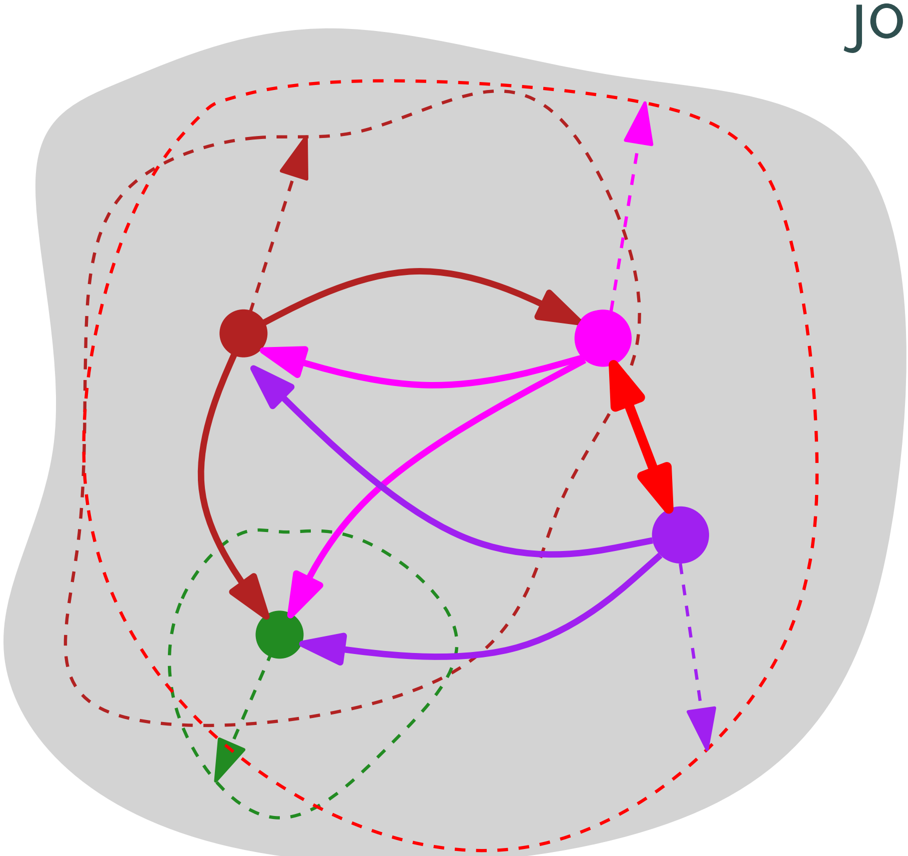
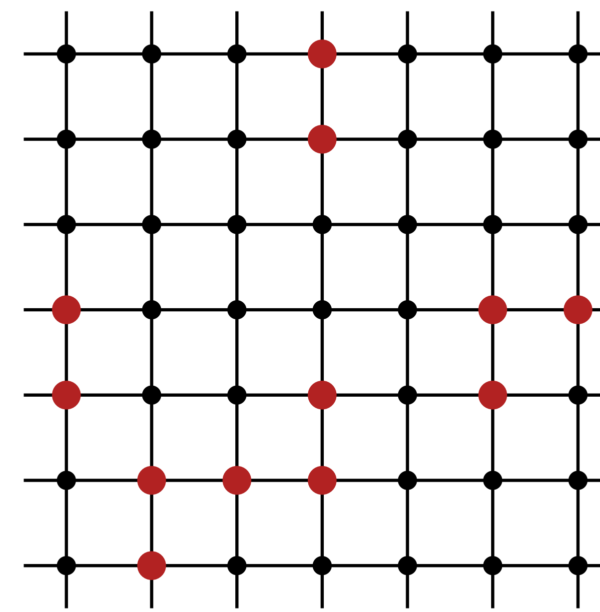
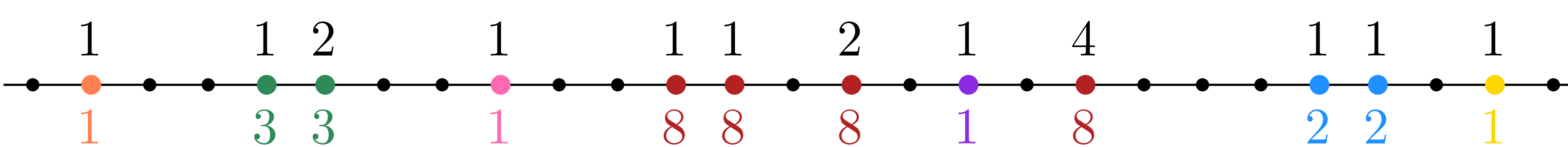
Percolation by cumulative merging and phase transition of the contact process

Laurent Ménard (Paris Nanterre)

joint work with **Arvind Singh** (Paris Orsay)





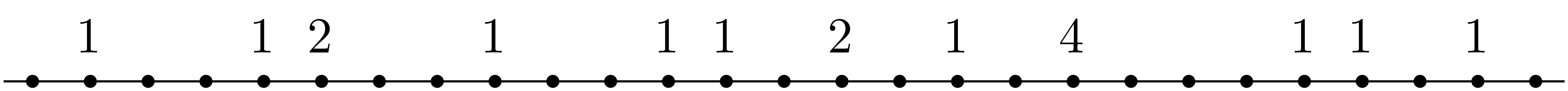


Outline

1. Cumulative merging
2. Phase transition for cumulative merging percolation
3. The contact process
4. Heuristics for the contact process
5. Link with cumulative merging

Cumulative Merging?

Take $G = (V, E, r)$ any locally finite connected weighted graph with $r : V \to [0, +\infty]$

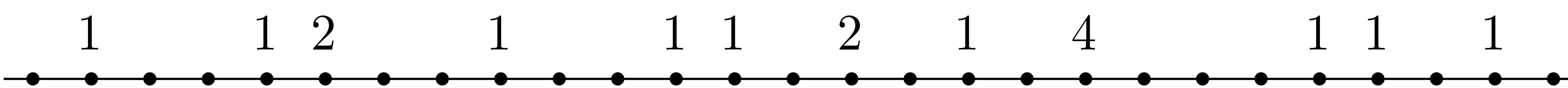


Cumulative Merging?

Take $G = (V, E, r)$ any locally finite connected weighted graph with $r : V \to [0, +\infty]$

Construct a partition $\mathscr{P}$ of V with the following algorithm:

1. Start with the finest partition $\mathscr{P} = \{\{x\}\}_{x \in V}$.
2. Pick $x \neq y \in V$. Denote $\mathscr{P}_x$ the element of $\mathscr{P}$ containing x.
3. If $d_G(\mathscr{P}_x, \mathscr{P}_y) \leq \min\{r(\mathscr{P}_x), r(\mathscr{P}_y)\}$, merge $\mathscr{P}_x$ and $\mathscr{P}_y$
4. Go back to 2.

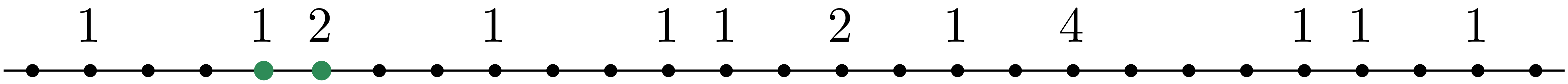


Cumulative Merging?

Take $G = (V, E, r)$ any locally finite connected weighted graph with $r : V \to [0, +\infty]$

Construct a partition $\mathscr{P}$ of V with the following algorithm:

1. Start with the finest partition $\mathscr{P} = \{\{x\}\}_{x \in V}$.
2. Pick $x \neq y \in V$. Denote $\mathscr{P}_x$ the element of $\mathscr{P}$ containing x.
3. If $d_G(\mathscr{P}_x, \mathscr{P}_y) \leq \min\{r(\mathscr{P}_x), r(\mathscr{P}_y)\}$, merge $\mathscr{P}_x$ and $\mathscr{P}_y$
4. Go back to 2.

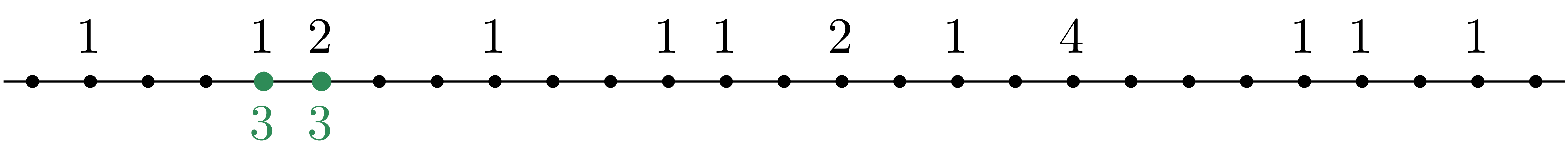


Cumulative Merging?

Take $G = (V, E, r)$ any locally finite connected weighted graph with $r : V \to [0, +\infty]$

Construct a partition $\mathscr{P}$ of V with the following algorithm:

1. Start with the finest partition $\mathscr{P} = \{\{x\}\}_{x \in V}$.
2. Pick $x \neq y \in V$. Denote $\mathscr{P}_x$ the element of $\mathscr{P}$ containing x.
3. If $d_G(\mathscr{P}_x, \mathscr{P}_y) \leq \min\{r(\mathscr{P}_x), r(\mathscr{P}_y)\}$, merge $\mathscr{P}_x$ and $\mathscr{P}_y$
4. Go back to 2.

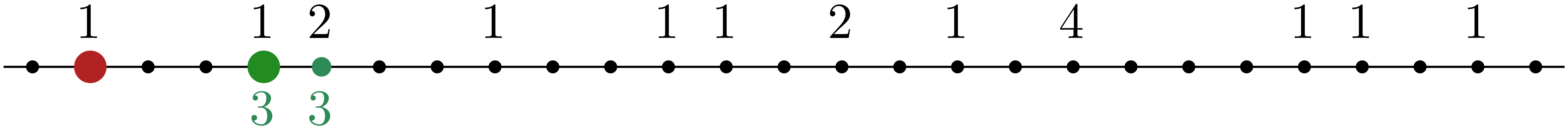


Cumulative Merging?

Take $G = (V, E, r)$ any locally finite connected weighted graph with $r : V \to [0, +\infty]$

Construct a partition $\mathscr{P}$ of V with the following algorithm:

1. Start with the finest partition $\mathscr{P} = \{\{x\}\}_{x \in V}$.
2. Pick $x \neq y \in V$. Denote $\mathscr{P}_x$ the element of $\mathscr{P}$ containing x.
3. If $d_G(\mathscr{P}_x, \mathscr{P}_y) \leq \min\{r(\mathscr{P}_x), r(\mathscr{P}_y)\}$, merge $\mathscr{P}_x$ and $\mathscr{P}_y$
4. Go back to 2.

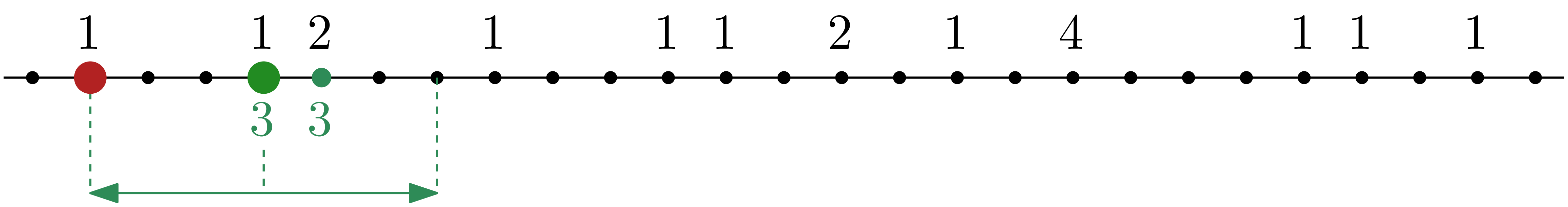


Cumulative Merging?

Take $G = (V, E, r)$ any locally finite connected weighted graph with $r : V \to [0, +\infty]$

Construct a partition $\mathscr{P}$ of V with the following algorithm:

1. Start with the finest partition $\mathscr{P} = \{\{x\}\}_{x \in V}$.
2. Pick $x \neq y \in V$. Denote $\mathscr{P}_x$ the element of $\mathscr{P}$ containing x.
3. If $d_G(\mathscr{P}_x, \mathscr{P}_y) \leq \min\{r(\mathscr{P}_x), r(\mathscr{P}_y)\}$, merge $\mathscr{P}_x$ and $\mathscr{P}_y$
4. Go back to 2.

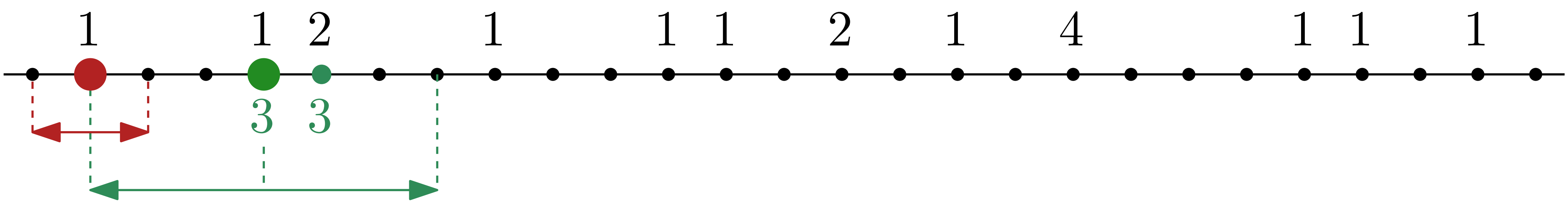


Cumulative Merging?

Take $G = (V, E, r)$ any locally finite connected weighted graph with $r : V \to [0, +\infty]$

Construct a partition $\mathscr{P}$ of V with the following algorithm:

1. Start with the finest partition $\mathscr{P} = \{\{x\}\}_{x \in V}$.
2. Pick $x \neq y \in V$. Denote $\mathscr{P}_x$ the element of $\mathscr{P}$ containing x.
3. If $d_G(\mathscr{P}_x, \mathscr{P}_y) \leq \min\{r(\mathscr{P}_x), r(\mathscr{P}_y)\}$, merge $\mathscr{P}_x$ and $\mathscr{P}_y$
4. Go back to 2.

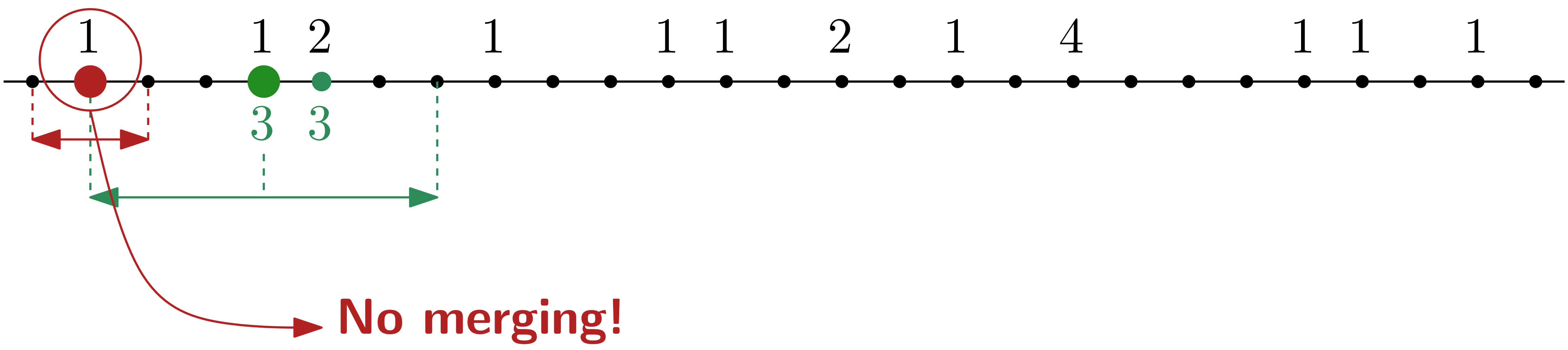


Cumulative Merging?

