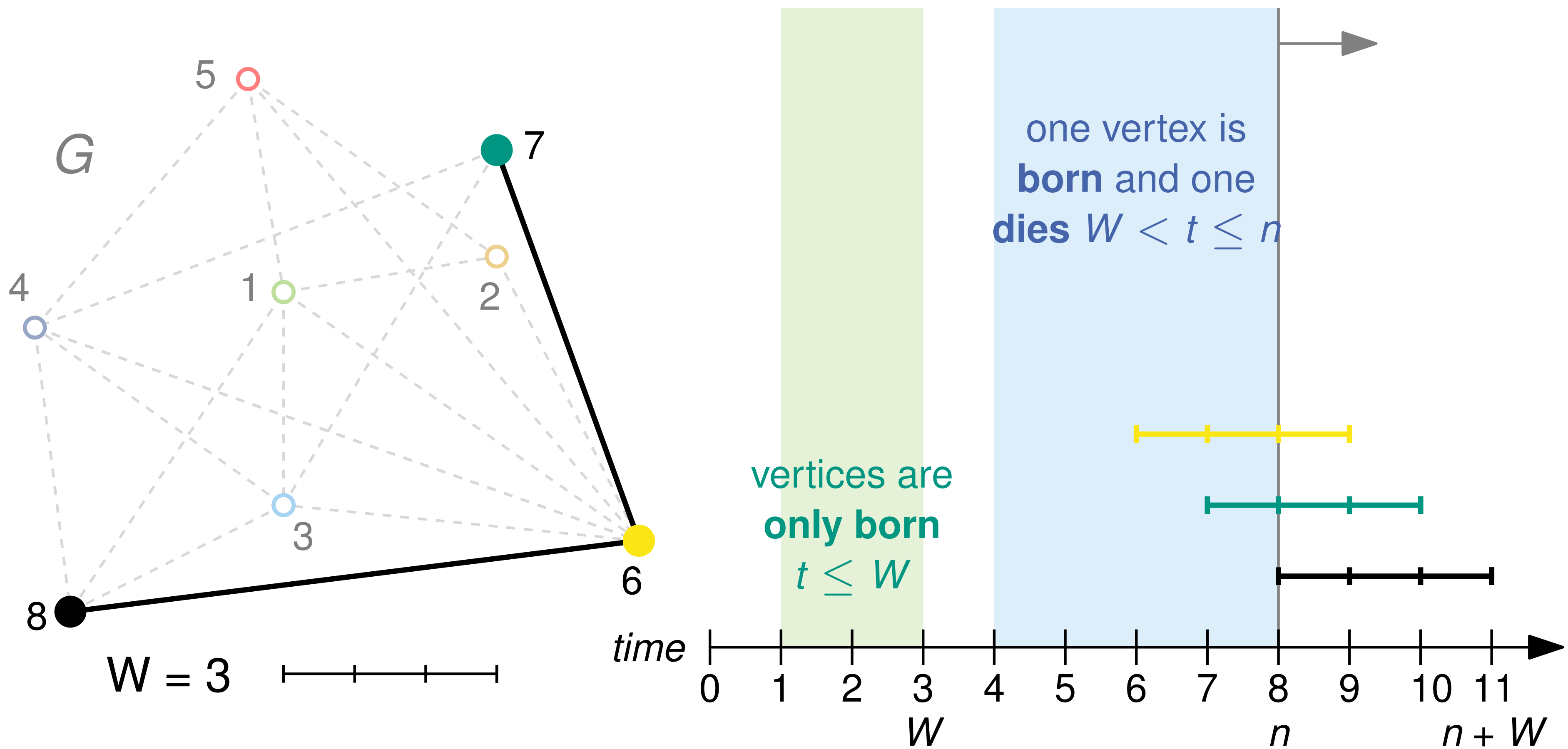
- vertices **enter** the graph one after the other, at discrete time instants
- each **persists** in the graph for a fixed amount of time W (**window size**)



Graph stories

Graphs that change over time:

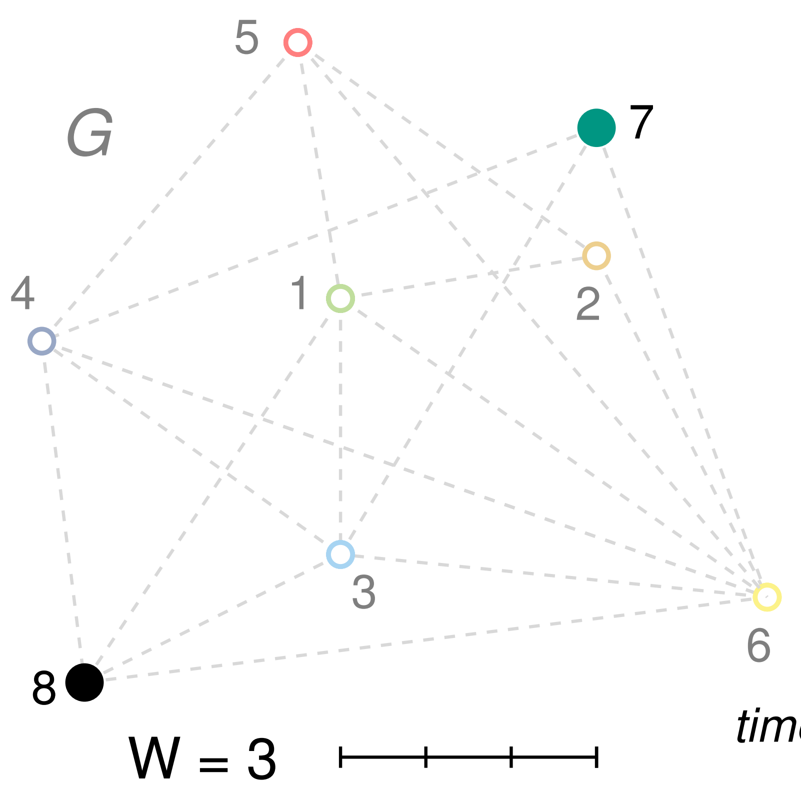
- vertices **enter** the graph one after the other, at discrete time instants
- each **persists** in the graph for a fixed amount of time W (**window size**)

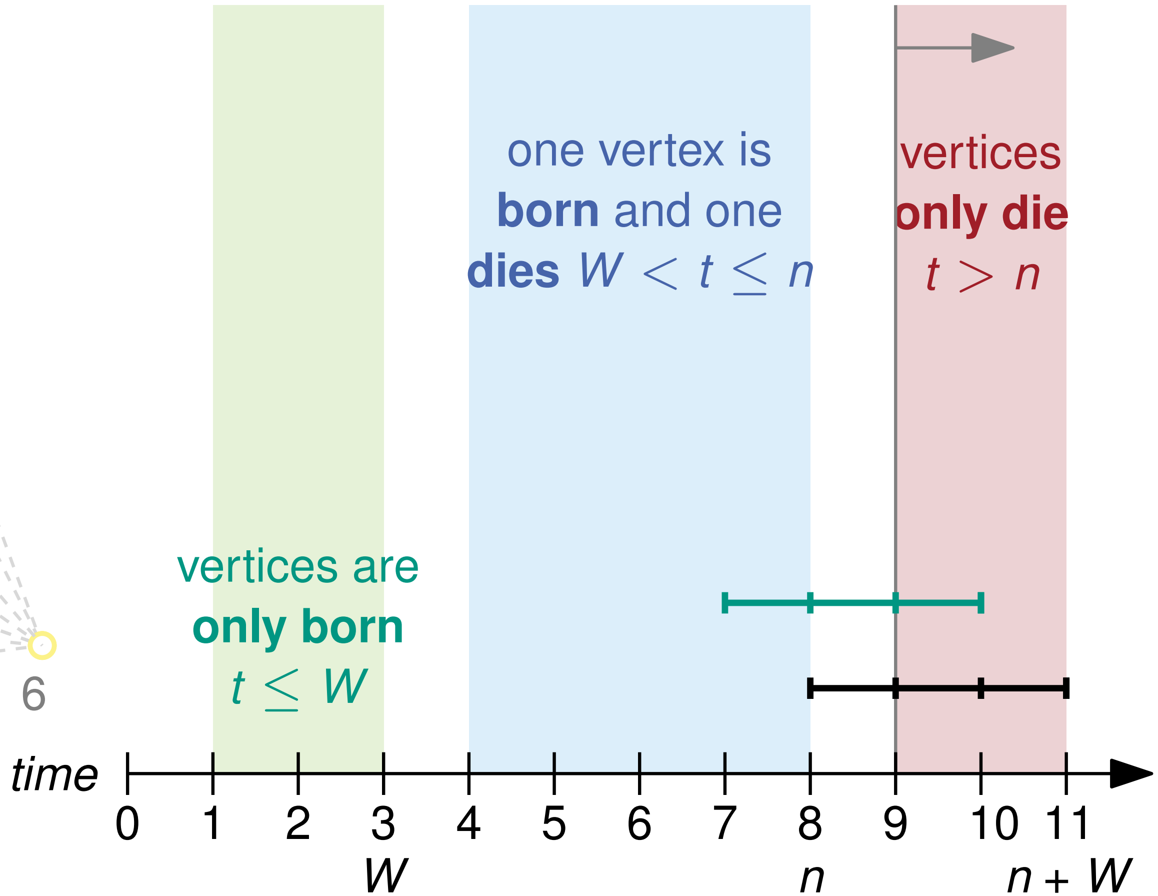


Graph stories

Graphs that change over time:

- vertices **enter** the graph one after the other, at discrete time instants
- each **persists** in the graph for a fixed amount of time W (**window size**)

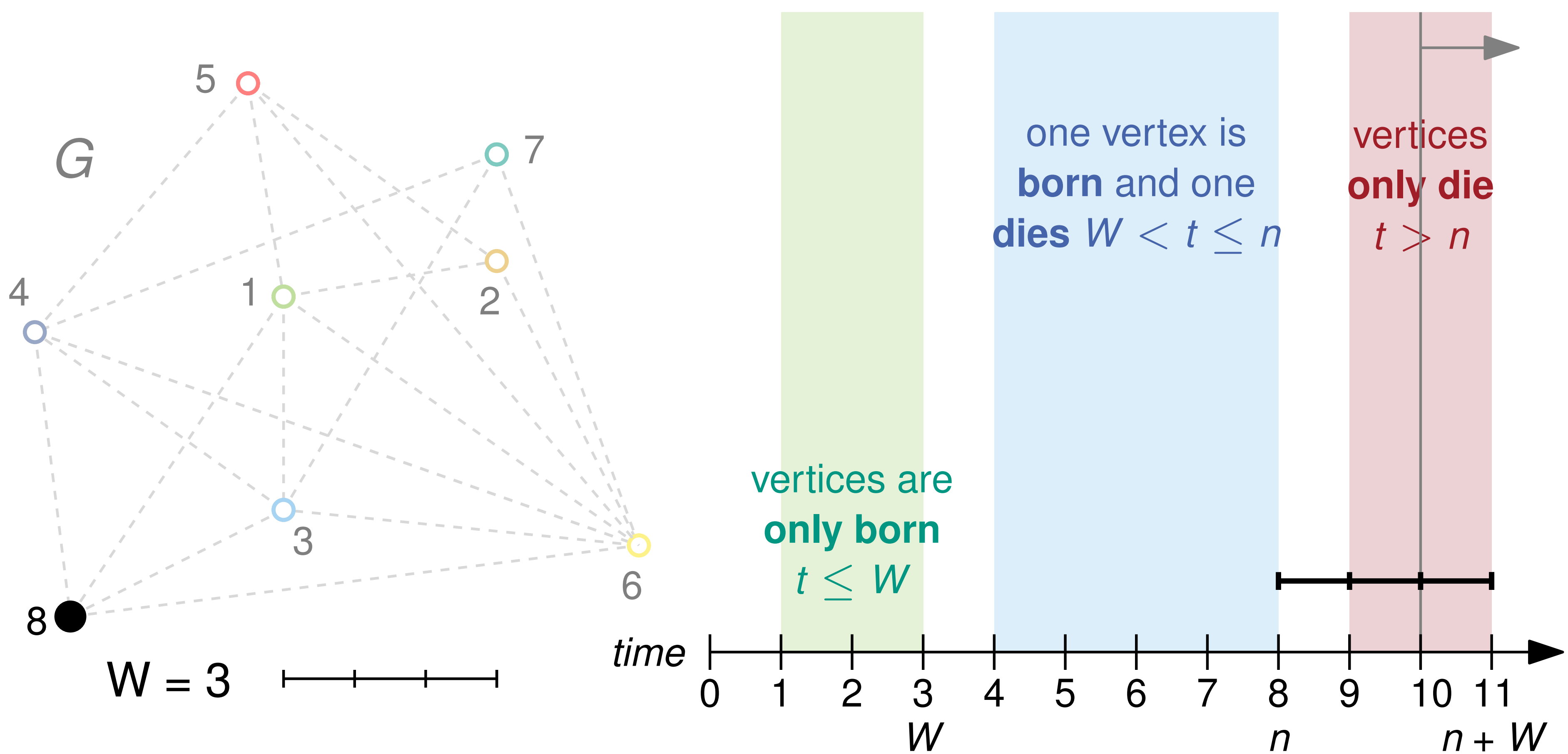




Graph stories

Graphs that change over time:

