Monotony Properties of Connected Visible Graph Searching

Pierre Fraigniaud[1] Nicolas Nisse[2]

CNRS, LRI, Université Paris-Sud, France.

LRI, Université Paris-Sud, France.

WG 06, Bergen, June 2006

Graph Searching

Goal

In an undirected simple graph, stand

- an invisible omniscient arbitrary fast fugitive;
- a team of searchers;

To find a strategy that catch the fugitive
using the fewest searchers as possible.

Motivations

- network security, speleological rescue,...
- problem related to well known graphs'parameters: treewidth and pathwidth.

Graph Searching

Goal

In an undirected simple graph, stand

- an invisible omniscient arbitrary fast fugitive;
- a team of searchers;

To find a strategy that catch the fugitive
using the fewest searchers as possible.

Motivations

- network security, speleological rescue,...
- problem related to well known graphs'parameters: treewidth and pathwidth.

Search Strategy, Parson. [GTC,1978]
Variant of Kirousis and Papadimitriou. [TCS,86]

Sequence of two basic operations,...

1. Place a searcher at a vertex of the graph;
2. Remove a searcher from a vertex of the graph.

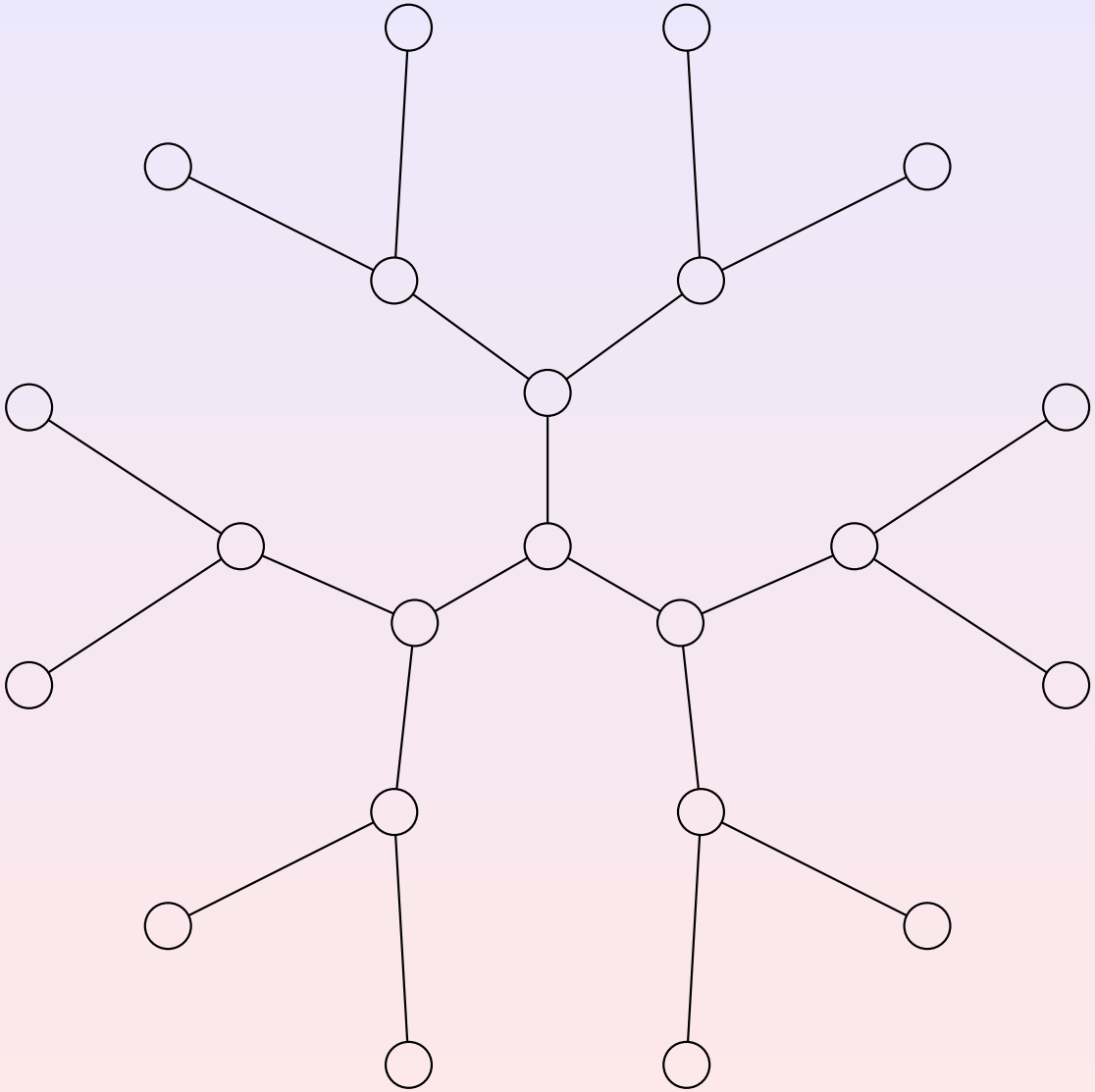
...that must result in catching the fugitive