Take $G = (V, E, r)$ any locally finite connected weighted graph with $r : V \to [0, +\infty]$

Construct a partition $\mathscr{P}$ of V with the following algorithm:

1. Start with the finest partition $\mathscr{P} = \{\{x\}\}_{x \in V}$.
2. Pick $x \neq y \in V$. Denote $\mathscr{P}_x$ the element of $\mathscr{P}$ containing x.
3. If $d_G(\mathscr{P}_x, \mathscr{P}_y) \leq \min\{r(\mathscr{P}_x), r(\mathscr{P}_y)\}$, merge $\mathscr{P}_x$ and $\mathscr{P}_y$
4. Go back to 2.

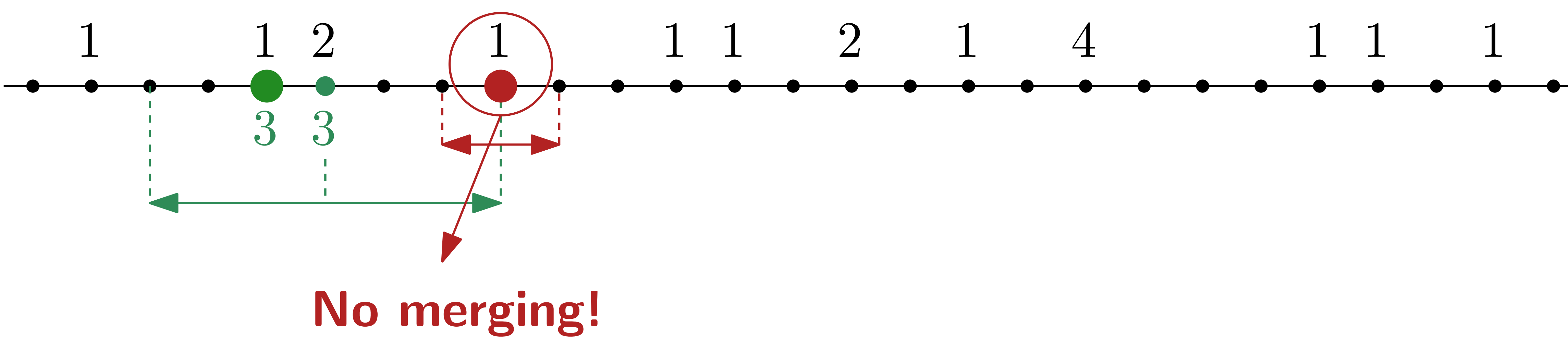


Cumulative Merging?

Take $G = (V, E, r)$ any locally finite connected weighted graph with $r : V \to [0, +\infty]$

Construct a partition $\mathscr{P}$ of V with the following algorithm:

1. Start with the finest partition $\mathscr{P} = \{\{x\}\}_{x \in V}$.
2. Pick $x \neq y \in V$. Denote $\mathscr{P}_x$ the element of $\mathscr{P}$ containing x.
3. If $d_G(\mathscr{P}_x, \mathscr{P}_y) \leq \min\{r(\mathscr{P}_x), r(\mathscr{P}_y)\}$, merge $\mathscr{P}_x$ and $\mathscr{P}_y$
4. Go back to 2.

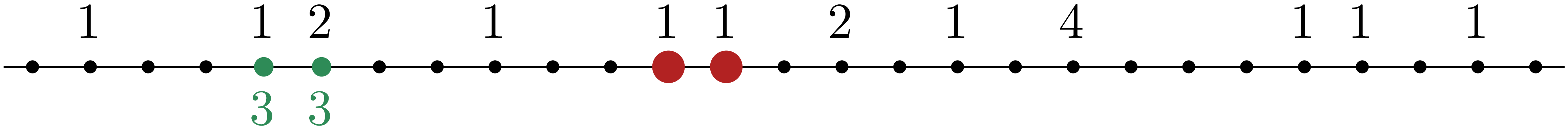


Cumulative Merging?

Take $G = (V, E, r)$ any locally finite connected weighted graph with $r : V \to [0, +\infty]$

Construct a partition $\mathscr{P}$ of V with the following algorithm:

1. Start with the finest partition $\mathscr{P} = \{\{x\}\}_{x \in V}$.
2. Pick $x \neq y \in V$. Denote $\mathscr{P}_x$ the element of $\mathscr{P}$ containing x.
3. If $d_G(\mathscr{P}_x, \mathscr{P}_y) \leq \min\{r(\mathscr{P}_x), r(\mathscr{P}_y)\}$, merge $\mathscr{P}_x$ and $\mathscr{P}_y$
4. Go back to 2.

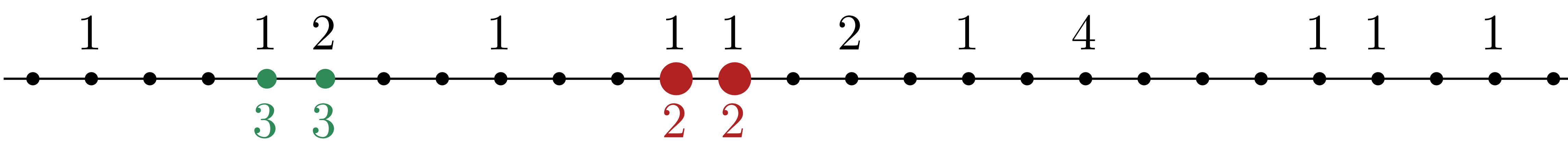


Cumulative Merging?

Take $G = (V, E, r)$ any locally finite connected weighted graph with $r : V \to [0, +\infty]$

Construct a partition $\mathscr{P}$ of V with the following algorithm:

1. Start with the finest partition $\mathscr{P} = \{\{x\}\}_{x \in V}$.
2. Pick $x \neq y \in V$. Denote $\mathscr{P}_x$ the element of $\mathscr{P}$ containing x.
3. If $d_G(\mathscr{P}_x, \mathscr{P}_y) \leq \min\{r(\mathscr{P}_x), r(\mathscr{P}_y)\}$, merge $\mathscr{P}_x$ and $\mathscr{P}_y$
4. Go back to 2.

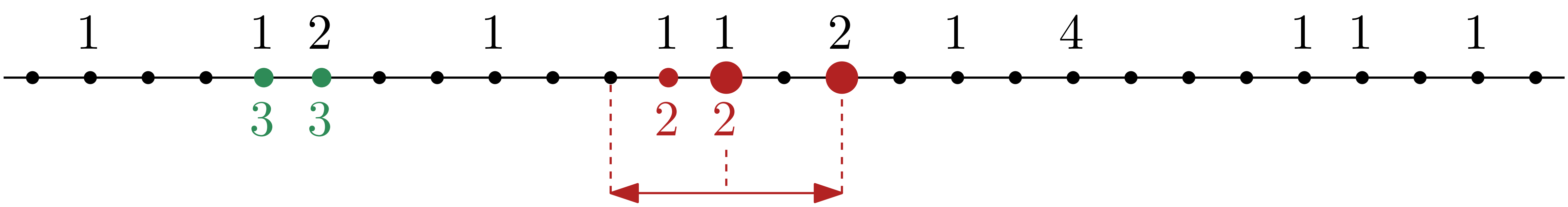


Cumulative Merging?

Take $G = (V, E, r)$ any locally finite connected weighted graph with $r : V \to [0, +\infty]$

Construct a partition $\mathscr{P}$ of V with the following algorithm:

1. Start with the finest partition $\mathscr{P} = \{\{x\}\}_{x \in V}$.
2. Pick $x \neq y \in V$. Denote $\mathscr{P}_x$ the element of $\mathscr{P}$ containing x.
3. If $d_G(\mathscr{P}_x, \mathscr{P}_y) \leq \min\{r(\mathscr{P}_x), r(\mathscr{P}_y)\}$, merge $\mathscr{P}_x$ and $\mathscr{P}_y$
4. Go back to 2.

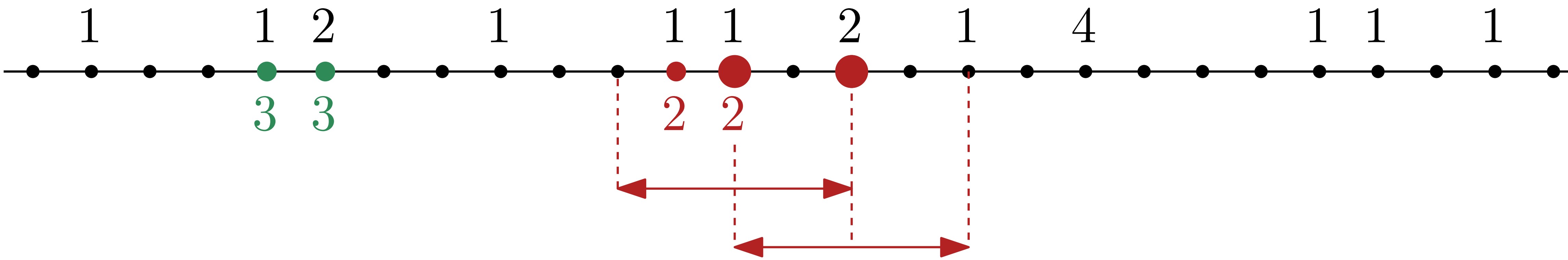


Cumulative Merging?

Take $G = (V, E, r)$ any locally finite connected weighted graph with $r : V \to [0, +\infty]$

Construct a partition $\mathscr{P}$ of V with the following algorithm:

1. Start with the finest partition $\mathscr{P} = \{\{x\}\}_{x \in V}$.
2. Pick $x \neq y \in V$. Denote $\mathscr{P}_x$ the element of $\mathscr{P}$ containing x.
3. If $d_G(\mathscr{P}_x, \mathscr{P}_y) \leq \min\{r(\mathscr{P}_x), r(\mathscr{P}_y)\}$, merge $\mathscr{P}_x$ and $\mathscr{P}_y$
4. Go back to 2.

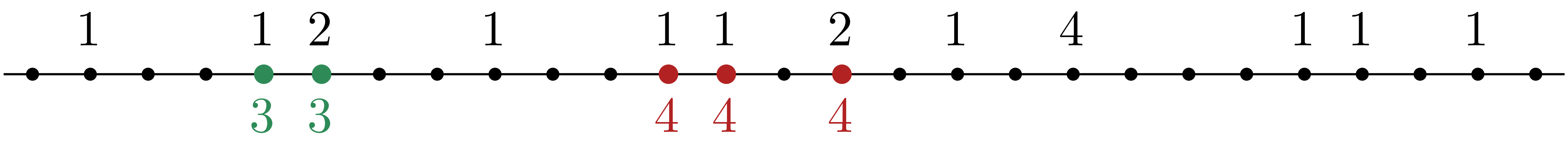


Cumulative Merging?

Take $G = (V, E, r)$ any locally finite connected weighted graph with $r : V \to [0, +\infty]$

Construct a partition $\mathscr{P}$ of V with the following algorithm:

1. Start with the finest partition $\mathscr{P} = \{\{x\}\}_{x \in V}$.
2. Pick $x \neq y \in V$. Denote $\mathscr{P}_x$ the element of $\mathscr{P}$ containing x.
3. If $d_G(\mathscr{P}_x, \mathscr{P}_y) \leq \min\{r(\mathscr{P}_x), r(\mathscr{P}_y)\}$, merge $\mathscr{P}_x$ and $\mathscr{P}_y$
4. Go back to 2.

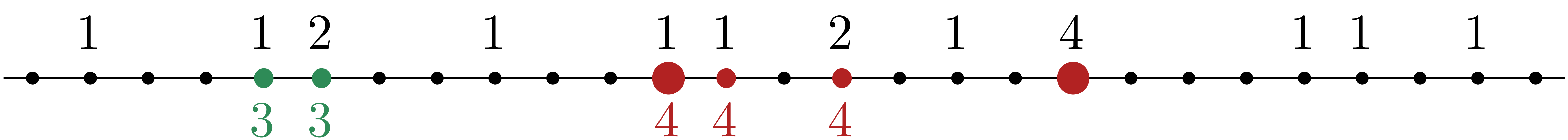


Cumulative Merging?

Take $G = (V, E, r)$ any locally finite connected weighted graph with $r : V \to [0, +\infty]$

Construct a partition $\mathscr{P}$ of V with the following algorithm:

1. Start with the finest partition $\mathscr{P} = \{\{x\}\}_{x \in V}$.
2. Pick $x \neq y \in V$. Denote $\mathscr{P}_x$ the element of $\mathscr{P}$ containing x.
3. If $d_G(\mathscr{P}_x, \mathscr{P}_y) \leq \min\{r(\mathscr{P}_x), r(\mathscr{P}_y)\}$, merge $\mathscr{P}_x$ and $\mathscr{P}_y$
4. Go back to 2.

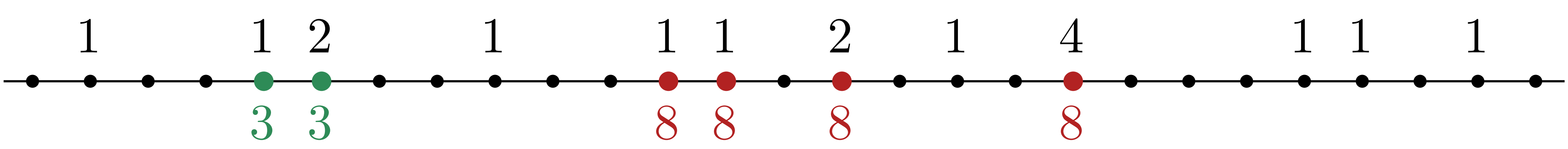


Cumulative Merging?

Take $G = (V, E, r)$ any locally finite connected weighted graph with $r : V \to [0, +\infty]$

Construct a partition $\mathscr{P}$ of V with the following algorithm:

1. Start with the finest partition $\mathscr{P} = \{\{x\}\}_{x \in V}$.
2. Pick $x \neq y \in V$. Denote $\mathscr{P}_x$ the element of $\mathscr{P}$ containing x.
3. If $d_G(\mathscr{P}_x, \mathscr{P}_y) \leq \min\{r(\mathscr{P}_x), r(\mathscr{P}_y)\}$, merge $\mathscr{P}_x$ and $\mathscr{P}_y$
4. Go back to 2.

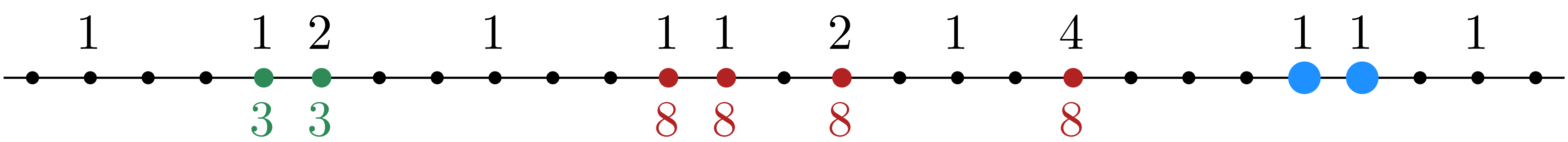


Cumulative Merging?

Take $G = (V, E, r)$ any locally finite connected weighted graph with $r : V \to [0, +\infty]$

Construct a partition $\mathscr{P}$ of V with the following algorithm:

1. Start with the finest partition $\mathscr{P} = \{\{x\}\}_{x \in V}$.
2. Pick $x \neq y \in V$. Denote $\mathscr{P}_x$ the element of $\mathscr{P}$ containing x.
3. If $d_G(\mathscr{P}_x, \mathscr{P}_y) \leq \min\{r(\mathscr{P}_x), r(\mathscr{P}_y)\}$, merge $\mathscr{P}_x$ and $\mathscr{P}_y$
4. Go back to 2.

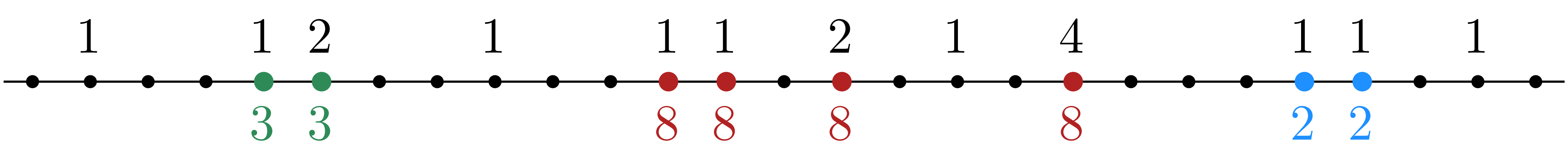


Cumulative Merging?