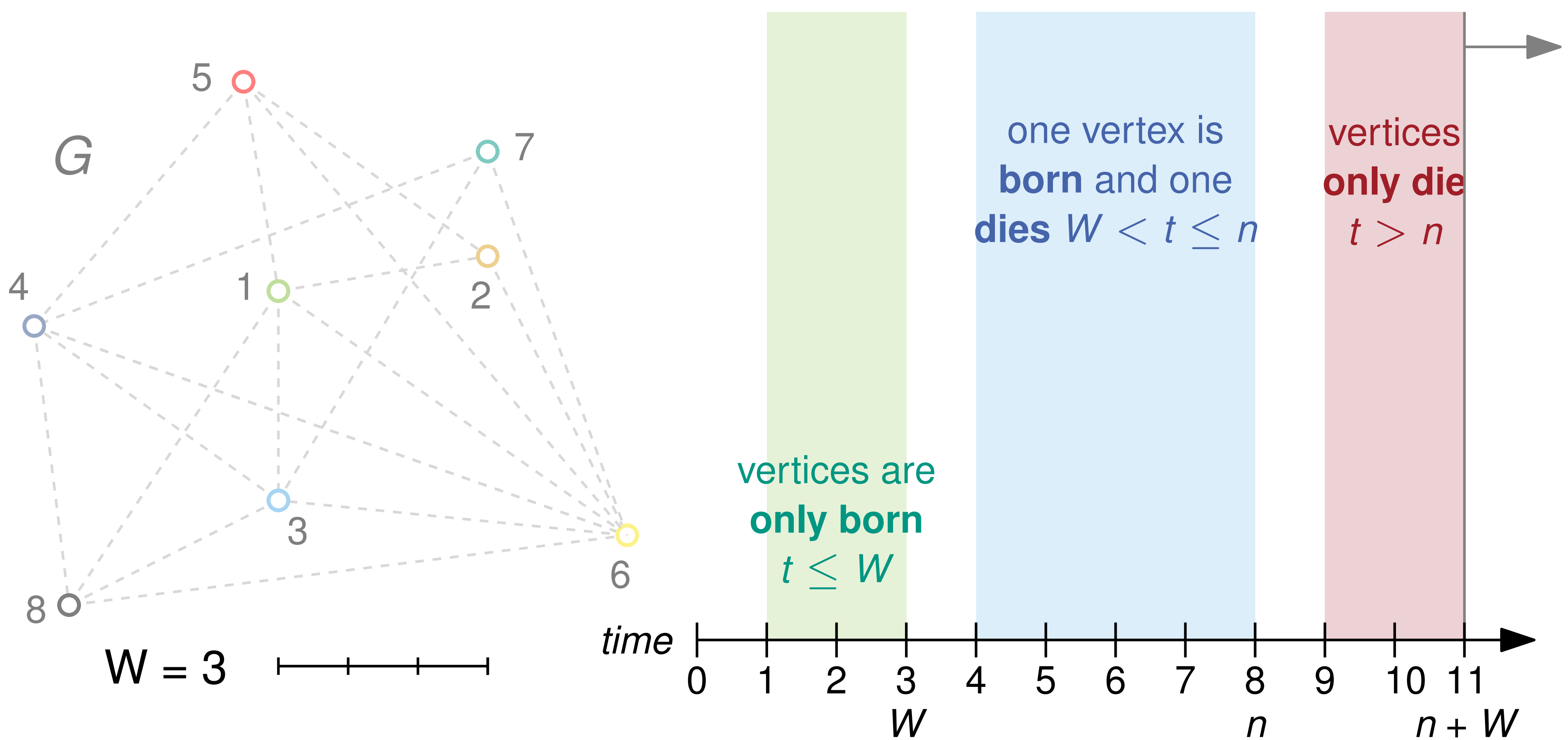
- vertices **enter** the graph one after the other, at discrete time instants
- each **persists** in the graph for a fixed amount of time W (**window size**)



Graph stories

Graphs that change over time:

- vertices **enter** the graph one after the other, at discrete time instants
- each **persists** in the graph for a fixed amount of time W (**window size**)

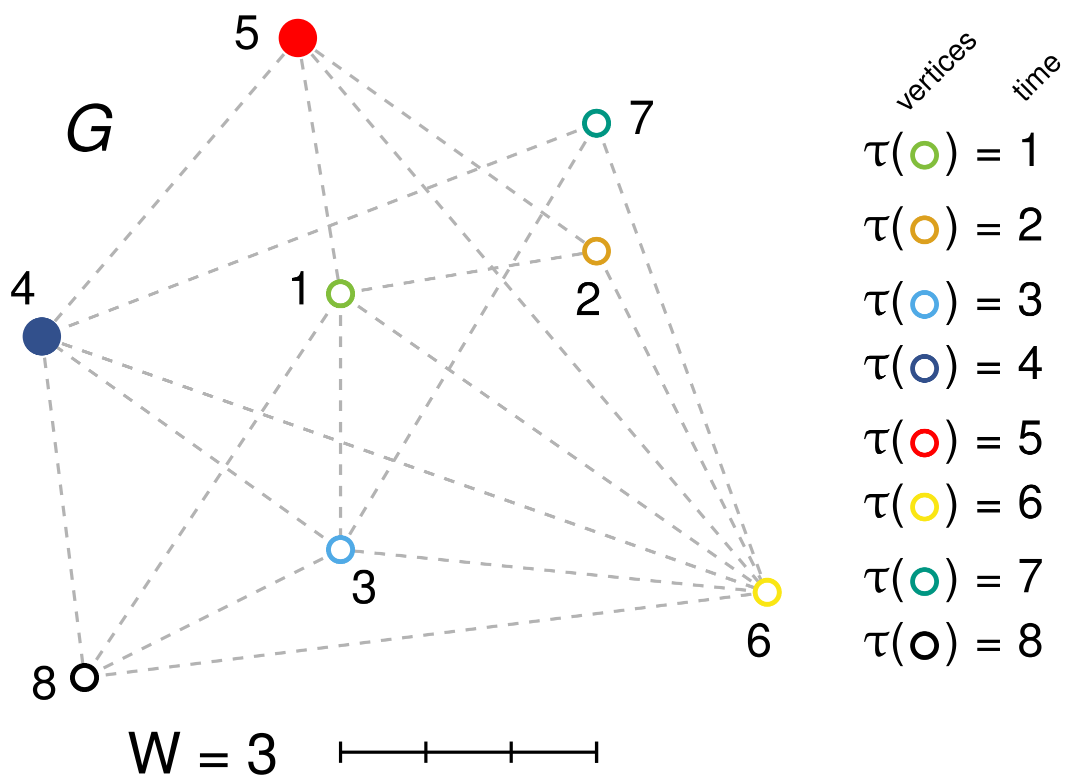


Graph stories

Definition 1

A **graph story** is a triple $\langle G, \tau, W \rangle$

- $G = (V, E)$ is a graph
- a bijection $\tau : V \leftrightarrow \{1, \ldots, |V|\}$ → vertex v **appears** in G at time $\tau(v)$
- W is a positive integer → vertex v **leaves** G at time $\tau(v) + W$

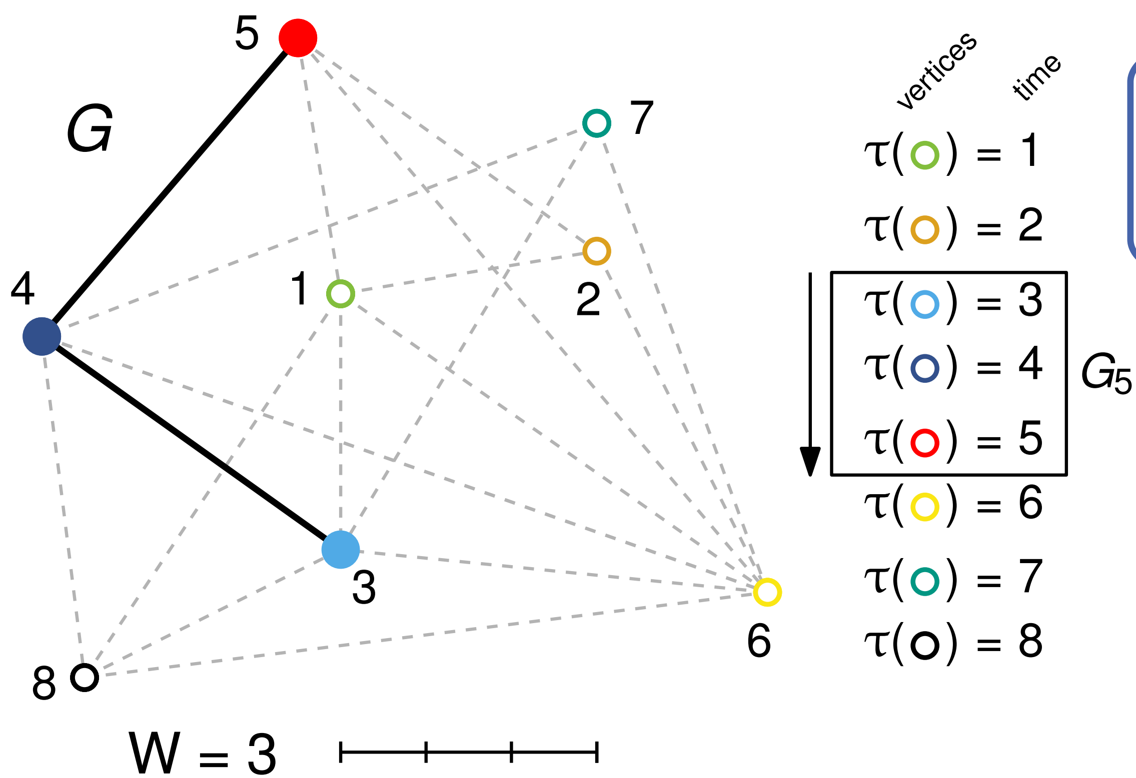


Graph stories

Definition 1

A **graph story** is a triple $\langle G, \tau, W \rangle$

- $G = (V, E)$ is a graph
- a bijection $\tau : V \leftrightarrow \{1, \ldots, |V|\}$ → vertex v **appears** in G at time $\tau(v)$
- W is a positive integer → vertex v **leaves** G at time $\tau(v) + W$



G_t := the subgraph of G **induced by** $V_t = \{v \in V : t - W < \tau(v) \leq t\}$