The fugitive is caugth when it occupies (or crosses) a vertex occupied by a searcher.

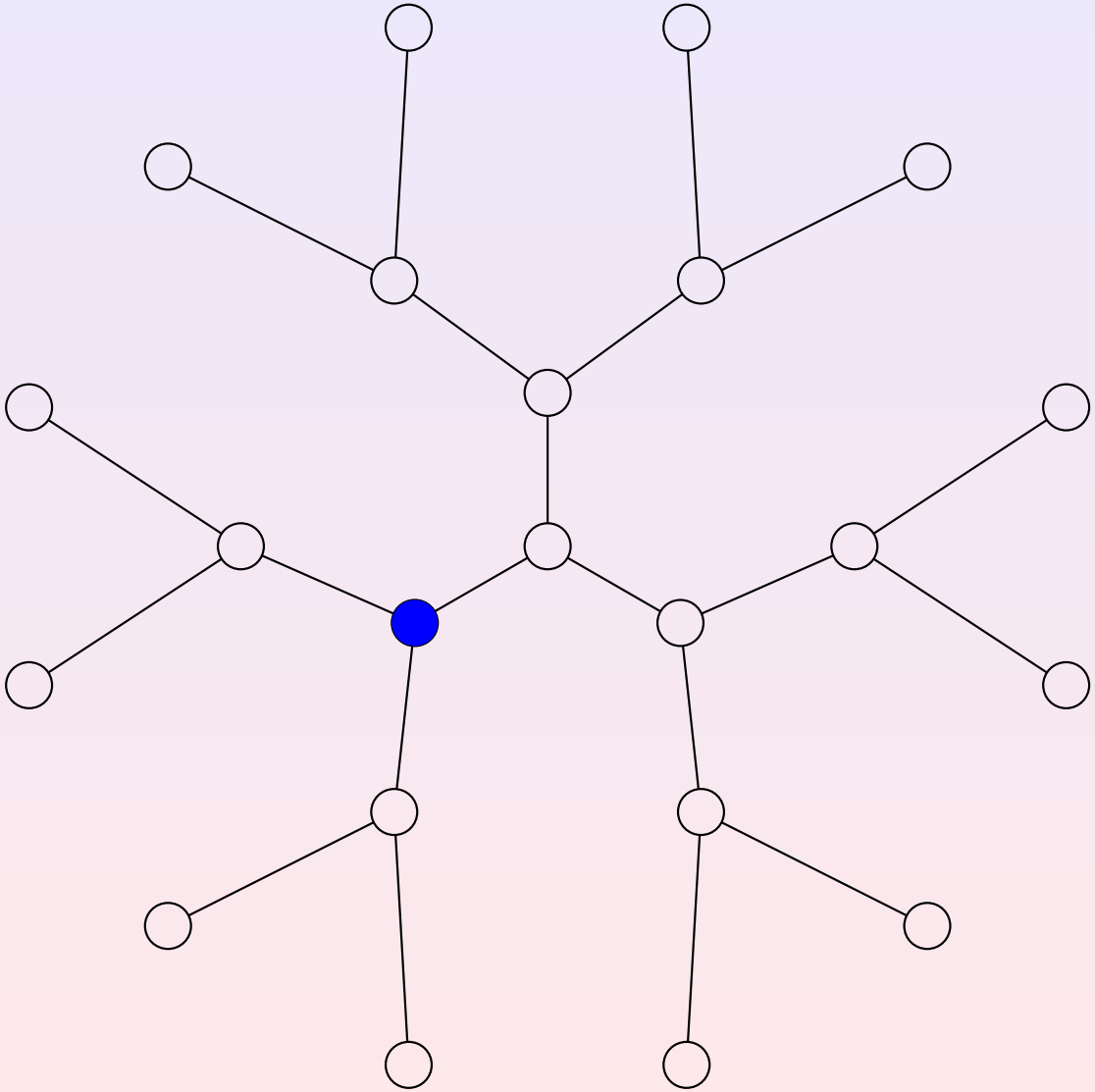
We want to minimize the number of searchers.