Take $G = (V, E, r)$ any locally finite connected weighted graph with $r : V \to [0, +\infty]$

Construct a partition $\mathscr{P}$ of V with the following algorithm:

1. Start with the finest partition $\mathscr{P} = \{\{x\}\}_{x \in V}$.
2. Pick $x \neq y \in V$. Denote $\mathscr{P}_x$ the element of $\mathscr{P}$ containing x.
3. If $d_G(\mathscr{P}_x, \mathscr{P}_y) \leq \min\{r(\mathscr{P}_x), r(\mathscr{P}_y)\}$, merge $\mathscr{P}_x$ and $\mathscr{P}_y$
4. Go back to 2.

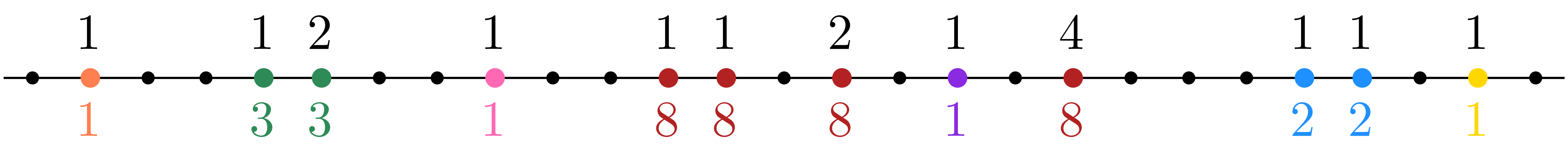


Cumulative Merging?

Take $G = (V, E, r)$ any locally finite connected weighted graph with $r : V \to [0, +\infty]$

Construct a partition $\mathscr{P}$ of V with the following algorithm:

1. Start with the finest partition $\mathscr{P} = \{\{x\}\}_{x \in V}$.
2. Pick $x \neq y \in V$. Denote $\mathscr{P}_x$ the element of $\mathscr{P}$ containing x.
3. If $d_G(\mathscr{P}_x, \mathscr{P}_y) \leq \min\{r(\mathscr{P}_x), r(\mathscr{P}_y)\}$, merge $\mathscr{P}_x$ and $\mathscr{P}_y$
4. Go back to 2.



Cumulative Merging: Admissible partitions

Consider a weighted graph $G = (V, E, r)$ with $r : V \to [0, \infty]$.

Definition

a partition $\mathscr{P}$ of V is **admissible** *iff* $\forall A \neq B \in \mathscr{P}$:

$$d_G(A, B) > r(A) \wedge r(B).$$

Cumulative Merging: Admissible partitions

Consider a weighted graph $G = (V, E, r)$ with $r : V \to [0, \infty]$.

Definition

a partition $\mathscr{P}$ of V is **admissible** *iff* $\forall A \neq B \in \mathscr{P}$:

$$d_G(A, B) > r(A) \wedge r(B).$$

- $\{V\}$ is admissible.
- If $\mathscr{P}_1$ and $\mathscr{P}_2$ are admissible, then

$$\mathscr{P}_1 \wedge \mathscr{P}_2 := \{C_1 \cap C_2 : C_1 \in \mathscr{P}_1, C_2 \in \mathscr{P}_2\}$$

  is admissible.

Cumulative Merging: Admissible partitions

Consider a weighted graph $G = (V, E, r)$ with $r : V \to [0, \infty]$.

Definition

a partition $\mathscr{P}$ of V is **admissible** *iff* $\forall A \neq B \in \mathscr{P}$:

$$d_G(A, B) > r(A) \wedge r(B).$$

- $\{V\}$ is admissible.
- If $\mathscr{P}_1$ and $\mathscr{P}_2$ are admissible, then

$$\mathscr{P}_1 \wedge \mathscr{P}_2 := \{C_1 \cap C_2 : C_1 \in \mathscr{P}_1, C_2 \in \mathscr{P}_2\}$$

is admissible.

Definition

$$\mathscr{C}(G, r) := \bigwedge_{\text{admissible } \mathscr{P}} \mathscr{P} \qquad \text{(finest admissible partition)}$$

Merging operators

Definition: Merging operators

For $x \neq y \in V$, $M_{x,y}$: {partitions of V} $\to$ {partitions of V} defined by

$$M_{x,y}(\mathscr{P}) := \begin{cases} (\mathscr{P} \setminus \{\mathscr{P}_x, \mathscr{P}_y\}) \cup \{\mathscr{P}_x \cup \mathscr{P}_y\} & \text{if } \mathscr{P}_x \neq \mathscr{P}_y \text{ and} \\ & d(x,y) \leq r(\mathscr{P}_x) \wedge r(\mathscr{P}_y), \\ \mathscr{P} & \text{otherwise.} \end{cases}$$

for every partition $\mathscr{P}$, where $\mathscr{P}_x$ is the **cluster** of x in $\mathscr{P}$.

Merging operators

Definition: Merging operators

For $x \neq y \in V$, $M_{x,y}$: {partitions of V} $\to$ {partitions of V} defined by

$$M_{x,y}(\mathscr{P}) := \begin{cases} (\mathscr{P} \setminus \{\mathscr{P}_x, \mathscr{P}_y\}) \cup \{\mathscr{P}_x \cup \mathscr{P}_y\} & \text{if } \mathscr{P}_x \neq \mathscr{P}_y \text{ and} \\ & d(x,y) \leq r(\mathscr{P}_x) \wedge r(\mathscr{P}_y), \\ \mathscr{P} & \text{otherwise.} \end{cases}$$

for every partition $\mathscr{P}$, where $\mathscr{P}_x$ is the **cluster** of x in $\mathscr{P}$.

Proposition:

The merging operators are monotone: for every $x \neq y \in V$ and every partitions $\mathscr{P}$ and $\mathscr{P}'$

- $\mathscr{P}$ is finer than $M_{x,y}(\mathscr{P})$;
- If $\mathscr{P}$ is finer than $\mathscr{P}'$, then $M_{x,y}(\mathscr{P})$ is finer than $M_{x,y}(\mathscr{P}')$.

Cumulative Merging

Proposition:
Take $(x_n, y_n) \in V^{\mathbb{N}} \times V^{\mathbb{N}}$ such that for every $x \neq y \in V$:

$$\{x_n, y_n\} = \{x, y\} \text{ for infinitely many } n.$$

Then

$$\mathscr{C}(V, E, r) = \lim_{n\to\infty} \uparrow \; M_{x_n, y_n} \circ \cdots \circ M_{x_1, y_1}(\bar{V})$$

where $\bar{V} = \{\{x\}\}_{x \in V}$ is the finest partition of V.

Cumulative Merging

Proposition:

Take $(x_n, y_n) \in V^{\mathbb{N}} \times V^{\mathbb{N}}$ such that for every $x \neq y \in V$:

$$\{x_n, y_n\} = \{x, y\} \text{ for infinitely many } n.$$

Then

Cumulatively Merged Partition (CMP)

$$\mathscr{C}(V, E, r) = \lim_{n\to\infty} \uparrow\ M_{x_n, y_n} \circ \cdots \circ M_{x_1, y_1}(\bar{V})$$

where $\bar{V} = \{\{x\}\}_{x \in V}$ is the finest partition of V.

Cumulative Merging

Proposition:
Take $(x_n, y_n) \in V^{\mathbb{N}} \times V^{\mathbb{N}}$ such that for every $x \neq y \in V$:

$$\{x_n, y_n\} = \{x, y\} \text{ for infinitely many } n.$$

Then

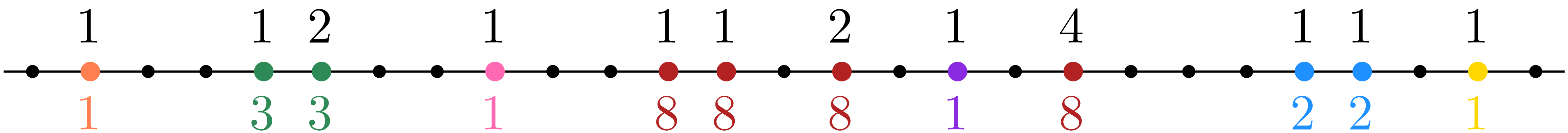
Cumulatively Merged Partition (CMP)

$$\mathscr{C}(V, E, r) = \lim_{n \to \infty} \uparrow \; M_{x_n, y_n} \circ \cdots \circ M_{x_1, y_1}(\bar{V})$$

where $\bar{V} = \{\{x\}\}_{x \in V}$ is the finest partition of V.

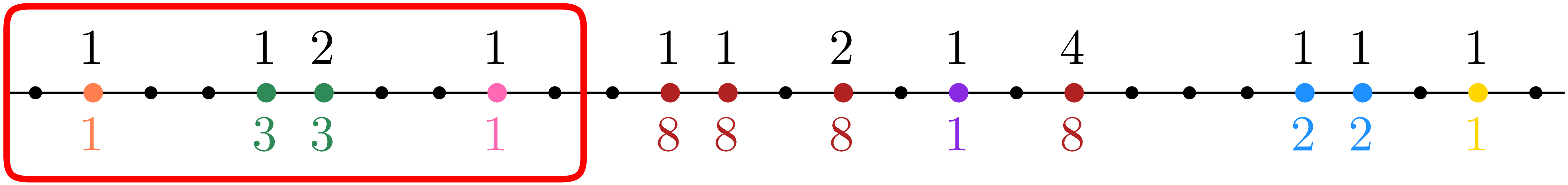
Proof: Monotonicity of the merging operators.

Cumulative Merging: Basic observations



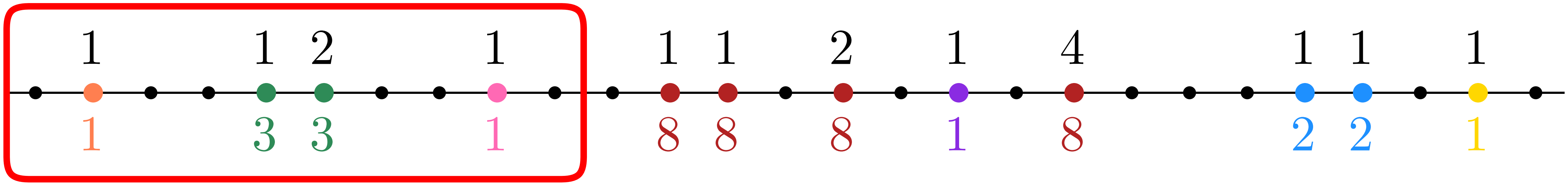
- Clusters in $\mathscr{C}$ are not necessarily connected sets!
- If $r(x) < 1$, then $\{x\} \in \mathscr{C}$.
- If $\mathscr{C}$ has an infinite cluster, it has infinite weight and is unique.
- For any $C \in \mathscr{C}$, one has $|C| \leq \max\{1, r(C)\}$.

CMP: Stable sets



Vertices inside that box will never merge again.

CMP: Stable sets



Vertices inside that box will never merge again.

Definition:
Fix $H \subset V$. We say that H is a **stable set** *iff*:

$$\forall C \in \mathscr{C}(H, E_H, r) \text{ one has } B\left(C, r(C)\right) \subset H.$$

Remark:

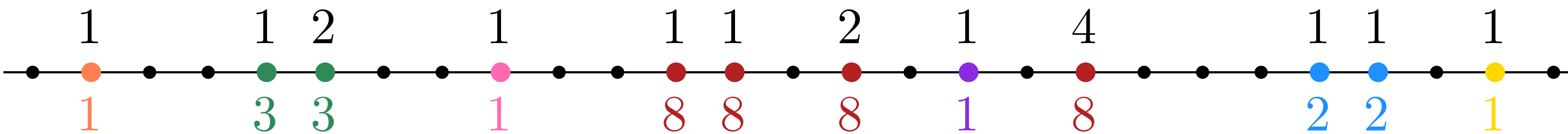
- Unions and intersections of stable sets are stable.
- Being stable is a **local** porperty.
- If H is stable, then

$$\mathscr{C}(G) = \mathscr{C}(H) \sqcup \mathscr{C}(G \setminus H).$$

CMP: Stabilisers

Definition

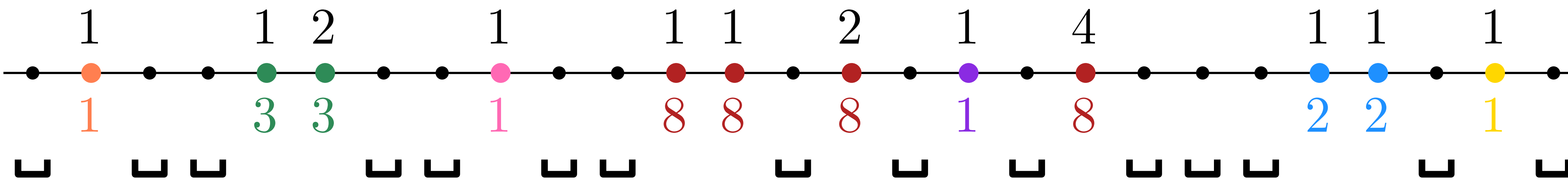
Let $H \subset V$ the **stabiliser** of H, $\mathcal{S}_H$, is the smallest stable set of vertices containing H.



CMP: Stabilisers

Definition

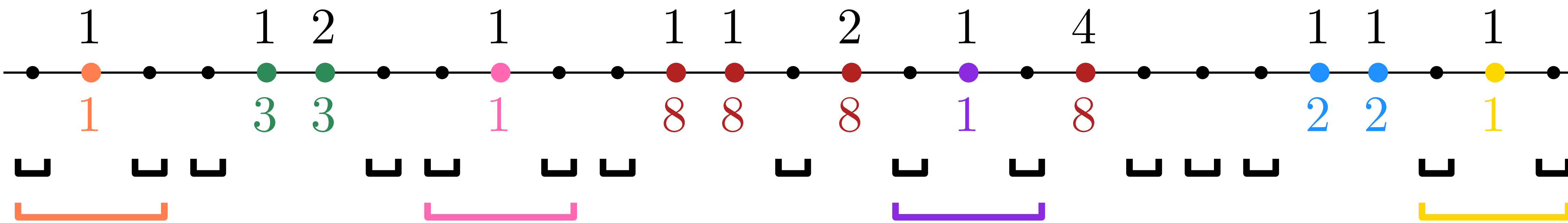
Let $H \subset V$ the **stabiliser** of H, $\mathcal{S}_H$, is the smallest stable set of vertices containing H.



CMP: Stabilisers

Definition

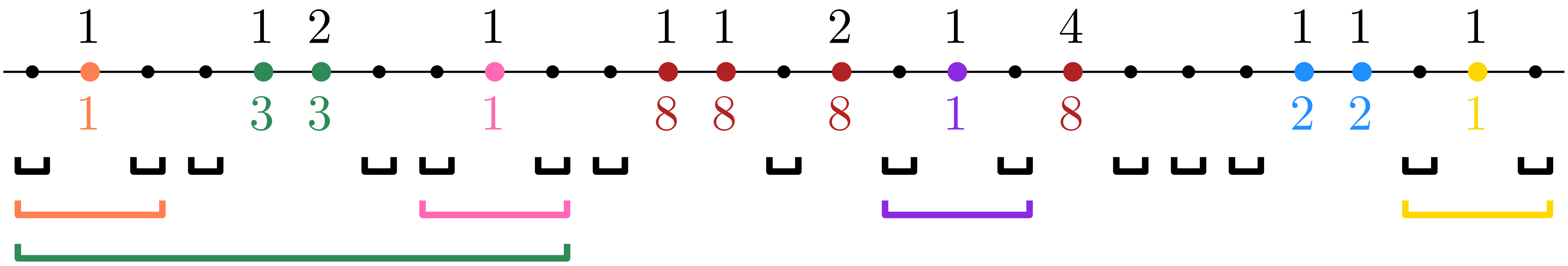
Let $H \subset V$ the **stabiliser** of H, $\mathcal{S}_H$, is the smallest stable set of vertices containing H.