↓

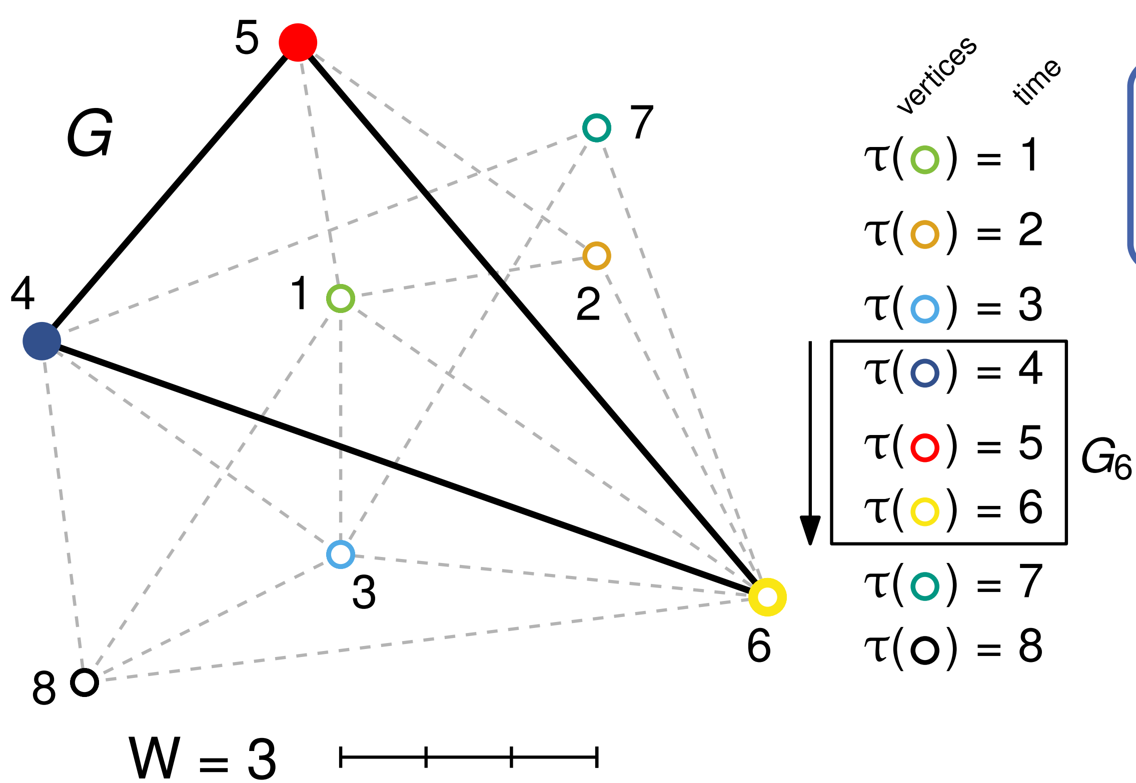
fixed lifetime $\simeq$ sliding window

Graph stories

Definition 1

A **graph story** is a triple $\langle G, \tau, W \rangle$

- $G = (V, E)$ is a graph
- a bijection $\tau : V \leftrightarrow \{1, \ldots, |V|\}$ → vertex v **appears** in G at time $\tau(v)$
- W is a positive integer → vertex v **leaves** G at time $\tau(v) + W$



G_t := the subgraph of G **induced by** $V_t = \{v \in V : t - W < \tau(v) \leq t\}$

fixed lifetime $\simeq$ sliding window

Graph stories

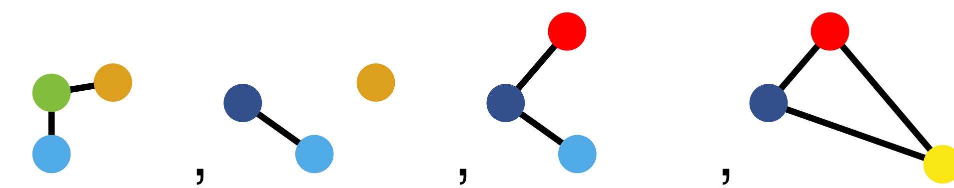
Definition 1

A **graph story** is a triple $\langle G, \tau, W \rangle$

- $G = (V, E)$ is a graph
- a bijection $\tau : V \leftrightarrow \{1, \ldots, |V|\}$ → vertex v **appears** in G at time $\tau(v)$
- W is a positive integer → vertex v **leaves** G at time $\tau(v) + W$

$\Downarrow$

sequence = $\langle \cdots\cdots, G_3, G_4, G_5, G_6, \cdots\cdots \rangle$



$$\langle \underbrace{G_1, G_2, \ldots, G_{W-1}}_{size\, <\, W}, \underbrace{G_W, \ldots, G_n}_{size\, =\, W}, \underbrace{G_{n+1}, \ldots, G_{n+W-1}}_{size\, <\, W} \rangle$$

Drawing stories (of graph stories)

Definition 2

A **drawing story** for $\langle G, \tau, W \rangle$ is a sequence $\Gamma = \langle \Gamma_1, \Gamma_2, \ldots, \Gamma_{n+W-1} \rangle$ such that:

- Γ_t is a drawing of G_t
- if $v \in V(G_i) \cap V(G_j)$, then v is in the **same position** in Γ_i and in Γ_j
- if $e \in E(G_i) \cap E(G_j)$, then e is represented by the **same curve** in Γ_i and in Γ_j

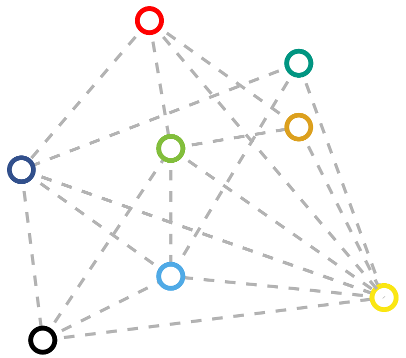
Benefit: preserve the user's mental map through the sequence

i.e., $\Gamma = \langle \Gamma_1, \Gamma_2, \ldots, \Gamma_{n+W-1} \rangle$ is a **SEFE** of $\langle G_1, G_2, \ldots, G_{n+W-1} \rangle$

Drawing stories (of graph stories)

Our focus: drawings stories that are **planar**, **straight-line** ($\rightarrow$ **SGE**), and **on the grid**

A graph story $\langle G, \tau, W \rangle$ **may admit** such drawing stories even if G is **not planar**

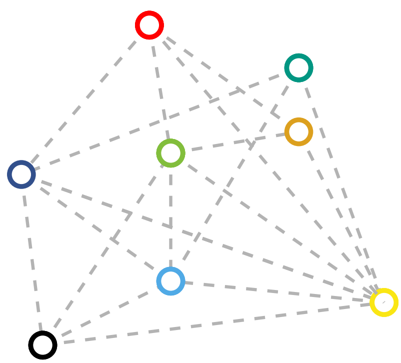


For instance, if **W=3** just place the vertices in **general positions on a grid**

Drawing stories (of graph stories)

Our focus: drawings stories that are **planar**, **straight-line** ($\rightarrow$ **SGE**), and **on the grid**

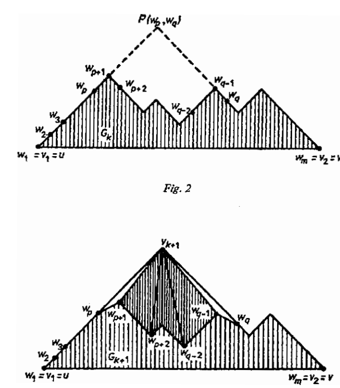
A graph story $\langle G, \tau, W \rangle$ **may admit** such drawing stories even if G is **not planar**

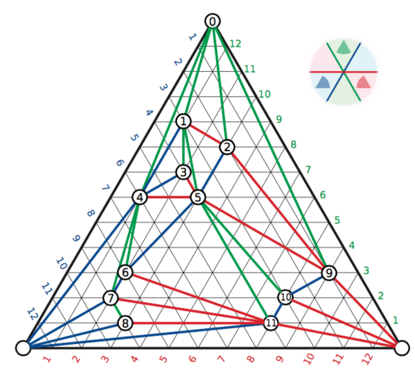


For instance, if **W=3** just place the vertices in **general positions on a grid**

A graph story $\langle G, \tau, W \rangle$ **always admits** such drawing stories if G is **planar**

A naive approach would produce drawing stories on the $O(n) \times O(n)$ grid (de Fraysseix, Pach and Pollack, Schnyder, . . .)





this may result in **unnecessarily large drawings**

Drawing stories in small area

We studied (straight-line planar grid) drawing stories of **planar graph stories** with the **goal** of producing drawing stories $\Gamma = \langle \Gamma_1, \Gamma_2, \ldots, \Gamma_{n+W-1} \rangle$ such that each $\Gamma_t \in \Gamma$ has an **area** that is a function of W, not of n

Drawing stories in small area

We studied (straight-line planar grid) drawing stories of **planar graph stories** with the **goal** of producing drawing stories $\Gamma = \langle \Gamma_1, \Gamma_2, \ldots, \Gamma_{n+W-1} \rangle$ such that each $\Gamma_t \in \Gamma$ has an **area** that is a function of W, not of n

Theorem 1 [Planar Graph Stories]

There are n-vertex **planar graph stories** whose every drawing story lies on an $\Omega(n) \times \Omega(n)$ **grid**

Theorem 2 [Path Stories]

Any n-vertex **path story** $\langle P, \tau, W \rangle$ admits a drawing story that lies on a $2W \times 2W$ **grid**, which is computable in $O(n)$ time

Theorem 3 [Tree Stories]

Any n-vertex **tree story** $\langle T, \tau, W \rangle$ admits a drawing story that lies on an $(8W + 1) \times (8W + 1)$ **grid**, which is computable in $O(n)$ time

Tree stories

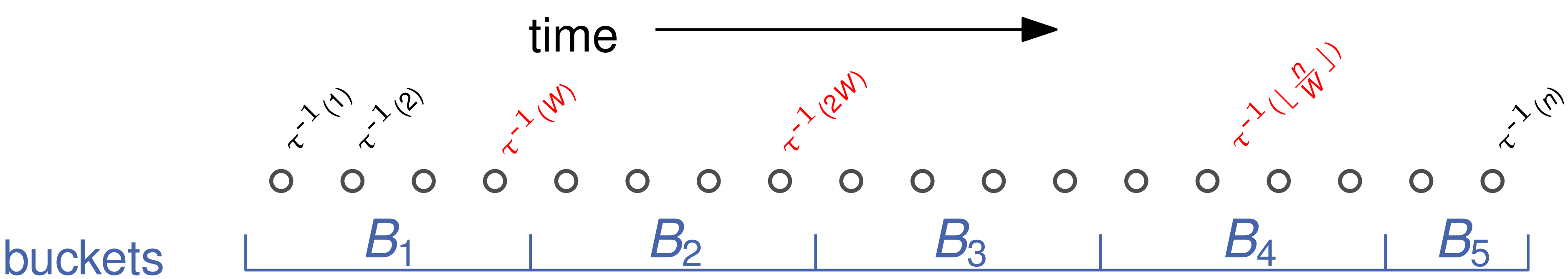
Dendrological Gardens, Průhonice

Tree stories: bucketing

Theorem 3 [Tree Stories]

Any n-vertex **tree story** $\langle T, \tau, W\rangle$ admits a drawing story that lies on an **(8W +1)** $\times$ **(8W +1) grid**, which is computable in $O(n)$ time

Key idea: vertices are assigned to **buckets** according to τ



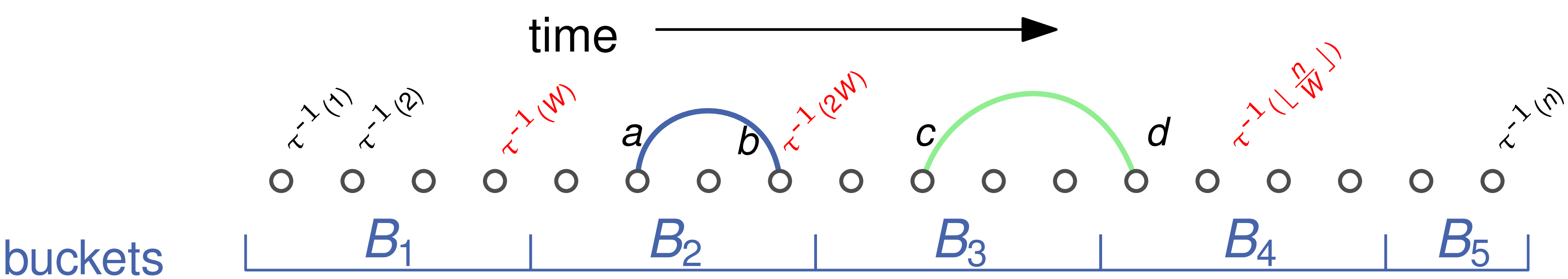
- $v \in B_i$ iff $i = \lfloor \frac{\tau(v)}{W} \rfloor$