CMP: Stabilisers

Definition

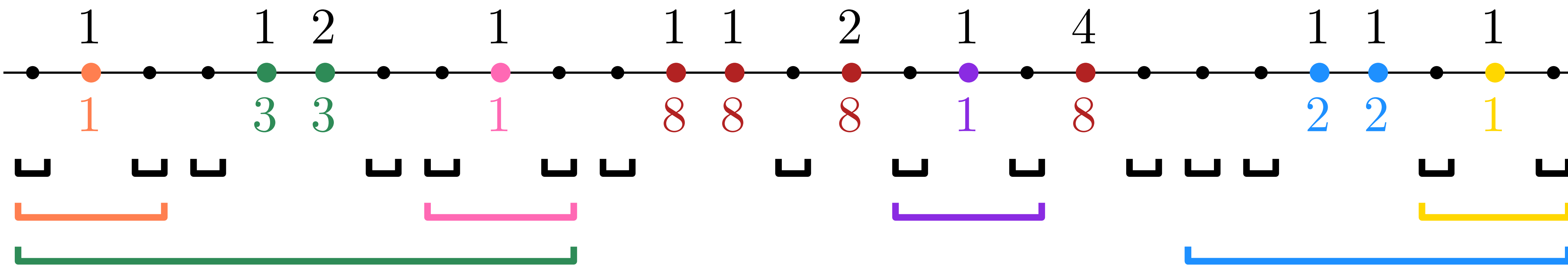
Let $H \subset V$ the **stabiliser** of H, $\mathcal{S}_H$, is the smallest stable set of vertices containing H.



CMP: Stabilisers

Definition

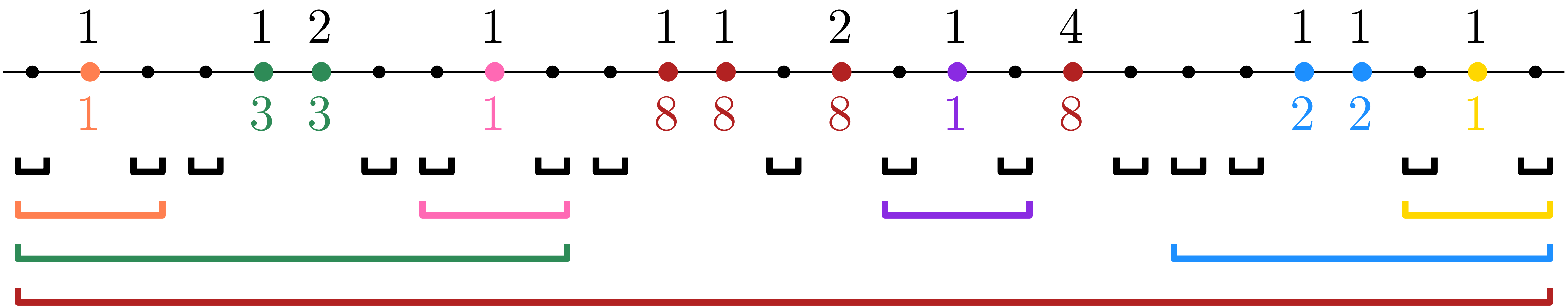
Let $H \subset V$ the **stabiliser** of H, $\mathcal{S}_H$, is the smallest stable set of vertices containing H.



CMP: Stabilisers

Definition

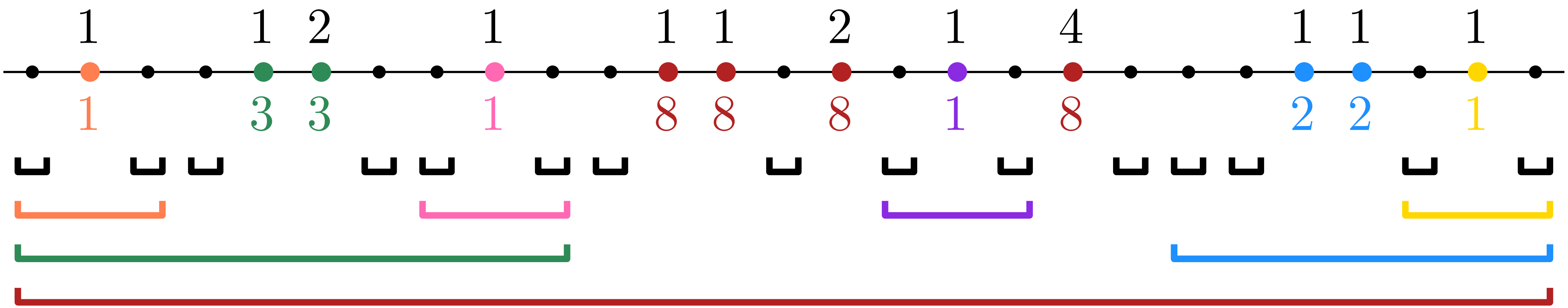
Let $H \subset V$ the **stabiliser** of H, $\mathcal{S}_H$, is the smallest stable set of vertices containing H.



CMP: Stabilisers

Definition

Let $H \subset V$ the **stabiliser** of H, $\mathcal{S}_H$, is the smallest stable set of vertices containing H.

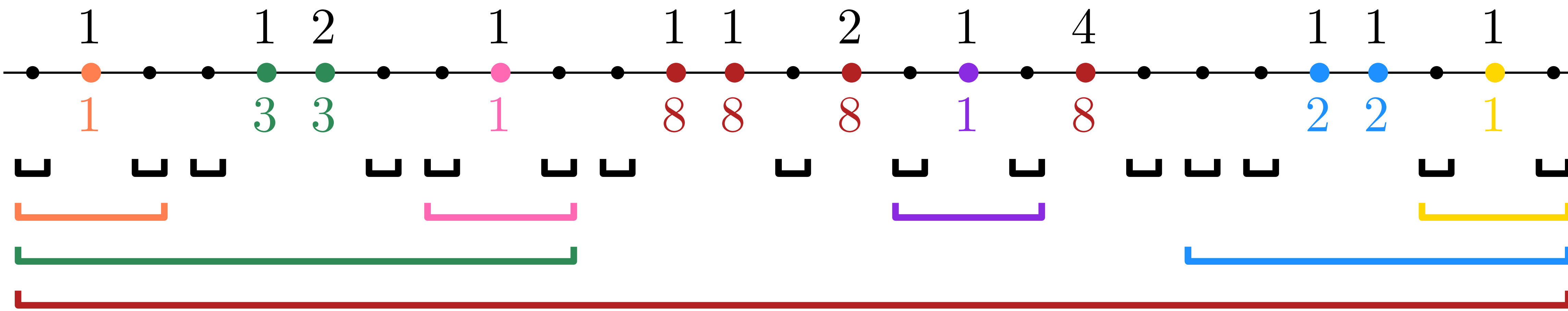


- Stabilisers are connected sets in G.
- Stabilisers are **nested**.
- Nice oriented graph structure on clusters and on stabilisers.

CMP: Stabilisers

Definition

Let $H \subset V$ the **stabiliser** of H, $\mathcal{S}_H$, is the smallest stable set of vertices containing H.



- Stabilisers are connected sets in G.
- Stabilisers are **nested**.
- Nice oriented graph structure on clusters and on stabilisers.

Theorem

Suppose G infinite:

1. $\forall x \in V : |\mathscr{C}_x| = \infty \Leftrightarrow |\mathcal{S}_x| = \infty \Leftrightarrow \mathcal{S}_x = V$.
2. $\mathscr{C}$ has no infinite cluster *iff* there exists an increasing sequence of stable sets S_n *s.t.* $\lim \uparrow S_n = V$.

Cumulative Merging: Phase transitions

Several natural choices of weights for a locally finite connected graph (V,E):

Cumulative Merging: Phase transitions

Several natural choices of weights for a locally finite connected graph (V,E):

- $(r(x))_{x \in V}$ *i.i.d.* Bernoulli *r.v.* with parameter p.

Cumulative Merging: Phase transitions

Several natural choices of weights for a locally finite connected graph (V,E):

- $(r(x))_{x\in V}$ *i.i.d.* Bernoulli *r.v.* with parameter p.
- $(r(x))_{x\in V}$ *i.i.d.* *r.v.* with law $\lambda \times Z$, $\lambda \geq 0$ deterministic.

Cumulative Merging: Phase transitions

Several natural choices of weights for a locally finite connected graph (V,E):

- $(r(x))_{x\in V}$ *i.i.d.* Bernoulli *r.v.* with parameter p.
- $(r(x))_{x\in V}$ *i.i.d.* *r.v.* with law $\lambda \times Z$, $\lambda \geq 0$ deterministic.
- $\forall x \in V,\ r(x) = \begin{cases} \deg(x) & \text{if } \deg(x) > \Delta; \\ 0 & \text{otherwise.} \end{cases}$

Cumulative Merging: Phase transitions

Several natural choices of weights for a locally finite connected graph (V,E):

- $(r(x))_{x\in V}$ *i.i.d.* Bernoulli *r.v.* with parameter p.
- $(r(x))_{x\in V}$ *i.i.d.* *r.v.* with law $\lambda \times Z$, $\lambda \geq 0$ deterministic.
- $\forall x \in V,\ r(x) = \begin{cases} \deg(x) & \text{if } \deg(x) > \Delta; \\ 0 & \text{otherwise.} \end{cases}$

Critical parameters for the existence of an infinite cluster in $\mathscr{C}(V, E, r)$.

Cumulative Merging: Phase transitions

Several natural choices of weights for a locally finite connected graph (V,E):

- $(r(x))_{x \in V}$ *i.i.d.* Bernoulli *r.v.* with parameter p.
- $(r(x))_{x \in V}$ *i.i.d.* *r.v.* with law $\lambda \times Z$, $\lambda \geq 0$ deterministic.
- $\forall x \in V$, $r(x) = \begin{cases} \deg(x) & \text{if } \deg(x) > \Delta; \\ 0 & \text{otherwise.} \end{cases}$

Critical parameters for the existence of an infinite cluster in $\mathscr{C}(V, E, r)$.

Theorem:

1. CMP on $\mathbb{Z}^d$: $p_c \in (0, 1)$.
2. CMP on $\mathbb{Z}^d$: if $E\left[Z^\beta\right] < \infty$ for $\beta > (4d)^2$, then $\lambda_c \in (0, \infty)$.
3. CMP on d-dimensional Delaunay triangulation or geometric graph: $\Delta_c < \infty$.

Cumulative Merging: Phase transitions

Several natural choices of weights for a locally finite connected graph (V,E):

- $(r(x))_{x\in V}$ *i.i.d.* Bernoulli *r.v.* with parameter p.
- $(r(x))_{x\in V}$ *i.i.d.* *r.v.* with law $\lambda \times Z$, $\lambda \geq 0$ deterministic.
- $\forall x \in V,\ r(x) = \begin{cases} \deg(x) & \text{if } \deg(x) > \Delta; \\ 0 & \text{otherwise.} \end{cases}$

Critical parameters for the existence of an infinite cluster in $\mathscr{C}(V, E, r)$.

Theorem:

1. CMP on $\mathbb{Z}^d$: $p_c \in (0,1)$.
2. CMP on $\mathbb{Z}^d$: if $E\left[Z^\beta\right] < \infty$ for $\beta > (4d)^2$, then $\lambda_c \in (0,\infty)$.
3. CMP on d-dimensional Delaunay triangulation or geometric graph: $\Delta_c < \infty$.

Proofs: Multiscale analysis

Cumulative Merging: Open questions

- Bernoulli CMP on $\mathbb{Z}^d$: **Compute p_c** ?

  (For $d = 1$ we know $p_c \geq 1/2$.)

Cumulative Merging: Open questions

- Bernoulli CMP on $\mathbb{Z}^d$: **Compute** p_c ?

  (For $d = 1$ we know $p_c \geq 1/2$.)

- CMP on $\mathbb{Z}^d$: if $E\left[Z^\beta\right] < \infty$ for $\beta > d + \varepsilon$, then $\lambda_c \in (0, \infty)$?

Cumulative Merging: Open questions

- Bernoulli CMP on $\mathbb{Z}^d$: **Compute** p_c ?

  (For $d = 1$ we know $p_c \geq 1/2$.)

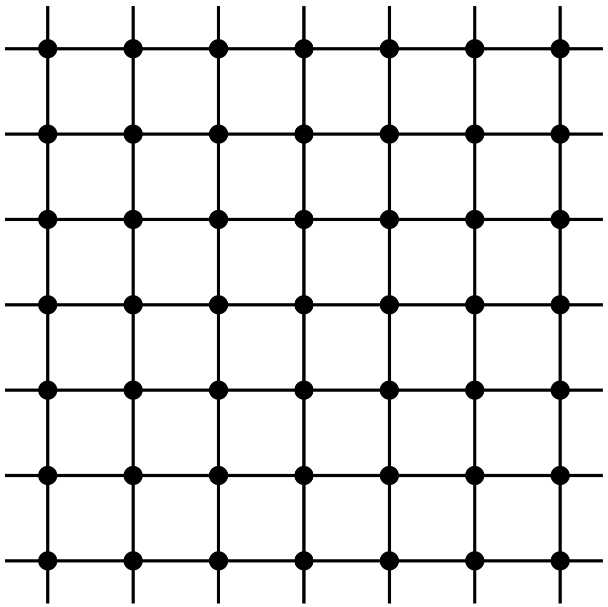
- CMP on $\mathbb{Z}^d$: if $E\left[Z^\beta\right] < \infty$ for $\beta > d + \varepsilon$, then $\lambda_c \in (0, \infty)$?

- Phase transition for CMP on a **tree**?

  1. Binary tree;
  2. Galton-Watson tree with light tails.

The contact process (Susceptible-Infected-Susceptible)

Epidemic model on graphs introduced by [Harris 74]

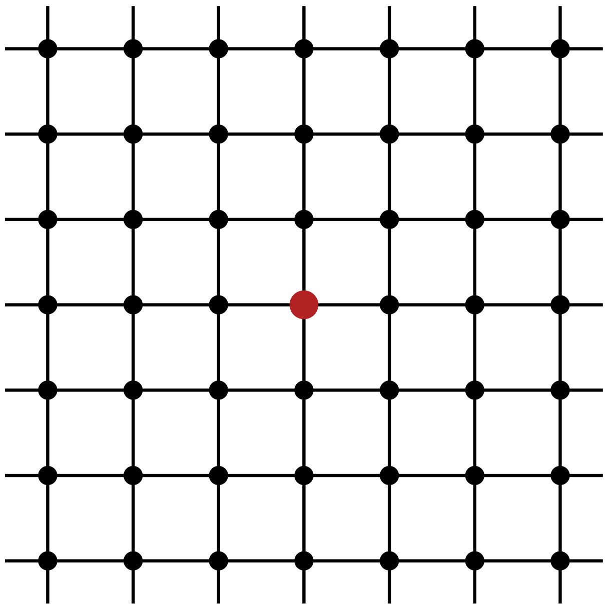


$G = (V, E)$ locally finite graph, $\lambda > 0$.

- Vertices are either **healthy** or **infected**.
- An infected site **recovers** at rate 1.
- A healthy site **is infected** at rate $\lambda\times$ **nbr of infected neighbors**.

The contact process (Susceptible-Infected-Susceptible)

Epidemic model on graphs introduced by [Harris 74]

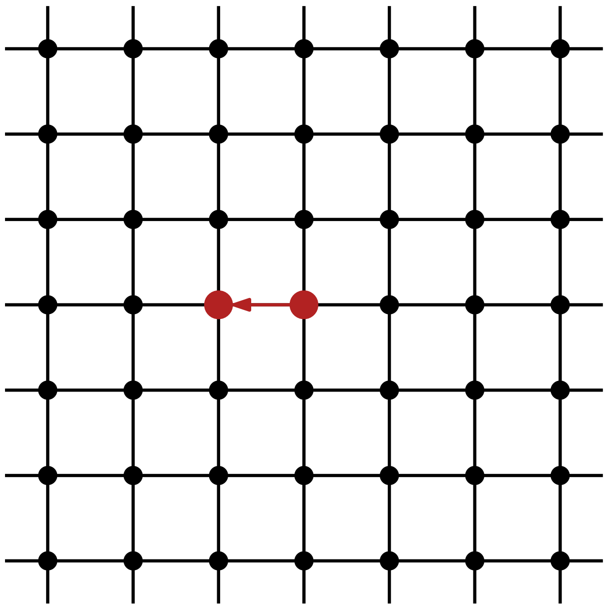


$G = (V, E)$ locally finite graph, $\lambda > 0$.