Tree stories: bucketing

Theorem 3 [Tree Stories]

Any n-vertex **tree story** $\langle T, \tau, W \rangle$ admits a drawing story that lies on an **(8W +1)** $\times$ **(8W +1) grid**, which is computable in $O(n)$ time

Key idea: vertices are assigned to **buckets** according to τ



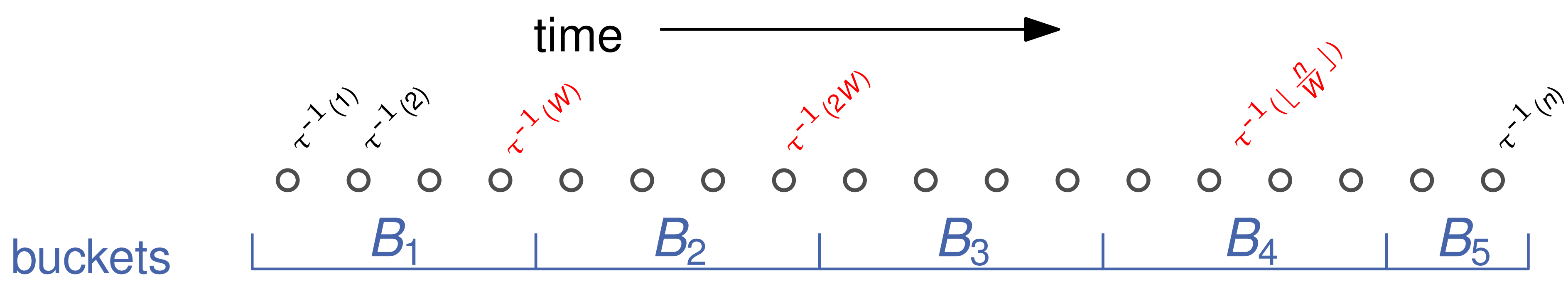
- $v \in B_i$ iff $i = \lfloor \frac{\tau(v)}{W} \rfloor$
- intra-bucket edges: $a, b \in B_i$
- inter-bucket edges: $c \in B_i$, $d \in B_j$, with $i \neq j$

Tree stories: bucketing

Theorem 3 [Tree Stories]

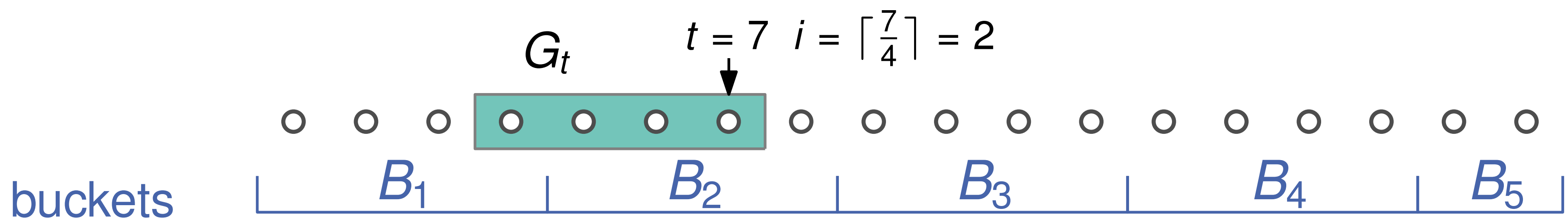
Any n-vertex **tree story** $\langle T, \tau, W \rangle$ admits a drawing story that lies on an **(8W +1) $\times$ (8W +1) grid**, which is computable in $O(n)$ time

Key idea: vertices are assigned to **buckets** according to τ



Property 1

For any $t = 1, 2, \ldots, n + W - 1$, let $i = \lceil \frac{t}{W} \rceil$. Then G_t is a subgraph of $T[B_{i-1} \cup B_i]$.



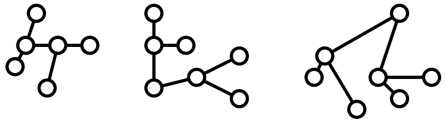
Tree stories: strategy

Theorem 3 [Tree Stories]

Any n-vertex **tree story** $\langle T, \tau, W \rangle$ admits a drawing story that lies on an **$(8W+1) \times (8W+1)$ grid**, which is computable in $O(n)$ time

High-level Strategy:

- partition $V(T)$ into buckets $B_1, \ldots, B_{\lceil \frac{n}{W} \rceil}$



$T[B_i]$ is a forest

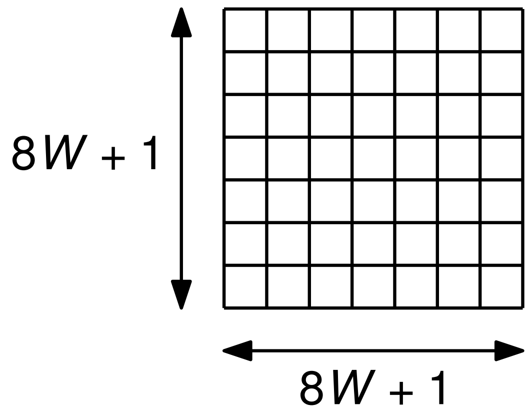
- draw T s.t. forests $T[B_i \cup B_{i+1}]$ are straight-line planar on an $(8W+1) \times (8W+1)$ grid

Property 1

G_t is a subgraph of $T[B_{i-1} \cup B_i]$, with $i = \lceil \frac{t}{W} \rceil$ $\square$

Tree Stories: requirements

Requirement 1: For $n >> W$, we need a strategy for **reusing space**



can only fit $O(W^2)$ vertices

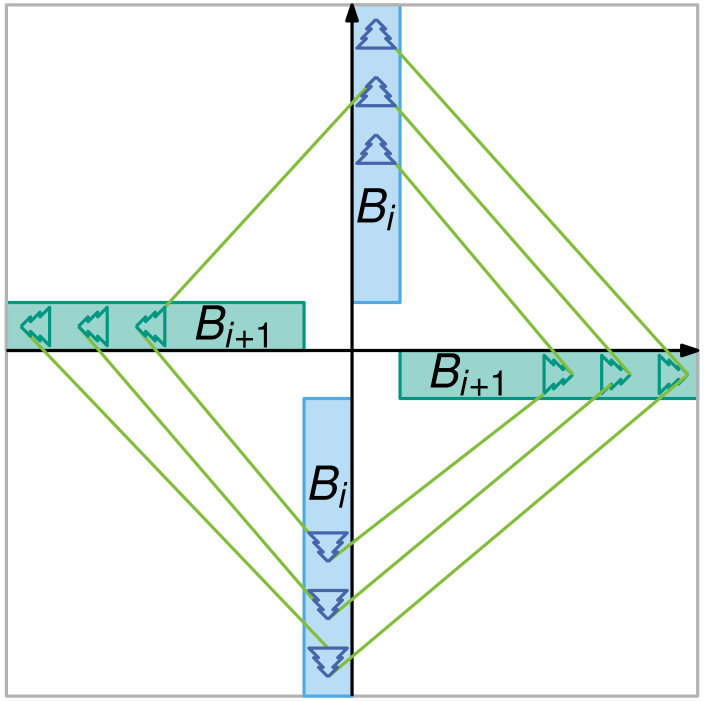
Requirement 2: Since we construct the drawing story for $\langle G, \tau, W\rangle$ by drawing the forests $T[B_1 \cup B_2], T[B_2 \cup B_3], \ldots, T[B_{\lceil \frac{n}{W} \rceil - 1} \cup B_{\lceil \frac{n}{W} \rceil}]$ independently, we need these drawings to **coincide** on their shared vertices

- i.e., the drawings for $T[B_i \cup B_{i+1}]$ and $T[B_{i+1} \cup B_{i+2}]$ should induce the same drawing of $T[B_{i+1}]$

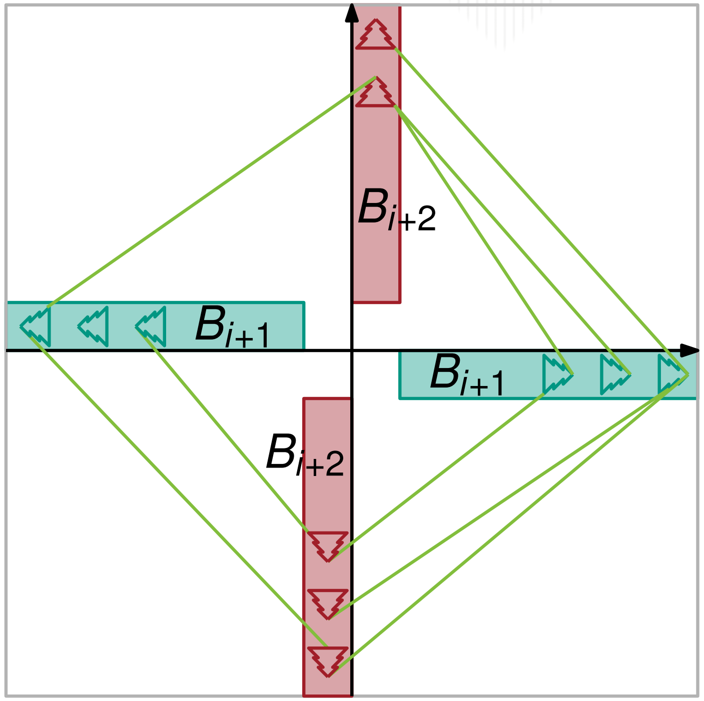
Tree Stories

A childish idea: *Give me a pinwheel and I will draw your trees!!*

Drawing Γ' of $T[B_i \cup B_{i+1}]$



Drawing Γ'' of $T[B_{i+1} \cup B_{i+2}]$



How to meet the requirements:

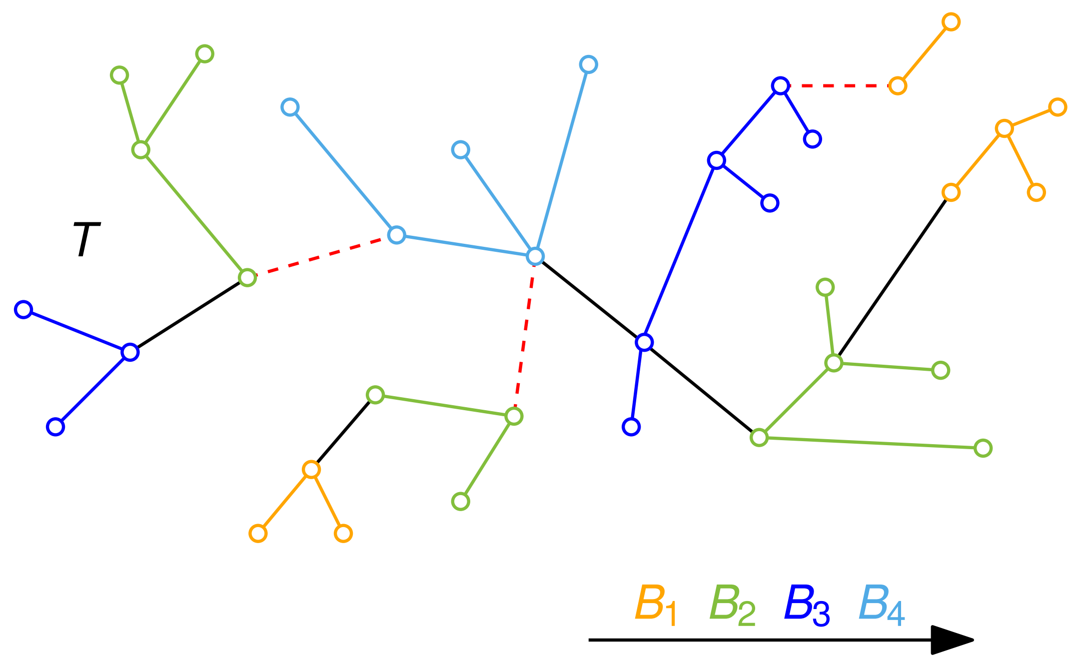
- we construct **special drawings** of the forests $T[B_i \cup B_{i+1}]$'s that **wrap around** the origin
- **all trees of** $T[B_j]$ are drawn **close to one of the axis**
- trees from $T[B_j]$ and $T[B_{j+2}]$ will **reuse the same drawing space**
- the drawing of the $T[B_j]$'s must provide **strong visibility properties** for inter-bucket edges