- Vertices are either **healthy** or **infected**.
- An infected site **recovers** at rate 1.
- A healthy site **is infected** at rate $\lambda\times$ **nbr of infected neighbors**.

The contact process (Susceptible-Infected-Susceptible)

Epidemic model on graphs introduced by [Harris 74]

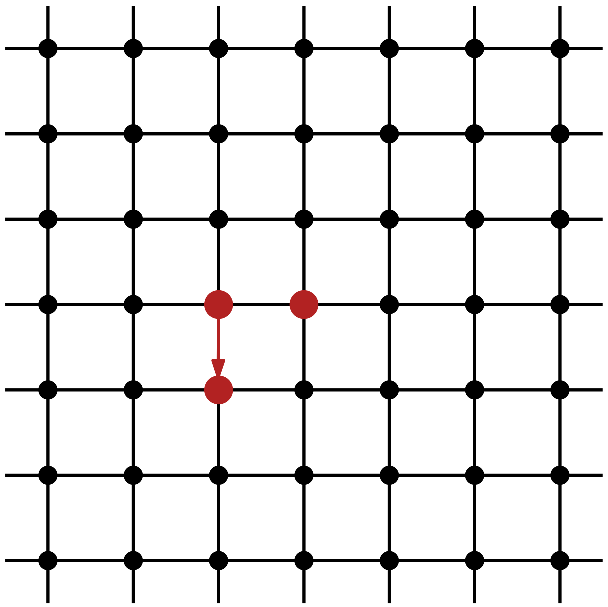


$G = (V, E)$ locally finite graph, $\lambda > 0$.

- Vertices are either **healthy** or **infected**.
- An infected site **recovers** at rate 1.
- A healthy site **is infected** at rate $\lambda \times$ **nbr of infected neighbors**.

The contact process (Susceptible-Infected-Susceptible)

Epidemic model on graphs introduced by [Harris 74]

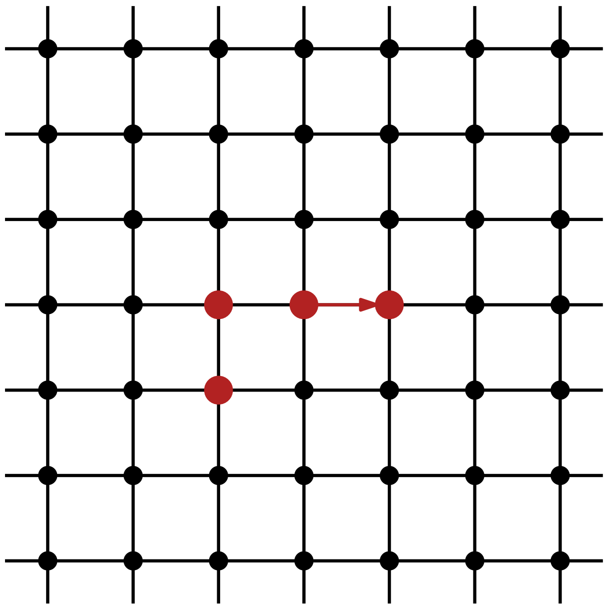


$G = (V, E)$ locally finite graph, $\lambda > 0$.

- Vertices are either **healthy** or **infected**.
- An infected site **recovers** at rate 1.
- A healthy site **is infected** at rate $\lambda \times$ **nbr of infected neighbors**.

The contact process (Susceptible-Infected-Susceptible)

Epidemic model on graphs introduced by [Harris 74]

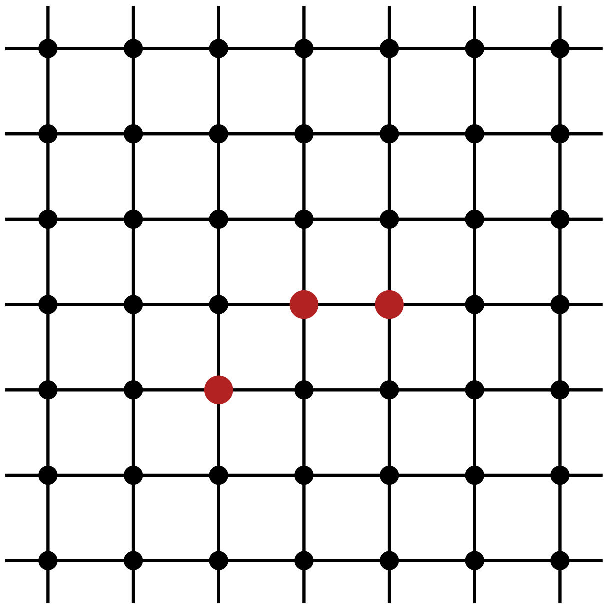


$G = (V, E)$ locally finite graph, $\lambda > 0$.

- Vertices are either **healthy** or **infected**.
- An infected site **recovers** at rate 1.
- A healthy site **is infected** at rate $\lambda \times$ **nbr of infected neighbors**.

The contact process (Susceptible-Infected-Susceptible)

Epidemic model on graphs introduced by [Harris 74]

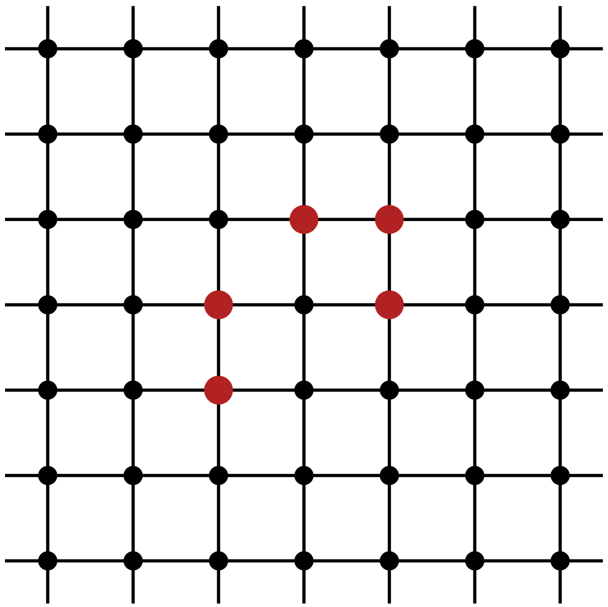


$G = (V, E)$ locally finite graph, $\lambda > 0$.

- Vertices are either **healthy** or **infected**.
- An infected site **recovers** at rate 1.
- A healthy site **is infected** at rate $\lambda\times$ **nbr of infected neighbors**.

The contact process (Susceptible-Infected-Susceptible)

Epidemic model on graphs introduced by [Harris 74]

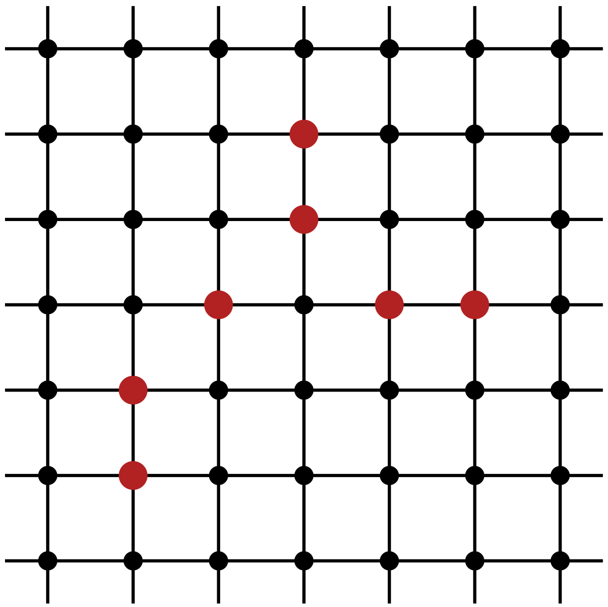


$G = (V, E)$ locally finite graph, $\lambda > 0$.

- Vertices are either **healthy** or **infected**.
- An infected site **recovers** at rate 1.
- A healthy site **is infected** at rate $\lambda \times$ **nbr of infected neighbors**.

The contact process (Susceptible-Infected-Susceptible)

Epidemic model on graphs introduced by [Harris 74]

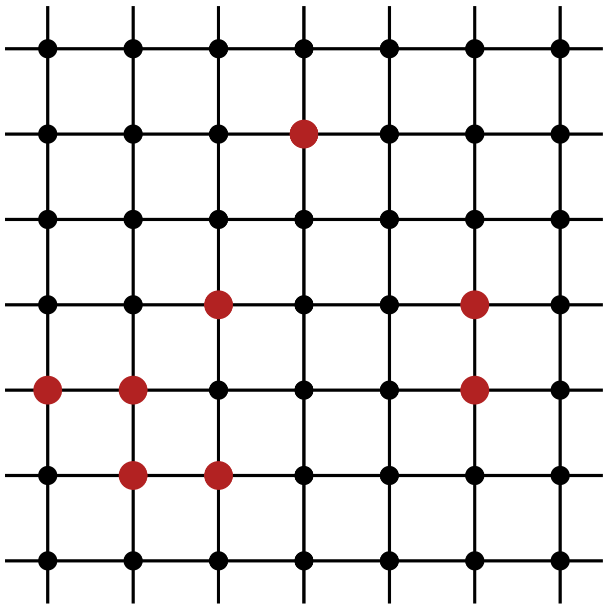


$G = (V, E)$ locally finite graph, $\lambda > 0$.

- Vertices are either **healthy** or **infected**.
- An infected site **recovers** at rate 1.
- A healthy site **is infected** at rate $\lambda \times$ **nbr of infected neighbors**.

The contact process (Susceptible-Infected-Susceptible)

Epidemic model on graphs introduced by [Harris 74]



$G = (V, E)$ locally finite graph, $\lambda > 0$.

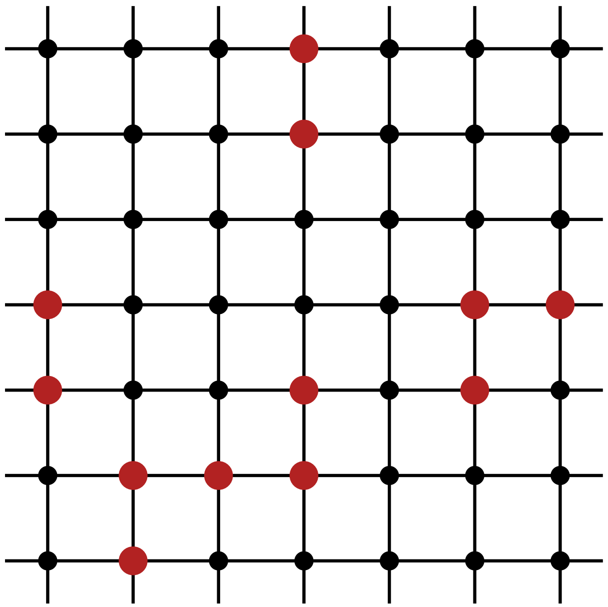
- Vertices are either **healthy** or **infected**.
- An infected site **recovers** at rate 1.
- A healthy site **is infected** at rate $\lambda\times$ **nbr of infected neighbors**.

On an infinite graph, **phase transition**: there is $\lambda_c \in [0, \infty[$ such that

- if $\lambda < \lambda_c$, the infection **dies out** *a.s.*;
- if $\lambda > \lambda_c$, the infection **survives** *a.s.*.

The contact process (Susceptible-Infected-Susceptible)

Epidemic model on graphs introduced by [Harris 74]



$G = (V, E)$ locally finite graph, $\lambda > 0$.

- Vertices are either **healthy** or **infected**.
- An infected site **recovers** at rate 1.
- A healthy site **is infected** at rate $\lambda \times$ **nbr of infected neighbors**.

On an infinite graph, **phase transition**: there is $\lambda_c \in [0, \infty[$ such that

- if $\lambda < \lambda_c$, the infection **dies out** *a.s.*;
- if $\lambda > \lambda_c$, the infection **survives** *a.s.*.

Question: condition on G to ensure $\lambda_c > 0$?

The contact process on a graph with bounded degrees

If G has **bounded** degrees, then $\lambda_c > 0$.

Compare with **branching random walk**:

- **No interaction** between particles;
- particles die at rate 1;
- particles give birth to new particles on neighboring sites at rate λ.

The contact process on a graph with bounded degrees

If G has **bounded** degrees, then $\lambda_c > 0$.

Compare with **branching random walk**:

- **No interaction** between particles;
- particles die at rate 1;
- particles give birth to new particles on neighboring sites at rate λ.

For this process: $\lambda_c \geq 1/d_{max}$.

The contact process on a graph with bounded degrees

If G has **bounded** degrees, then $\lambda_c > 0$.

Compare with **branching random walk**:

- **No interaction** between particles;
- particles die at rate 1;
- particles give birth to new particles on neighboring sites at rate λ.

For this process: $\lambda_c \geq 1/d_{max}$.

Comparison gives nothing for graphs with **unbounded** degrees: BRW survives locally on large degree star-graphs.

The contact process on a graph with bounded degrees

If G has **bounded** degrees, then $\lambda_c > 0$.

Compare with **branching random walk**:

- **No interaction** between particles;
- particles die at rate 1;
- particles give birth to new particles on neighboring sites at rate λ.

For this process: $\lambda_c \geq 1/d_{max}$.

Comparison gives nothing for graphs with **unbounded** degrees: BRW survives locally on large degree star-graphs.

- No other method to prove that contact process dies out.
- No example of graph with unbounded degrees for which we know $\lambda_c > 0$.

Contact process on geometric graphs

Theorem

Let G be either a

- (supercritical) random geometric graph
- Delaunay triangulation

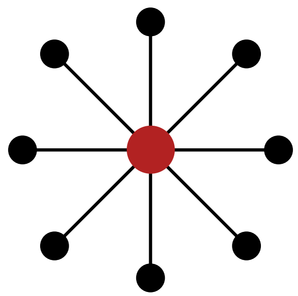
constructed from a Poisson point process on $\mathbb{R}^d$ with Lebesgue intensity. Then one has $\lambda_c > 0$.

Proof:

Criterion on G for $\lambda_c > 0$ in terms of Cumulative Merging Percolation.

Heuristics for the contact process

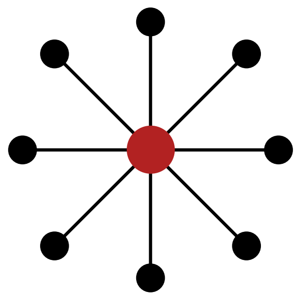
Contact process on a star graph of large degree d:



- start with only ● infected.
- If $\lambda > \lambda_c(d)$, survival time of the process is $\approx \exp(d)$.

Heuristics for the contact process

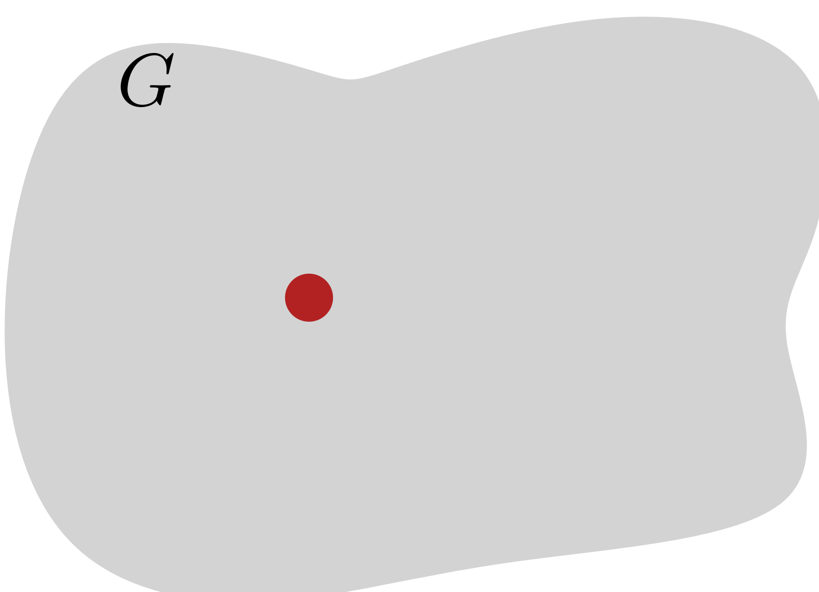
Contact process on a star graph of large degree d:



- start with only ● infected.
- If $\lambda > \lambda_c(d)$, survival time of the process is $\approx \exp(d)$.

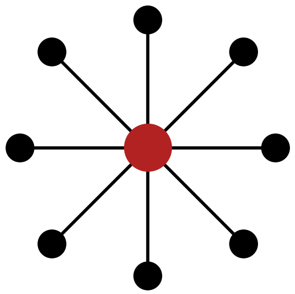
Now fix $\lambda > 0$ and consider contact process on a gaph G *s.t.*

- One vertex (●) has large degree d_0 with $\lambda > \lambda_c(d_0)$;
- all other vertices have small degrees d with $\lambda \ll \lambda_c(d)$.



Heuristics for the contact process

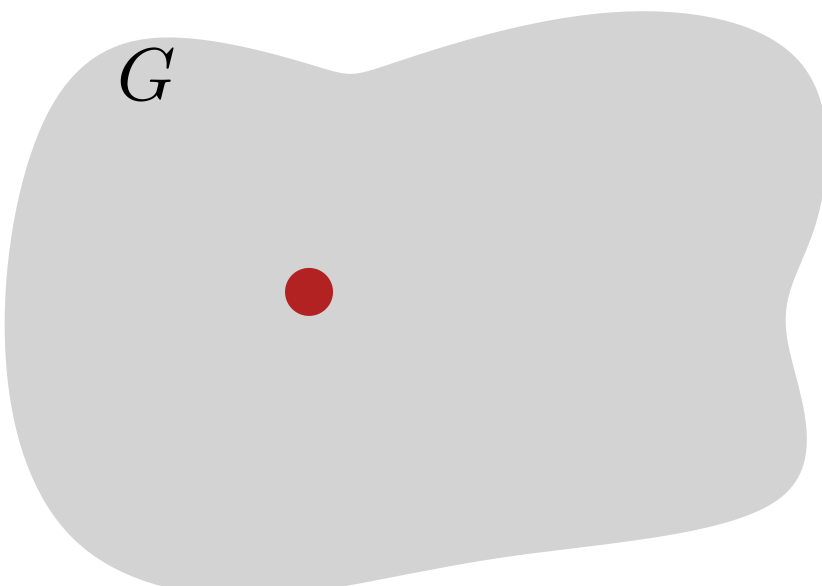
Contact process on a star graph of large degree d:



- start with only ● infected.
- If $\lambda > \lambda_c(d)$, survival time of the process is $\approx \exp(d)$.

Now fix $\lambda > 0$ and consider contact process on a gaph G *s.t.*

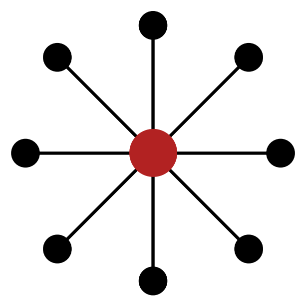
- One vertex (●) has large degree d_0 with $\lambda > \lambda_c(d_0)$;
- all other vertices have small degrees d with $\lambda \ll \lambda_c(d)$.



- Start with only ● infected.
- Force ● to stay infected a time $\exp(d_0)$.
- After that time, force the whole star around ● to recover.

Heuristics for the contact process

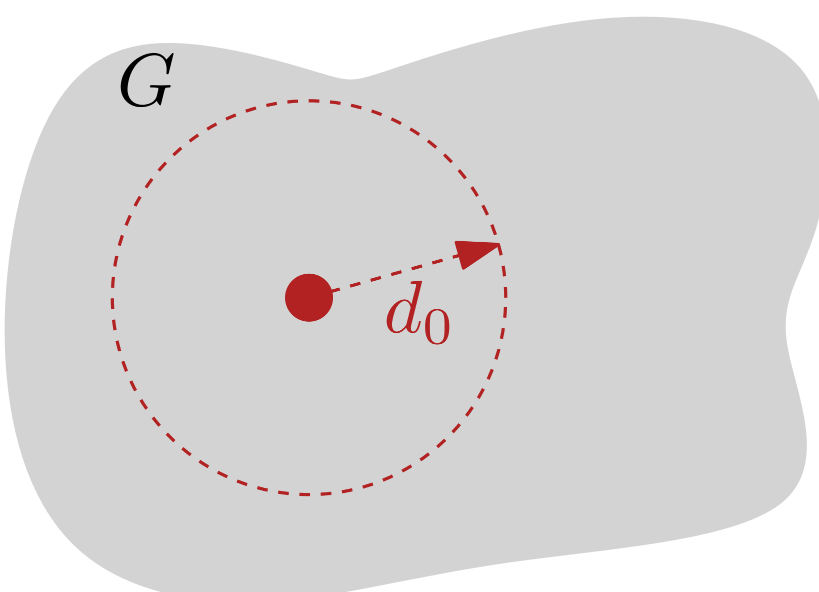
Contact process on a star graph of large degree d:



- start with only ● infected.
- If $\lambda > \lambda_c(d)$, survival time of the process is $\approx \exp(d)$.

Now fix $\lambda > 0$ and consider contact process on a gaph G *s.t.*

- One vertex (●) has large degree d_0 with $\lambda > \lambda_c(d_0)$;
- all other vertices have small degrees d with $\lambda \ll \lambda_c(d)$.

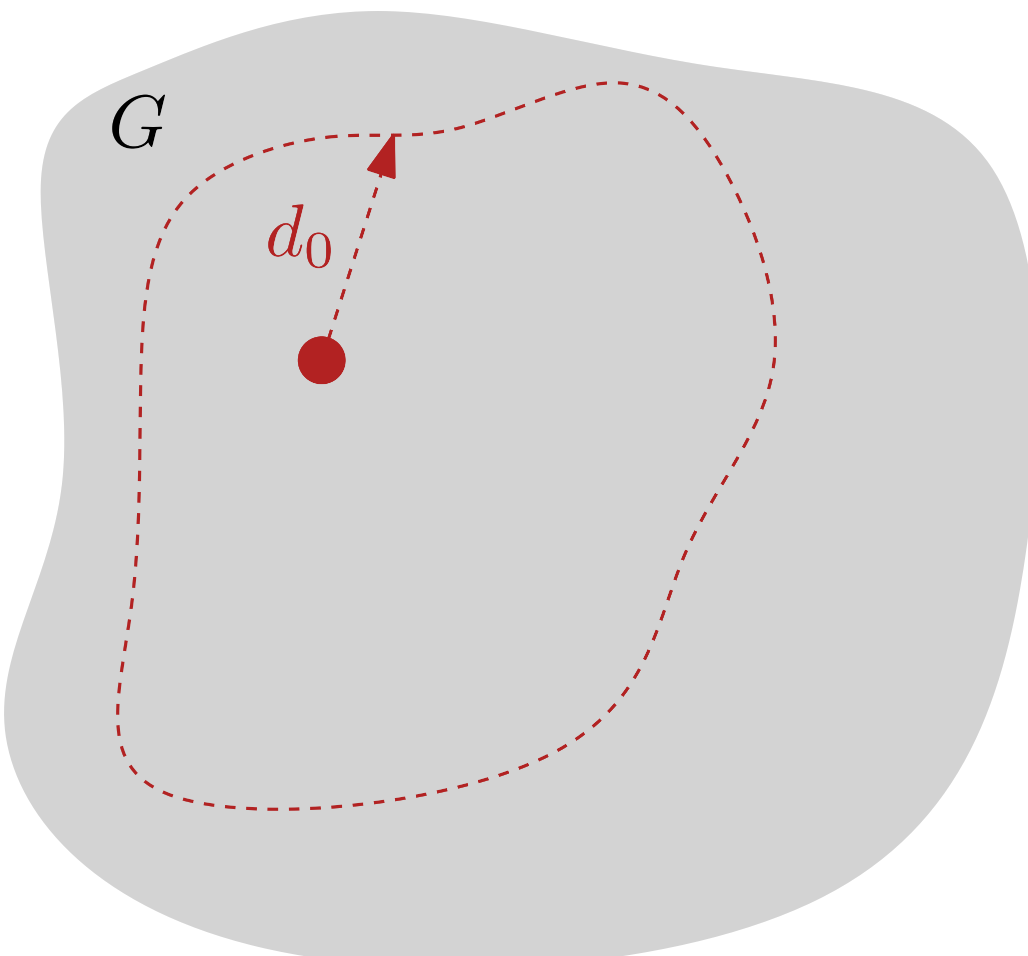


- Start with only ● infected.
- Force ● to stay infected a time $\exp(d_0)$.
- After that time, force the whole star around ● to recover.

Maximal distance reached by infection is $\approx d_0$.

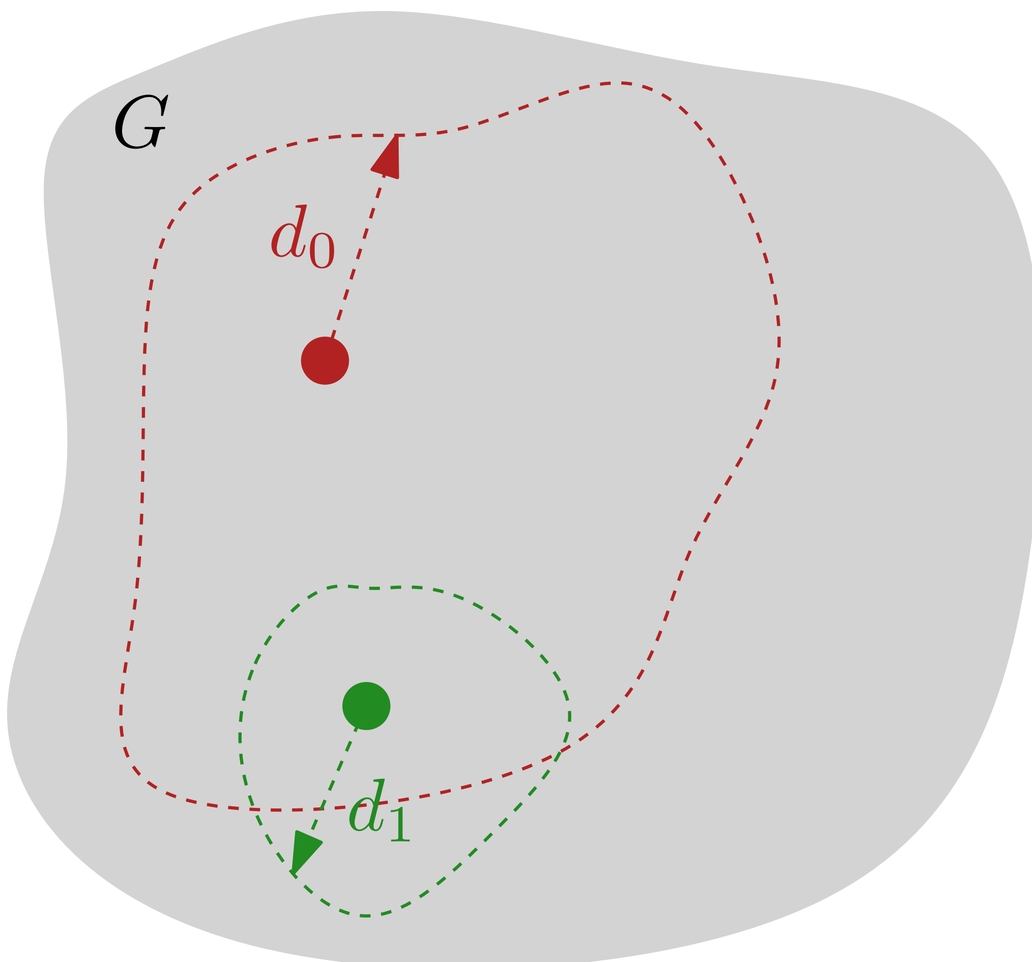
Heuristics for the contact process

Same as before for ● and start of the infection.



Heuristics for the contact process

Same as before for ● and start of the infection.

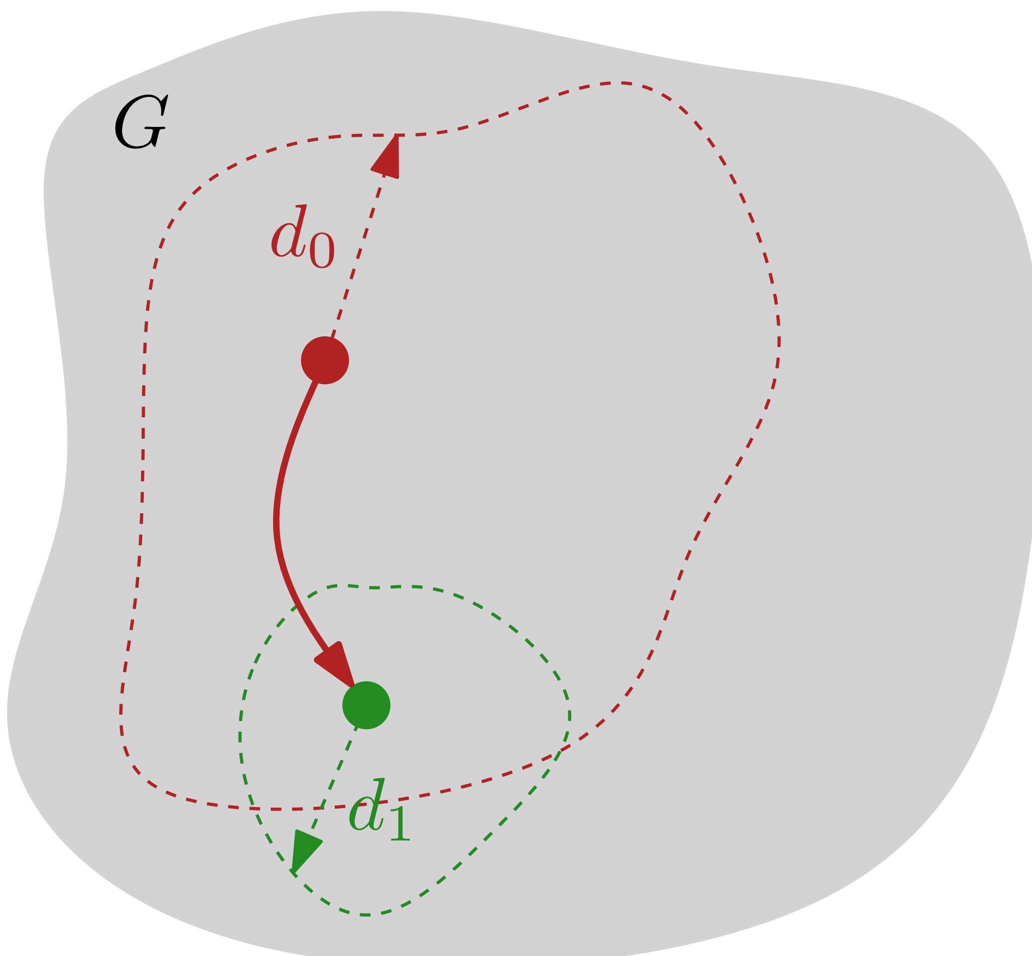


In addition suppose:

- At distance $< d_0$ from ● , there is another vertex (●) with large degree d_1 *s.t.* $\lambda > \lambda_c(d_1)$.
- Suppose also $d_1 \ll d_0$.

Heuristics for the contact process

Same as before for ● and start of the infection.