Algorithm's Phases 1/5

Phase 1: a) assign each vertex to the bucket B_i it belongs to

b) remove from T edges between **non-consecutive buckets** (*never visualized*)

$\langle T, \tau, W \rangle$



$u \in B_j \qquad v \in B_j$

$|j - 1| > 1 \Rightarrow |\tau(u) - \tau(v)| > W$

$u \in B_j \qquad v \in B_j$

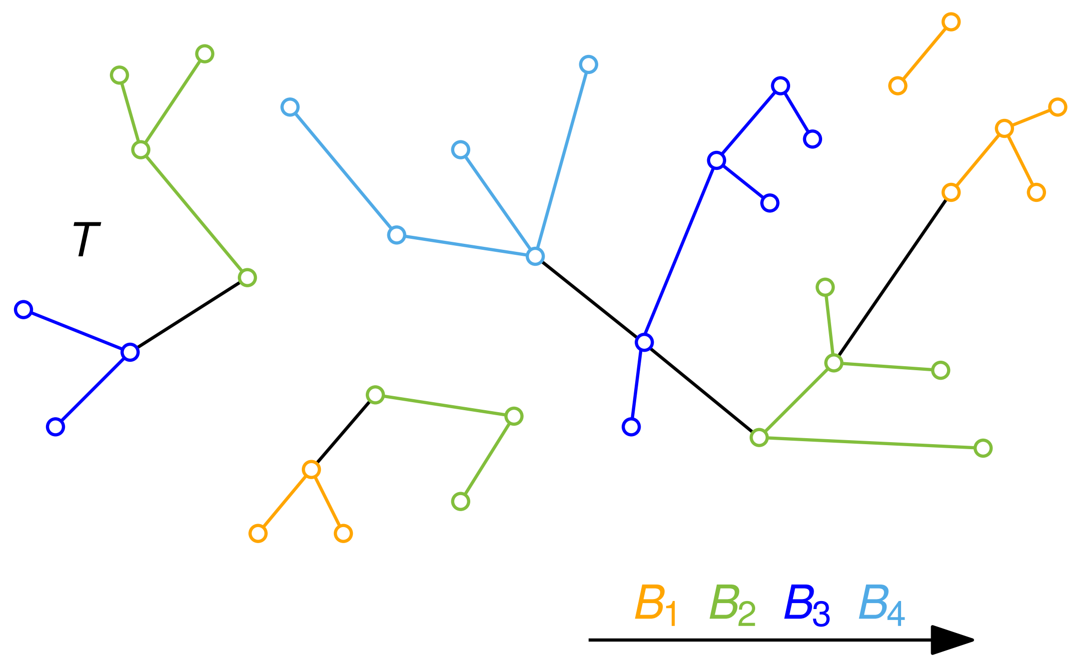
$|j - 1| = 1$

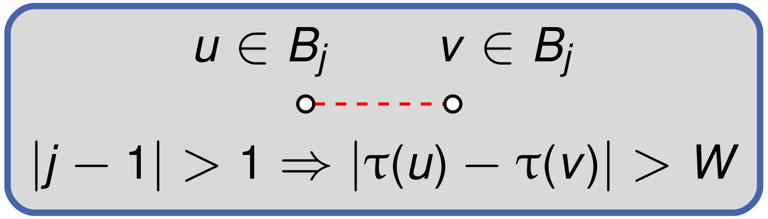
Algorithm's Phases 1/5

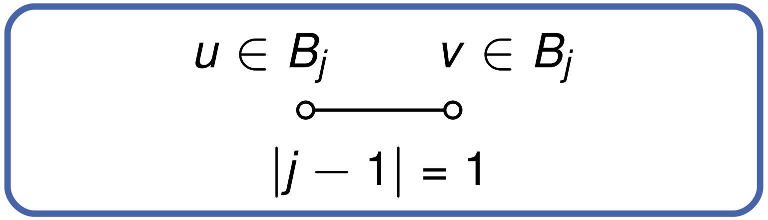
Phase 1: a) assign each vertex to the bucket B_i it belongs to

b) remove from T edges between **non-consecutive buckets** (*never visualized*)

$\langle T, \tau, W \rangle$



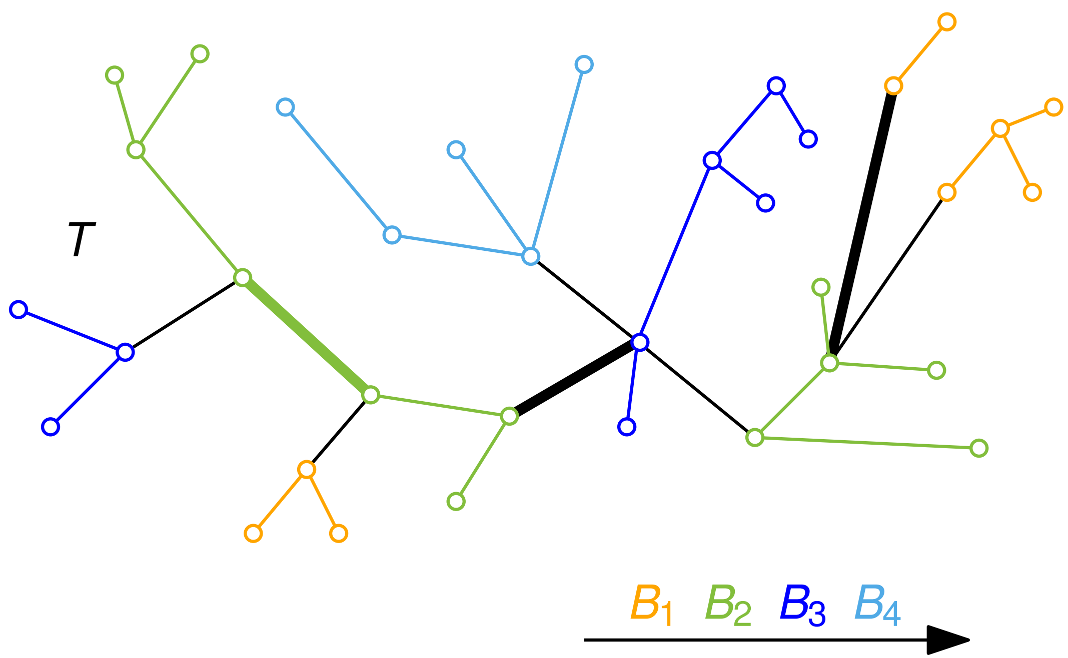




Algorithm's Phases 2/5

Phase 2: - add **inter- or intra-bucket edges** to reconnect T, while ensuring that the new intra-bucket edges only connect **consecutive buckets**

$\langle T, \tau, W \rangle$



$u \in B_j \qquad v \in B_j$

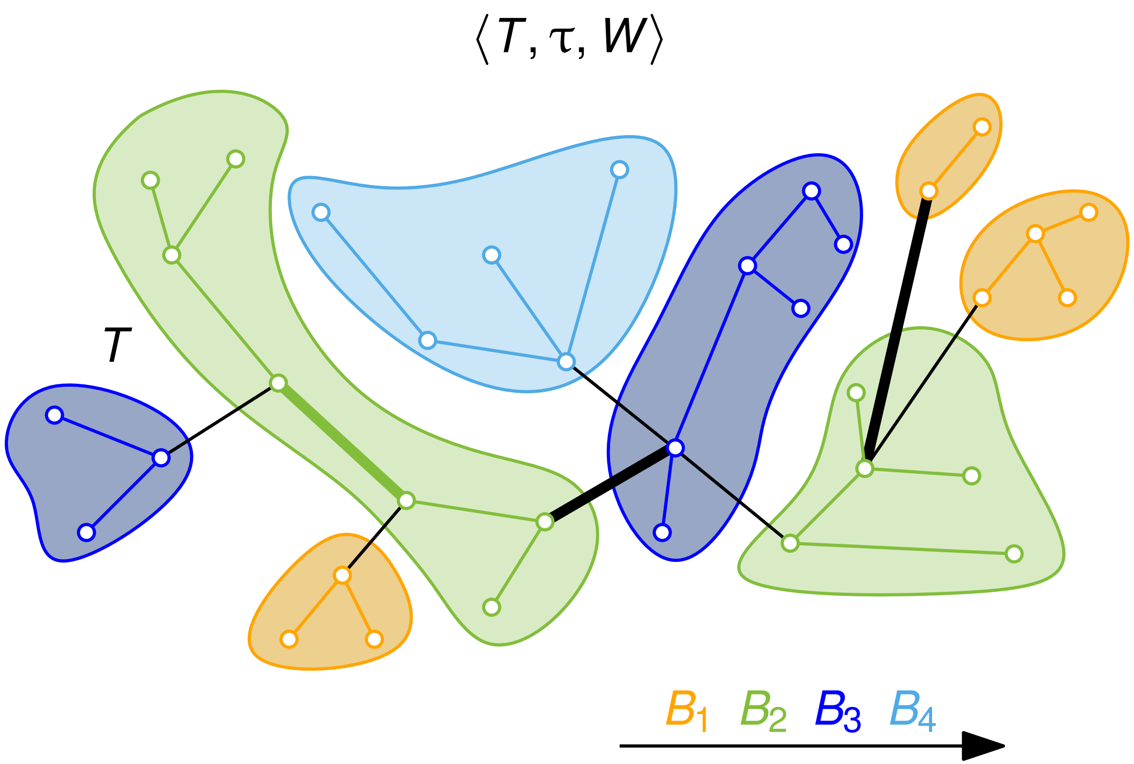
$|j - 1| > 1 \Rightarrow |\tau(u) - \tau(v)| > W$

$u \in B_j \qquad v \in B_j$

$|j - 1| = 1$

Algorithm's Phases 2/5

Phase 2: - add **inter- or intra-bucket edges** to reconnect T, while ensuring that the new intra-bucket edges only connect **consecutive buckets**



$u \in B_j$ $v \in B_j$

$$|j - 1| > 1 \Rightarrow |\tau(u) - \tau(v)| > W$$

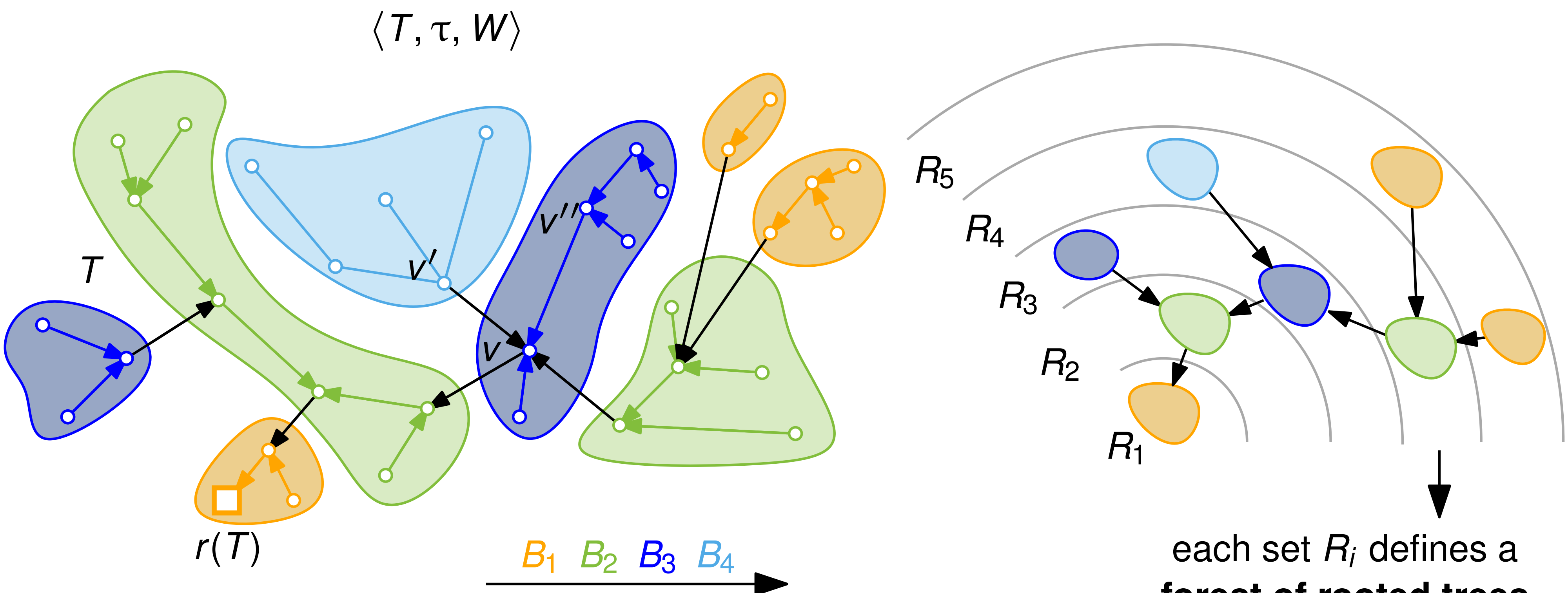
$u \in B_j$ $v \in B_j$

$$|j - 1| = 1$$

pertinent components: maximal connected component of vertices in the same bucket B_i

Algorithm's Phases 3/5

Phase 3: - **root** T at a vertex in a pertinent component of B_1
- assign pertinent components to sets R_i's based on "distance" from the root



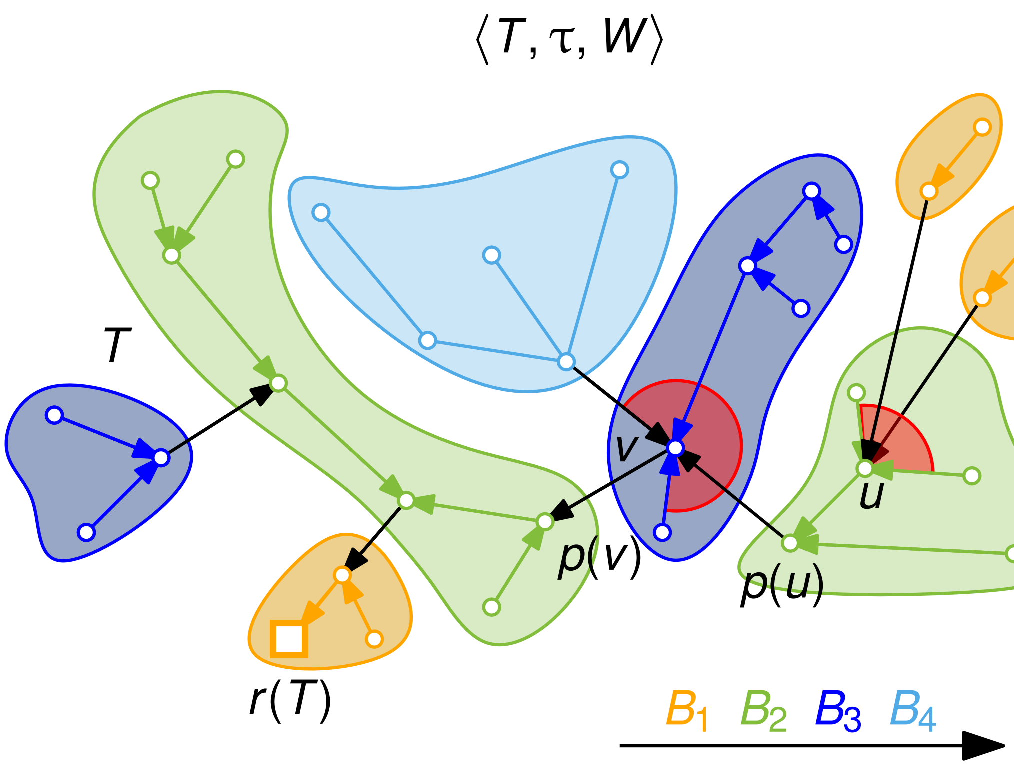
each set R_i defines a **forest of rooted trees**

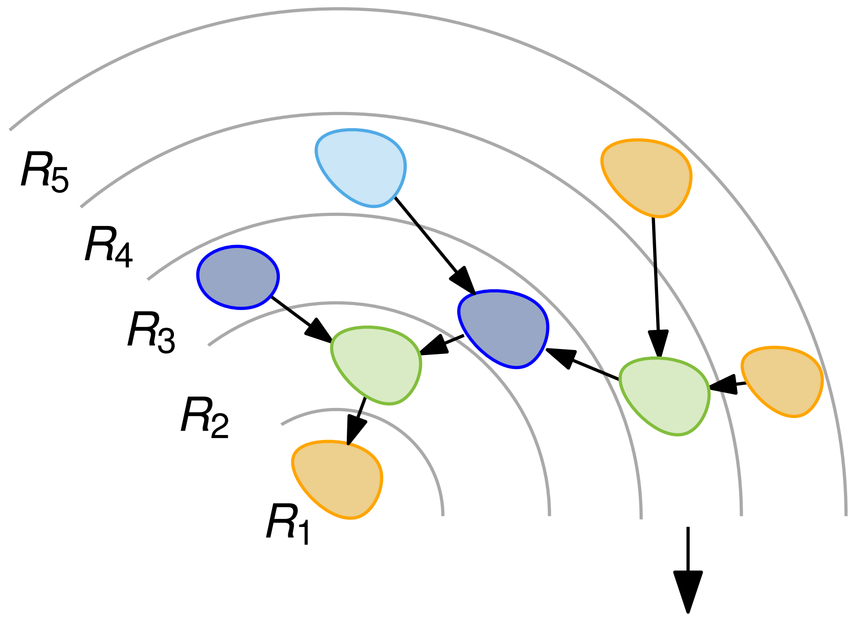
Property 2

The children of a vertex $v \in R_i$ are either in R_i or in R_{i+1}

Algorithm's Phases 4/5

Phase 4: - turn T into an **(rooted) ordered tree** such that:
left-to-right order of the children of $v \in R_i$: children in $R_i \prec$ children in R_{i+1}
(recall Property 2)

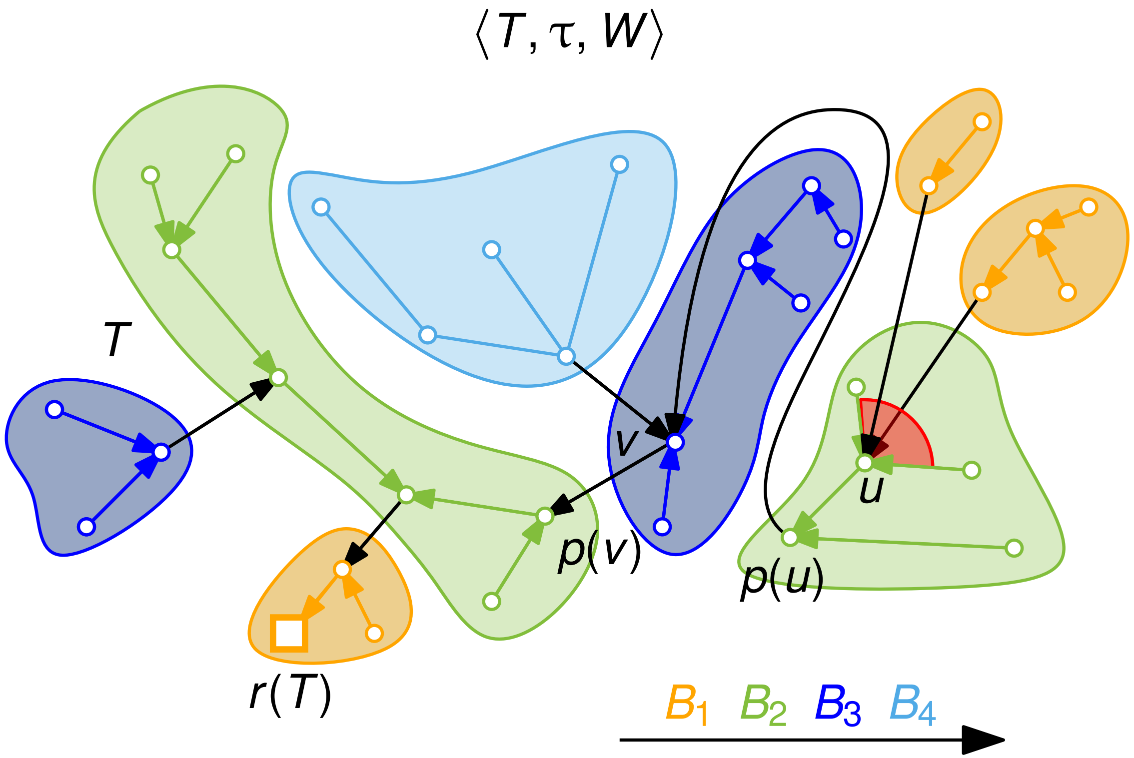


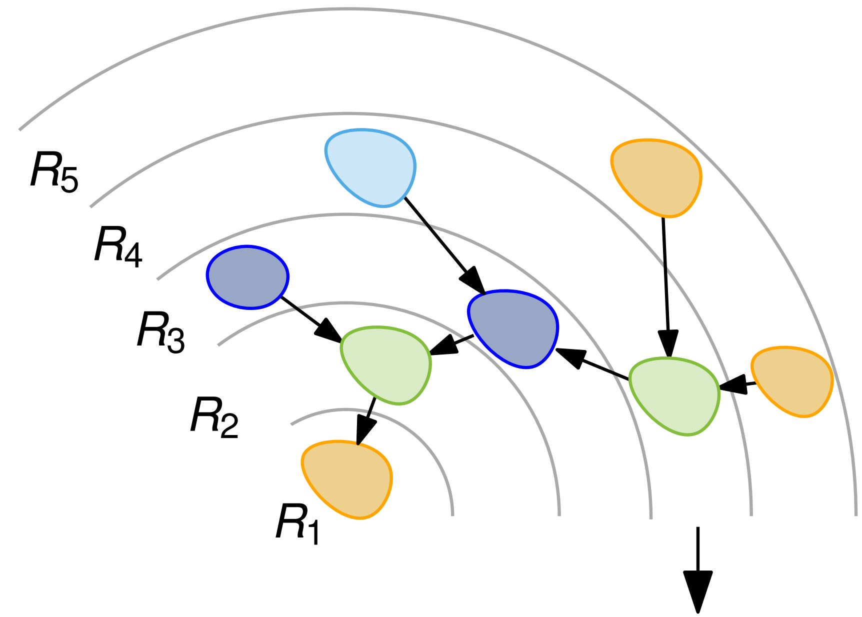


each set R_i defines a **forest of rooted trees**

Algorithm's Phases 4/5

Phase 4: - turn T into an **(rooted) ordered tree** such that:
left-to-right order of the children of $v \in R_i$: children in $R_i \prec$ children in R_{i+1}
(recall Property 2)