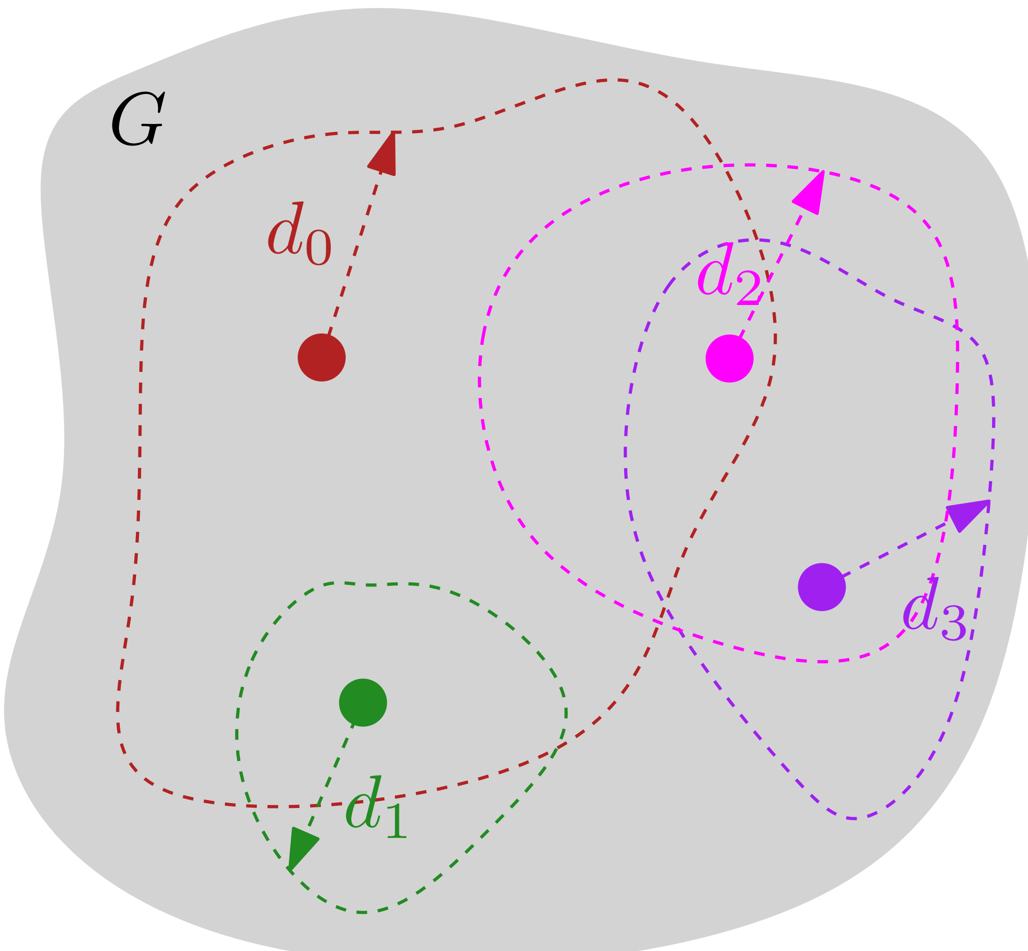
In addition suppose:

- At distance $< d_0$ from ● , there is another vertex (●) with large degree d_1 *s.t.* $\lambda > \lambda_c(d_1)$.
- Suppose also $d_1 \ll d_0$.

● cannot send infections to ● and the survival time of the process is $\approx \exp(d_0) + \exp(d_1) \approx \exp(d_0)$.

Heuristics for the contact process

Same as before for ● and start of the infection.

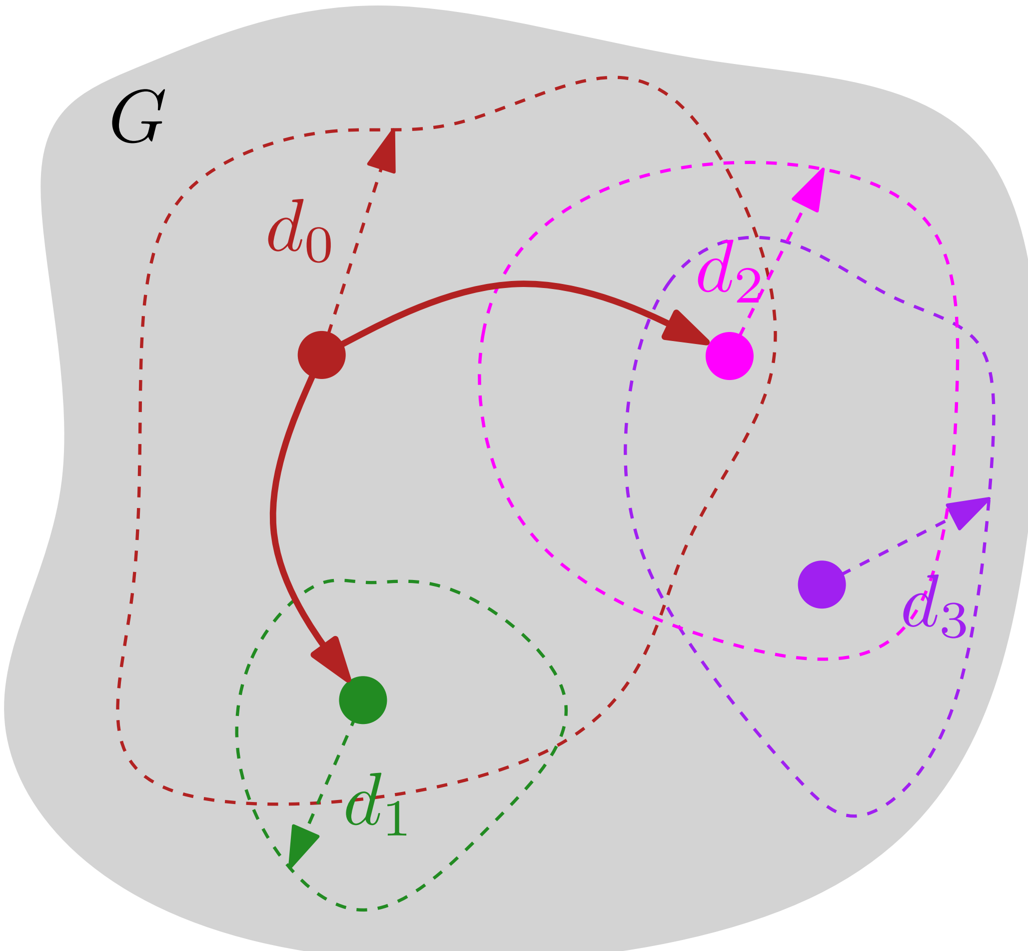


Now suppose:

- another vertex (●) with large degree d_2 *s.t.* $\lambda > \lambda_c(d_2)$.
- And a last one (●) with large degree d_3 *s.t.* $\lambda > \lambda_c(d_3)$.
- dist(● , ●) $< d_2 \wedge d_3$

Heuristics for the contact process

Same as before for ● and start of the infection.

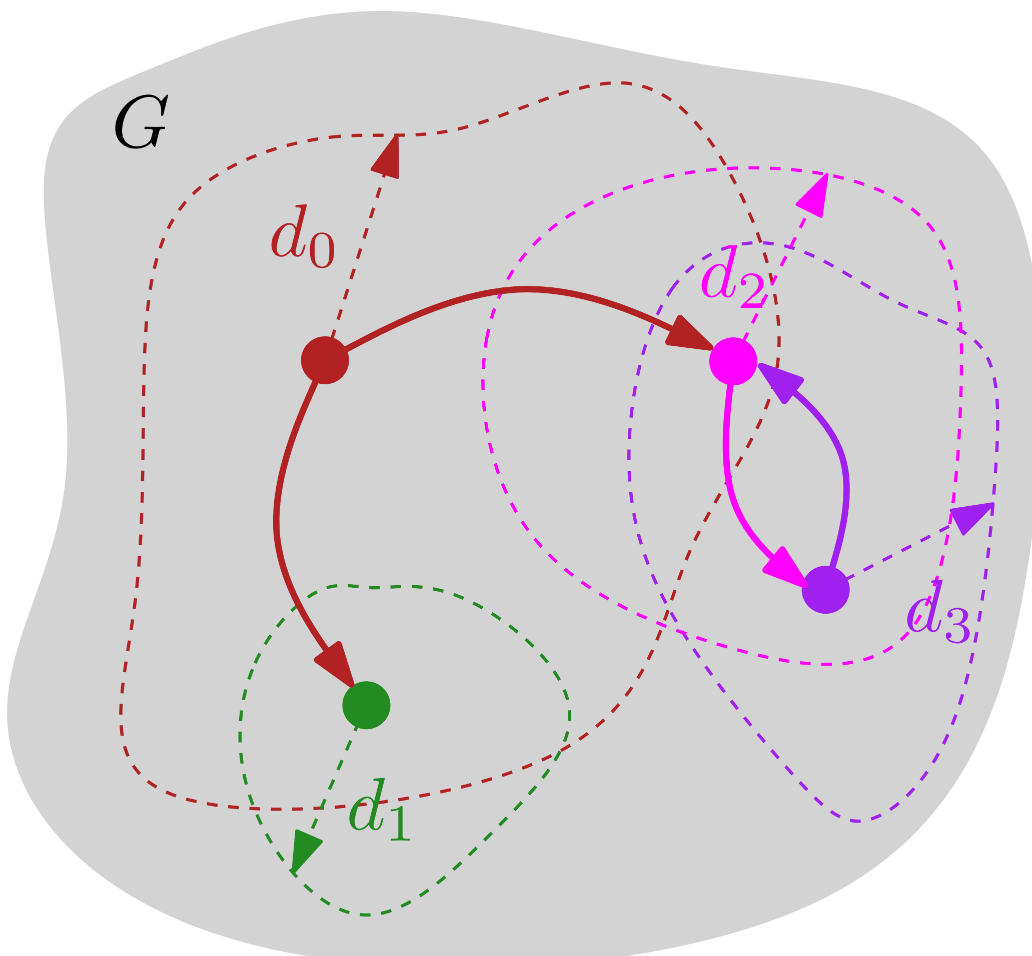


Now suppose:

- another vertex (●) with large degree d_2 *s.t.* $\lambda > \lambda_c(d_2)$.
- And a last one (●) with large degree d_3 *s.t.* $\lambda > \lambda_c(d_3)$.
- dist(●, ●) $< d_2 \wedge d_3$

Heuristics for the contact process

Same as before for ● and start of the infection.



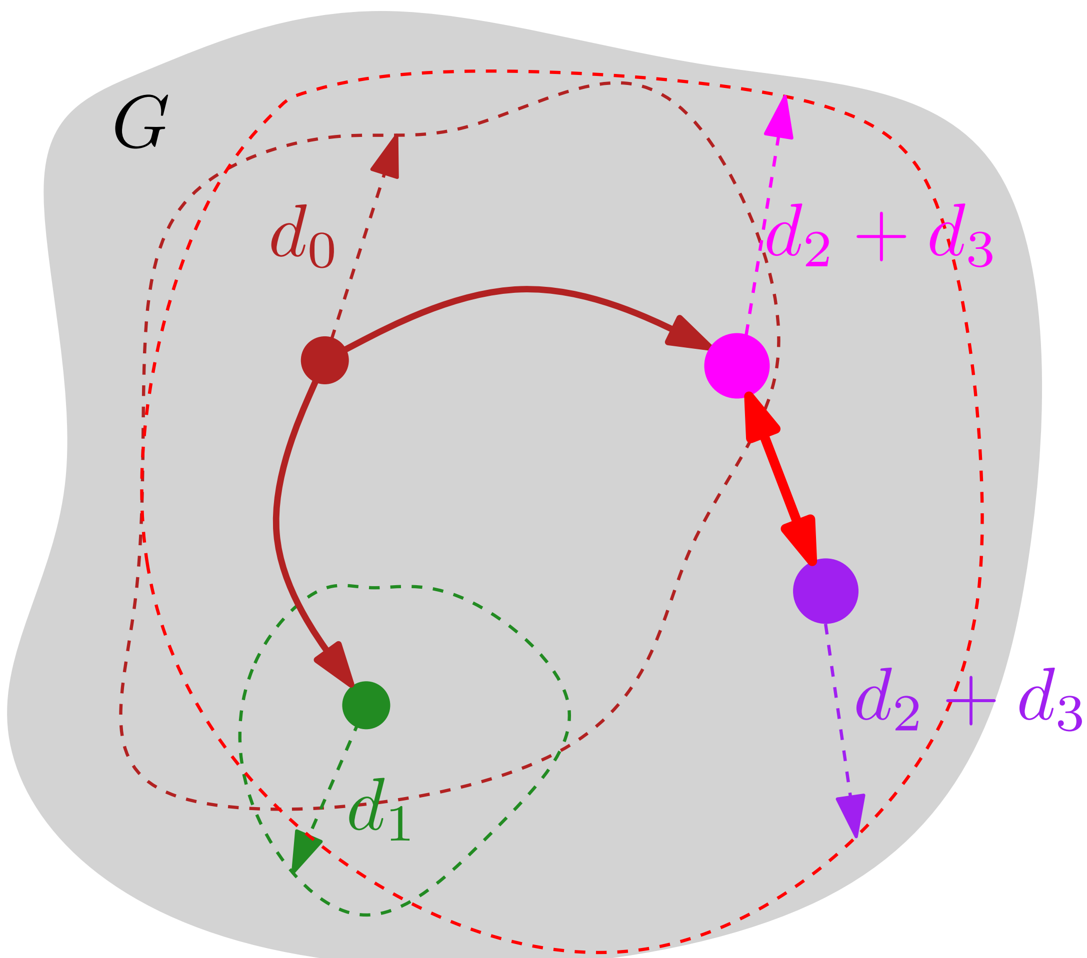
Now suppose:

- another vertex (●) with large degree d_2 *s.t.* $\lambda > \lambda_c(d_2)$.
- And a last one (●) with large degree d_3 *s.t.* $\lambda > \lambda_c(d_3)$.
- dist(●, ●) $< d_2 \wedge d_3$

● and ● interact and their combined survival time is $\approx \exp(d_2) \times \exp(d_3) = \exp(d_2 + d_3)$

Heuristics for the contact process

Same as before for ● and start of the infection.



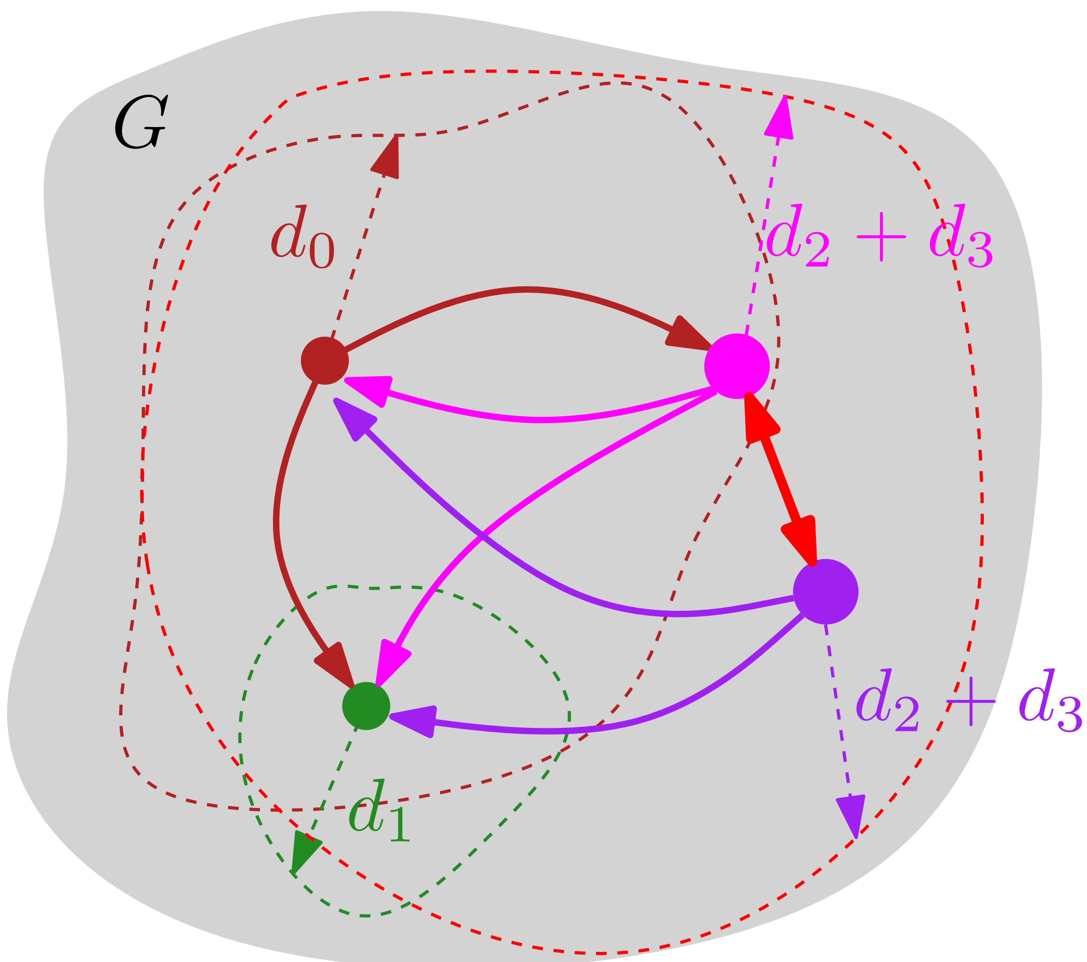
Now suppose:

- another vertex (●) with large degree d_2 *s.t.* $\lambda > \lambda_c(d_2)$.
- And a last one (●) with large degree d_3 *s.t.* $\lambda > \lambda_c(d_3)$.
- dist(●, ●) $< d_2 \wedge d_3$

● and ● interact and their combined survival time is $\approx \exp(d_2) \times \exp(d_3) = \exp(d_2 + d_3)$

Heuristics for the contact process

Same as before for ● and start of the infection.