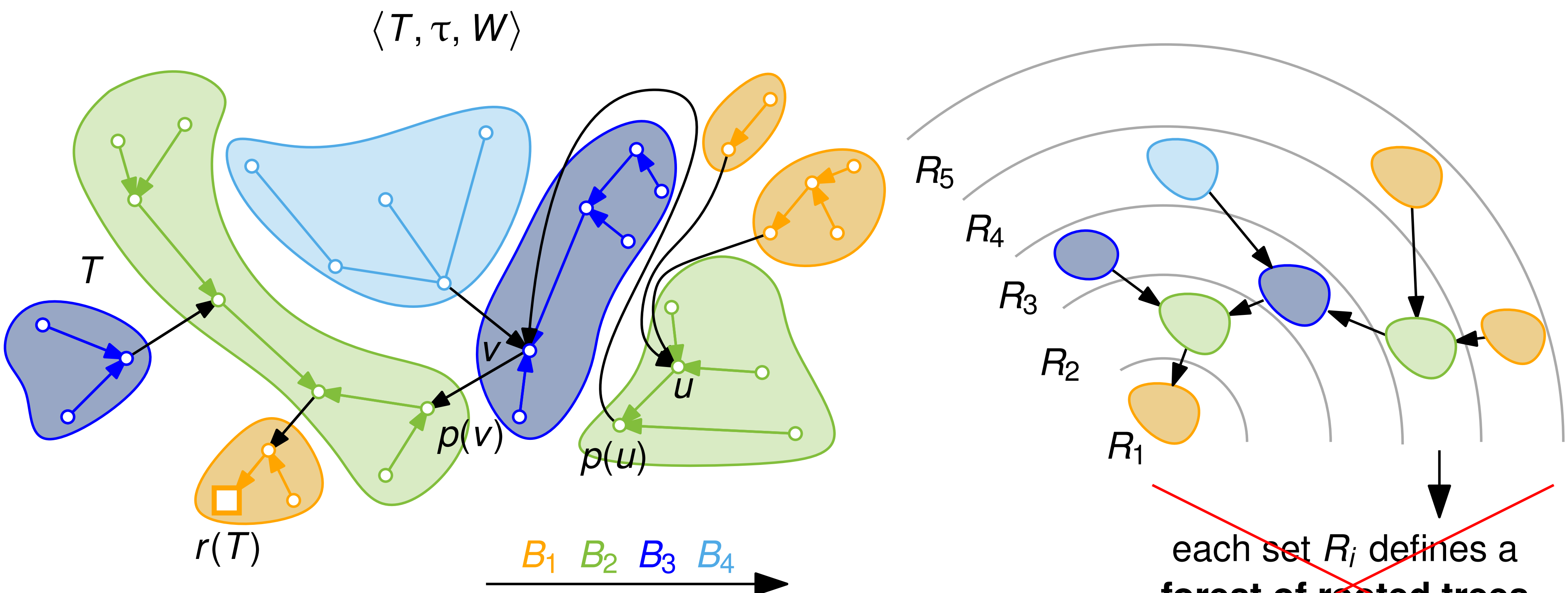




each set R_i defines a
forest of rooted trees

Algorithm's Phases 4/5

Phase 4: - turn T into an **(rooted) ordered tree** such that:
left-to-right order of the children of $v \in R_i$: children in R_i $\prec$ children in R_{i+1}
(recall Property 2)



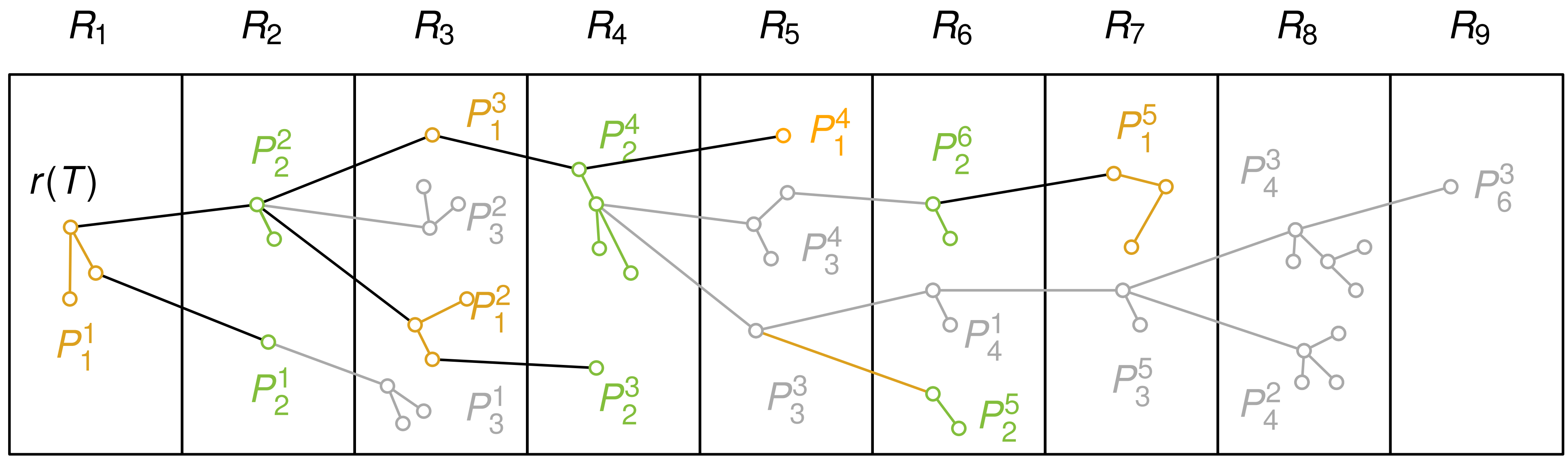
~~each set R_i defines a **forest of rooted trees**~~

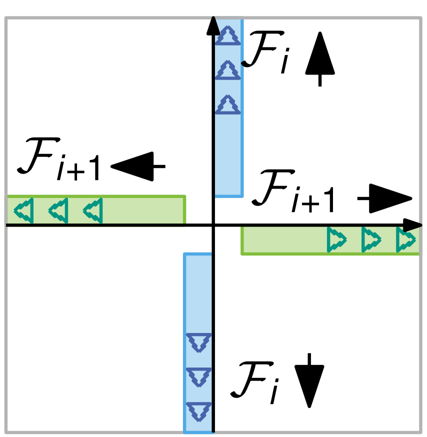
each set R_i defines a **forest of rooted ordered trees**

Algorithm's Phases 5/5

Phase 5: - define **rooted ordered forests** for each B_i, i.e, lists of rooted ordered trees

goal: establish which component is close to which semi-axis and in which order





Forests for the semi-axis:

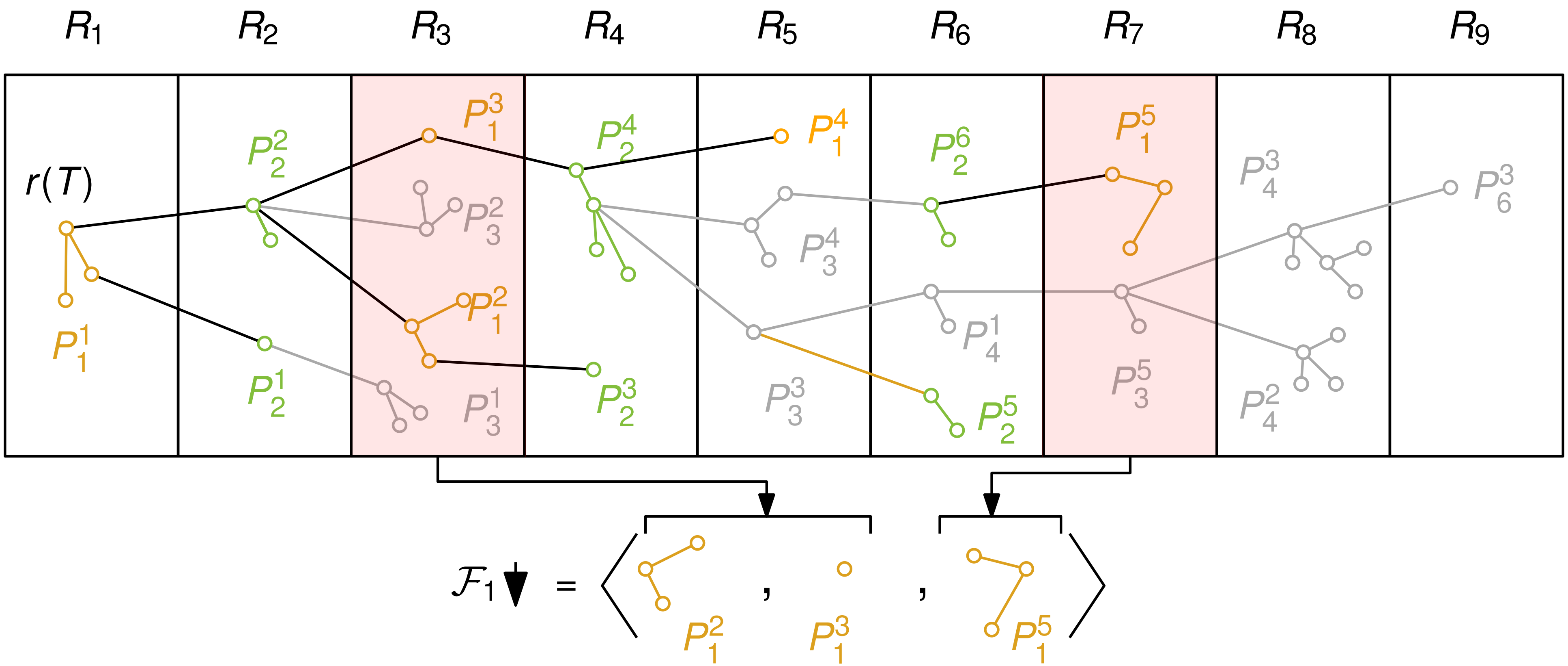
$\mathcal{F}_i \blacktriangle$, $\mathcal{F}_i \blacktriangledown$, $\mathcal{F}_i \blacktriangleright$, $\mathcal{F}_i \blacktriangledown$ — contain all the **pertinent components of B_i** in the set R_j with

- $j \equiv 1 \mod 4$
- $j \equiv 3 \mod 4$
- $j \equiv 2 \mod 4$
- $j \equiv 0 \mod 4$

Algorithm's Phases 5/5

Phase 5: - define **rooted ordered forests** for each B_i, i.e, lists of rooted ordered trees

goal: establish which component is close to which semi-axis and in which order



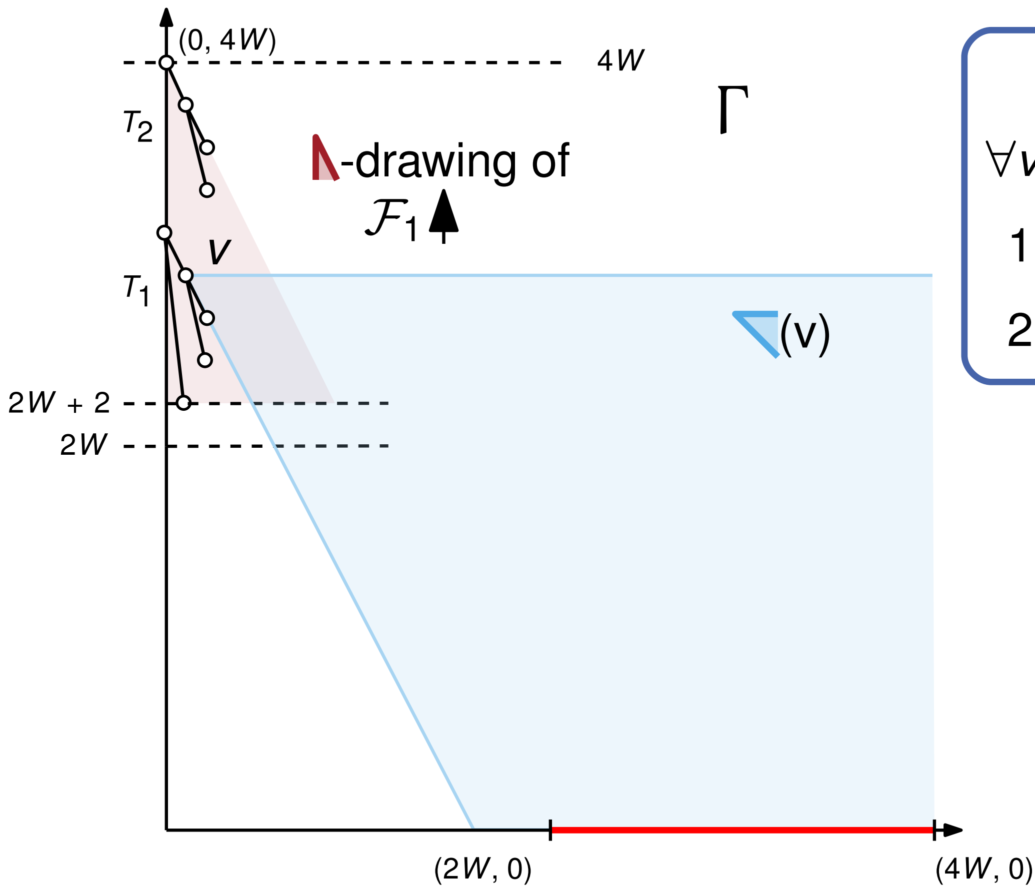
special ordering:

1. primarily, according to the ordering of the R_i's
2. within the same R_i, according to a **counter-clockwise Eulerian tour** of T

-, -, -, and -Drawings of Rooted Ordered Forests

Definition 4

Let $\mathcal{F} = \langle T_1, T_2, \ldots, T_k \rangle$ be a **list of rooted ordered trees**, with a total of $m \leq W$ **vertices**.
A -*drawing* Γ of $\mathcal{F}$ is a **planar straight-line strictly-upward strictly-leftward order-preserving grid drawing** of $\mathcal{F}$ on the $(4W+1) \times (4W+1)$ grid with **special visibility properties**



Geometric Visibility

$\forall v \in V(\mathcal{F}_1)$, the wedge (v) :

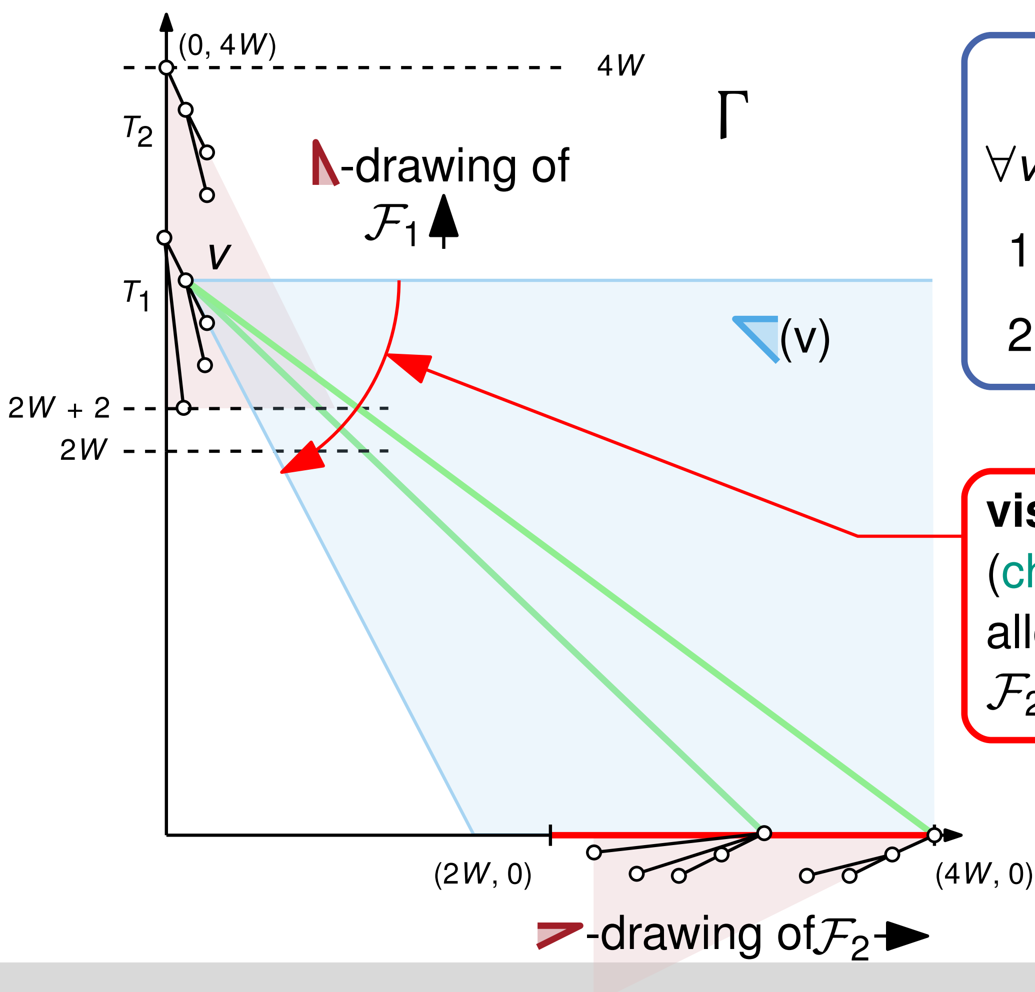
1) does not intersect Γ in its interior

2) contains the segment [(2W,0),(4W,0)]

-, -, -, and -Drawings of Rooted Ordered Forests

Definition 4

Let $\mathcal{F} = \langle T_1, T_2, \ldots, T_k \rangle$ be a **list of rooted ordered trees**, with a total of $m \leq W$ **vertices**.
A -*drawing* Γ of $\mathcal{F}$ is a **planar straight-line strictly-upward strictly-leftward order-preserving grid drawing** of $\mathcal{F}$ on the $(4W+1) \times (4W+1)$ grid with **special visibility properties**



Geometric Visibility

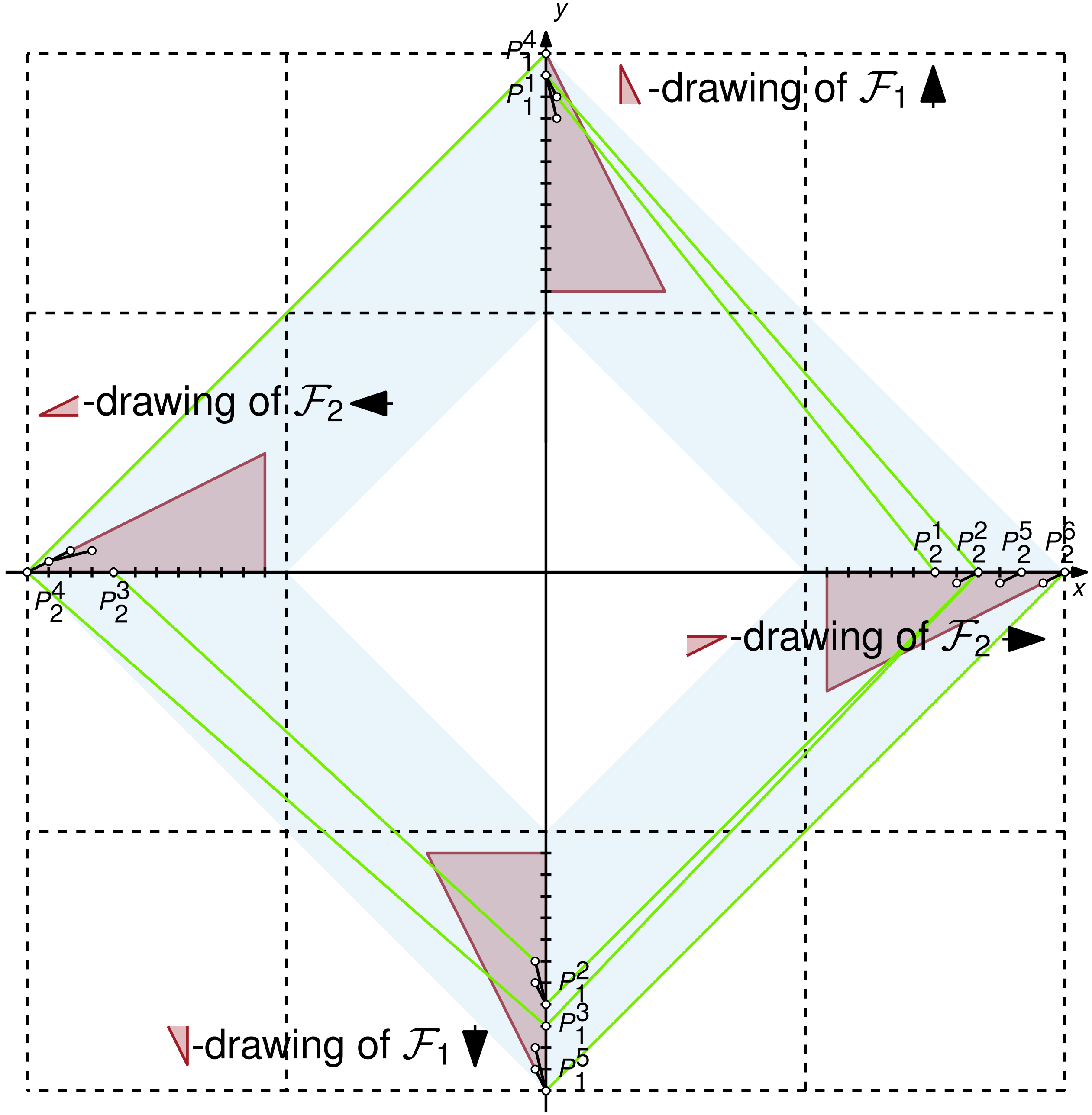
$\forall v \in V(\mathcal{F}_1)$, the wedge (v) :

1) does not intersect Γ in its interior

2) contains the segment [(2W,0),(4W,0)]

visibilty and **embedding choice** in **Phase 4**
(children in $R_i \prec$ children in R_{i+1})
allow us to draw the inter-bucket edges toward $\mathcal{F}_2$ without crossing the drawing of $\mathcal{F}_1$

Drawing of $T[B_i \cup B_{i+1}]$...at last!



How to draw $T[B_1 \cup B_2]$:

- compute the , , , - drawings for the forests $\mathcal{F}_1$, $\mathcal{F}_2$, $\mathcal{F}_1$, $\mathcal{F}_2$
- draw the inter-bucket edges

No two inter-bucket edges cross each other because of the **ordering of the pertinent components** in the forests (**Phase 5**)

Open Problems

Planar Graph Stories

- Do other **notable families** of planar graphs admit straight-line planar drawing stories on a grid of size **polynomial in** W or **polynomial in** $\mathbf{W}$ **and sublinear in** $\mathbf{n}$?
  - e.g.: outerplanar and series-parallel graphs
- What bounds can be shown if G is not a tree, but **each** G_t **is a forest**?
- Which bounds can be shown if we allow **bends**?

Different Models

- the **same vertex** is allowed to appear **several times**
- **multiple vertices** may appear simultaneously (and still $|V(G_t)| \leq W$)
- how about **edges**, and not vertices, appearing over time?

Graph Stories in Small Area

27th Int. Symposium on Graph Drawing and Network Visualization (GD 2019)

Manuel Borrazzo, **Giordano Da Lozzo**,
Fabrizio Frati, Maurizio Patrignani

Department of Engineering · Roma Tre University

Thanks for your attention!!

Open Problems

Planar Graph Stories

- Do other **notable families** of planar graphs admit straight-line planar drawing stories on a grid of size **polynomial in** W or **polynomial in** $\mathbf{W}$ **and sublinear in** $\mathbf{n}$?
  - e.g.: outerplanar and series-parallel graphs
- What bounds can be shown if G is not a tree, but **each** G_t **is a forest**?
- Which bounds can be shown if we allow **bends**?

Different Models

- the **same vertex** is allowed to appear **several times**
- **multiple vertices** may appear simultaneously (and still $|V(G_t)| \leq W$)
- how about **edges**, and not vertices, appearing over time?