Now suppose:

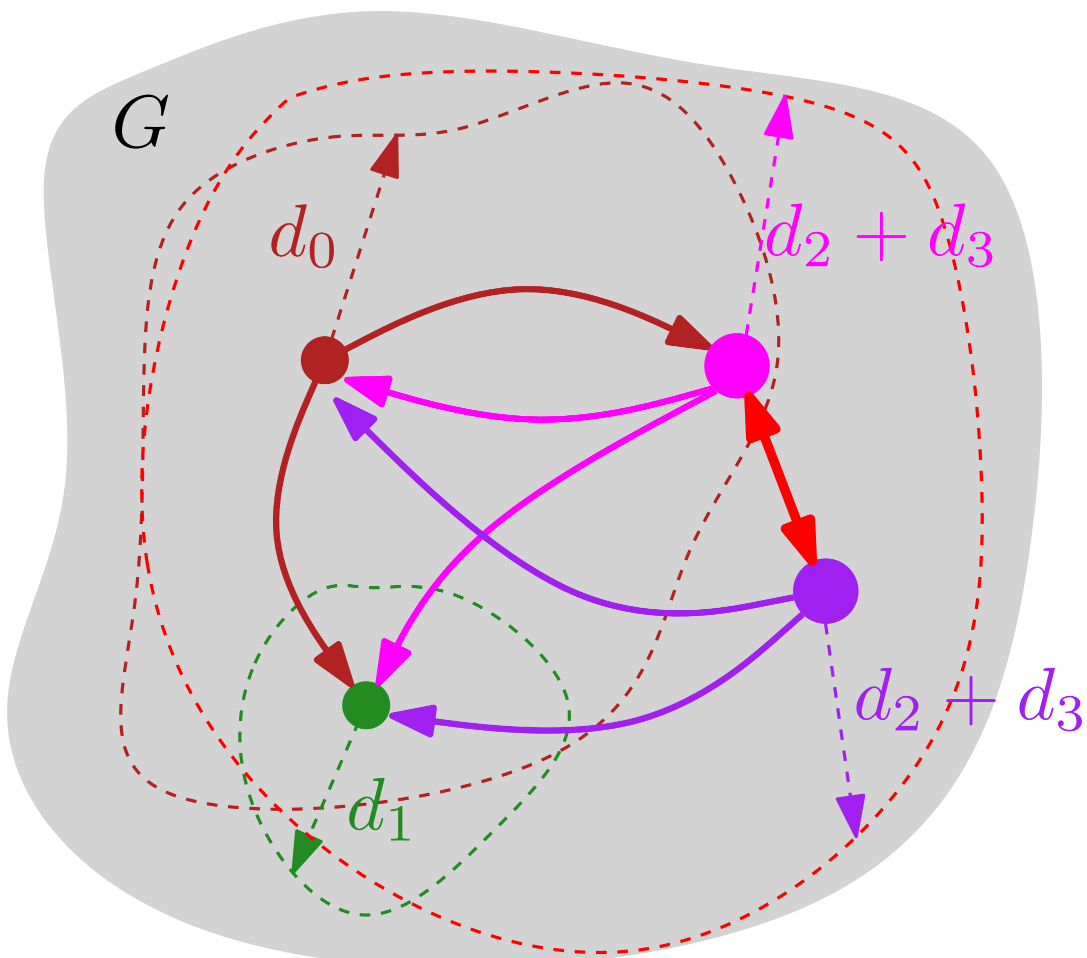
- another vertex (●) with large degree d_2 *s.t.* $\lambda > \lambda_c(d_2)$.
- And a last one (●) with large degree d_3 *s.t.* $\lambda > \lambda_c(d_3)$.
- dist(● , ●) $< d_2 \wedge d_3$

● and ● interact and their combined survival time is $\approx \exp(d_2) \times \exp(d_3) = \exp(d_2 + d_3)$

⟶ Now ● and ● can reach ● or ● !

Heuristics for the contact process

Same as before for ● and start of the infection.



Now suppose:

- another vertex (●) with large degree d_2 *s.t.* $\lambda > \lambda_c(d_2)$.
- And a last one (●) with large degree d_3 *s.t.* $\lambda > \lambda_c(d_3)$.
- dist(●, ●) $< d_2 \wedge d_3$

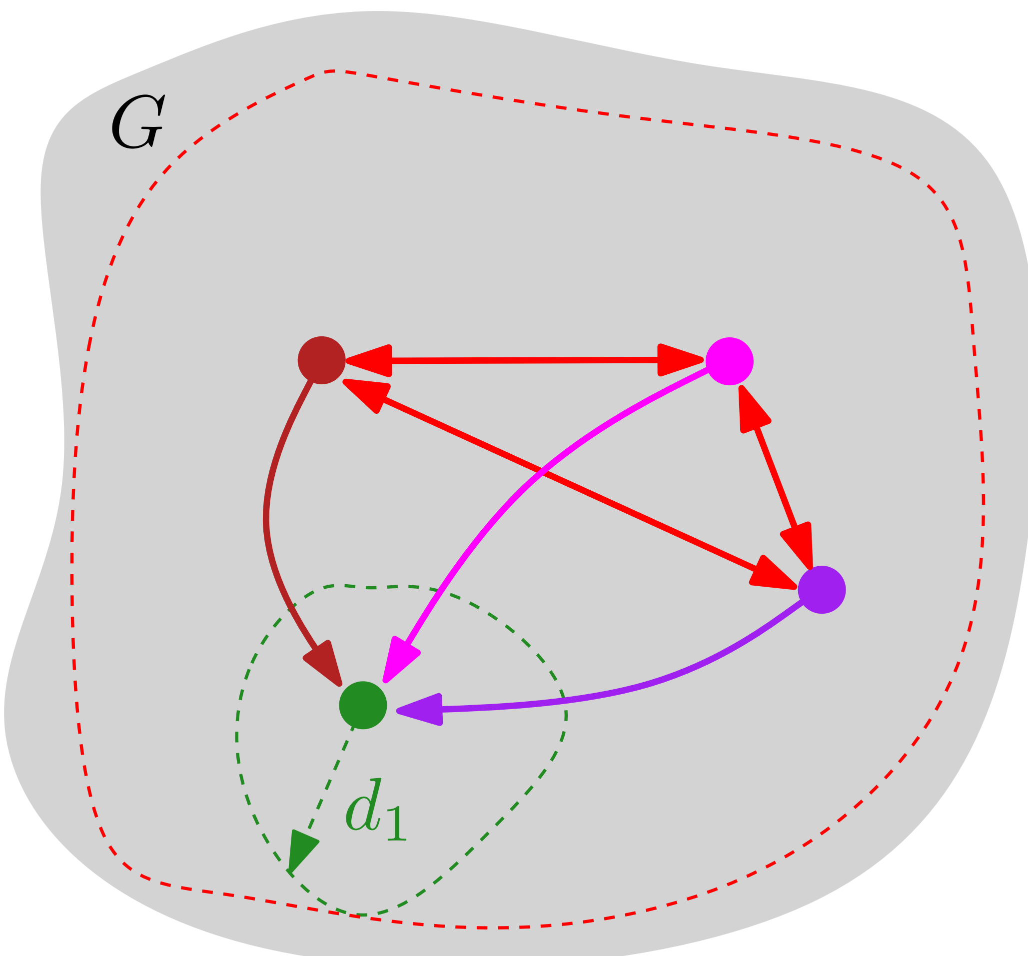
● and ● interact and
their combined survival time is
$\approx \exp(d_2) \times \exp(d_3) = \exp(d_2 + d_3)$

→ Now ● and ● can reach ● or ● !

→ ●, ● and ● are all in interaction.

Heuristics for the contact process

Same as before for ● and start of the infection.



Now suppose:

- another vertex (●) with large degree d_2 *s.t.* $\lambda > \lambda_c(d_2)$.
- And a last one (●) with large degree d_3 *s.t.* $\lambda > \lambda_c(d_3)$.
- dist(● , ●) $< d_2 \wedge d_3$

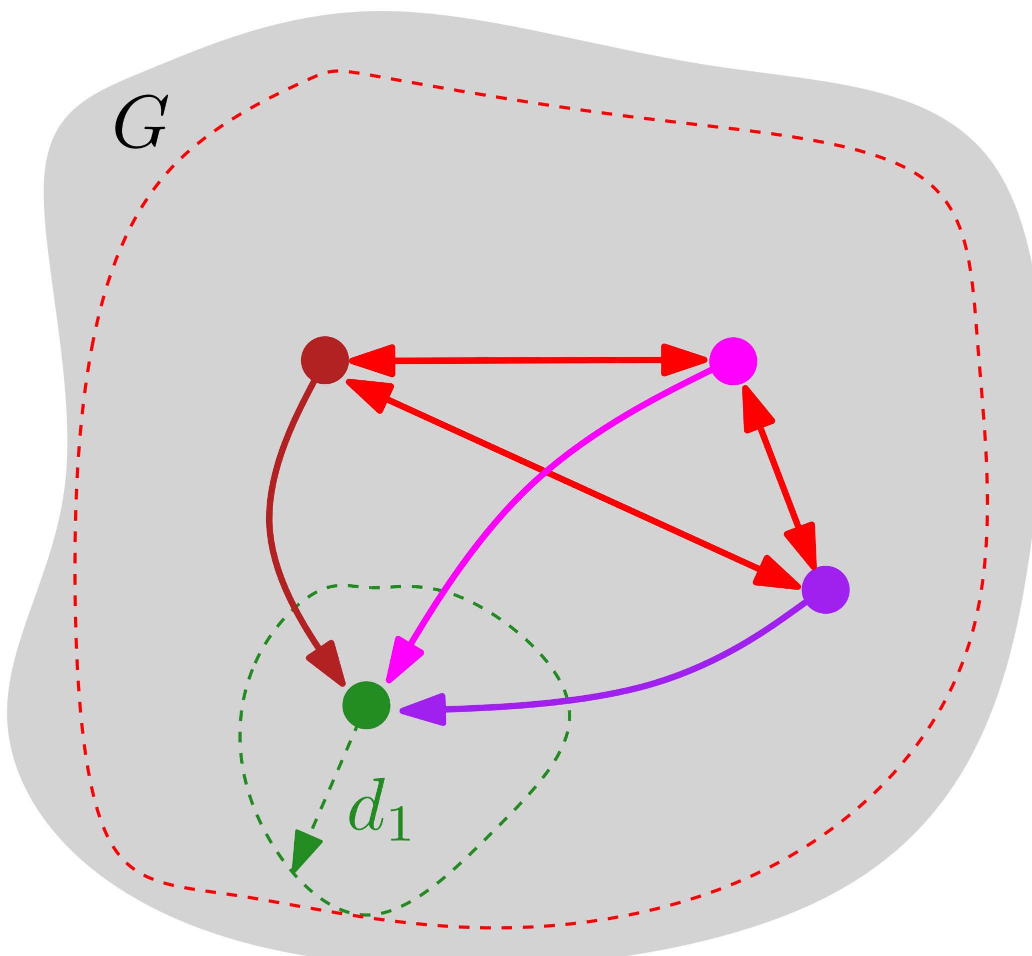
● and ● interact and
their combined survival time is
$\approx \exp(d_2) \times \exp(d_3) = \exp(d_2 + d_3)$

⟶ Now ● and ● can reach ● or ● !

⟶ ●, ● and ● are all in interaction.

Heuristics for the contact process

Same as before for ● and start of the infection.



Now suppose:

- another vertex (●) with large degree d_2 *s.t.* $\lambda > \lambda_c(d_2)$.
- And a last one (●) with large degree d_3 *s.t.* $\lambda > \lambda_c(d_3)$.
- dist(●, ●) $< d_2 \wedge d_3$

● and ● interact and
their combined survival time is
$\approx \exp(d_2) \times \exp(d_3) = \exp(d_2 + d_3)$

⟶ Now ● and ● can reach ● or ● !

⟶ ●, ● and ● are all in interaction.

⟶ ● still cannot reach the other 3 vertices to interact.

Cumulative Merging: Link with the contact process

Need to change the definition of admissible partitions:

$\mathcal{P}$ is **admissible** *iff* $\forall A, B \in \mathcal{P}$

$$d_G(A, B) > \quad r(A) \wedge r(B) \qquad .$$

Cumulative Merging: Link with the contact process

Need to change the definition of admissible partitions:

Fix $\alpha > 1$, $\mathcal{P}$ is **admissible** *iff* $\forall A, B \in \mathcal{P}$

$$d_G(A, B) > (r(A) \wedge r(B))^{\alpha}\,.$$

Cumulative Merging: Link with the contact process

Need to change the definition of admissible partitions:

Fix $\alpha > 1$, $\mathcal{P}$ is **admissible** *iff* $\forall A, B \in \mathcal{P}$

$$d_G(A, B) > (r(A) \wedge r(B))^{\alpha}.$$

Theorem:
Let $G = (V, E)$ be **any** locally finite graph. Suppose that, for $\alpha > 2.5$, CMP on G with weights given by:

$$\forall x \in V,\ r(x) = \begin{cases} \deg(x) & \text{if } \deg(x) > \Delta; \\ 0 & \text{otherwise.} \end{cases}$$

has a non-trivial phase transition (*i.e.* $\Delta_c < \infty$).

Then the contact process on G has a non trivial phase transition (*i.e.* it dies out for small infection rates).

Cumulative Merging: Link with the contact process

Need to change the definition of admissible partitions:

Fix $\alpha > 1$, $\mathcal{P}$ is **admissible** *iff* $\forall A, B \in \mathcal{P}$

$$d_G(A,B) > \left(r(A) \wedge r(B)\right)^{\alpha}.$$

Theorem:
Let $G = (V,E)$ be **any** locally finite graph. Suppose that, for $\alpha > 2.5$, CMP on G with weights given by:

$$\forall x \in V,\, r(x) = \begin{cases} \deg(x) & \text{if } \deg(x) > \Delta; \\ 0 & \text{otherwise.} \end{cases}$$

has a non-trivial phase transition (*i.e.* $\Delta_c < \infty$).
Then the contact process on G has a non trivial phase transition (*i.e.* it dies out for small infection rates).

Question: true for $\alpha = 1$?

Conference **Dynamics on random graphs and random planar maps**

October 23 to 27, 2017 in Marseille France

Org. LM, Pierre Nolin, Bruno Schapira and Arvind Singh

Thank you for your attention!