Let $s(G)$ be the smallest number of searchers needed to catch an invisible fugitive in a graph G.

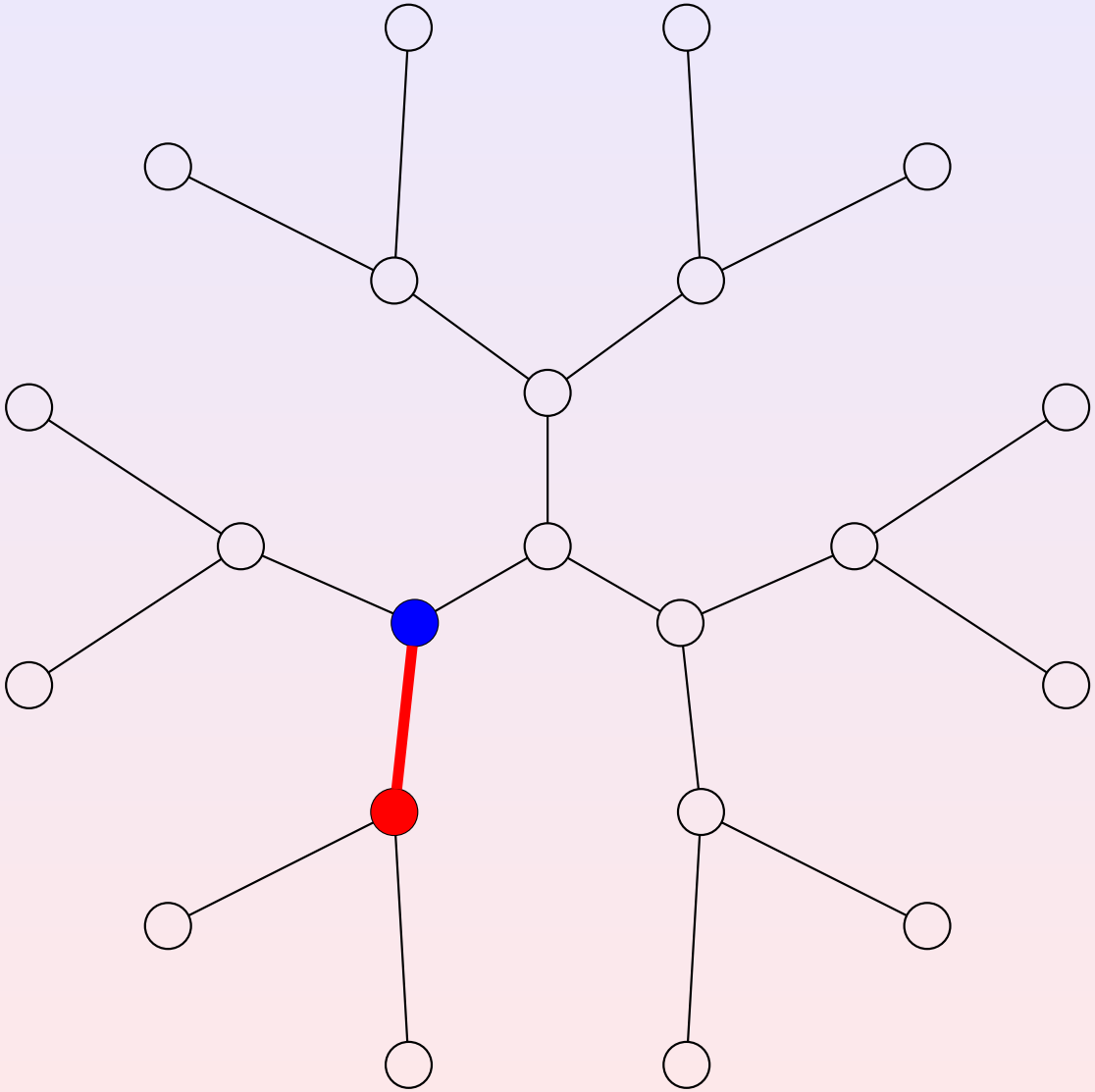
A simple example



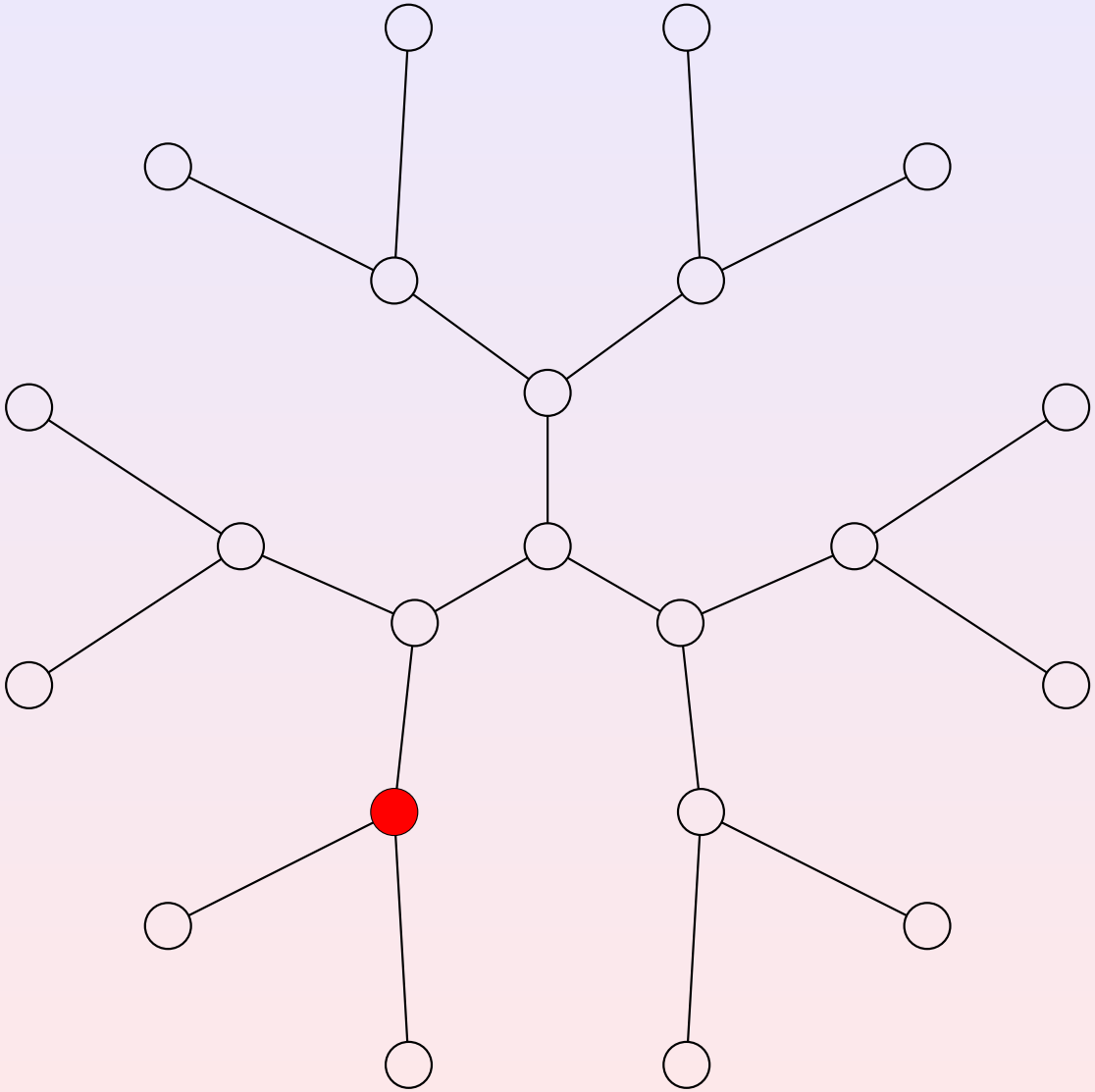
A simple example



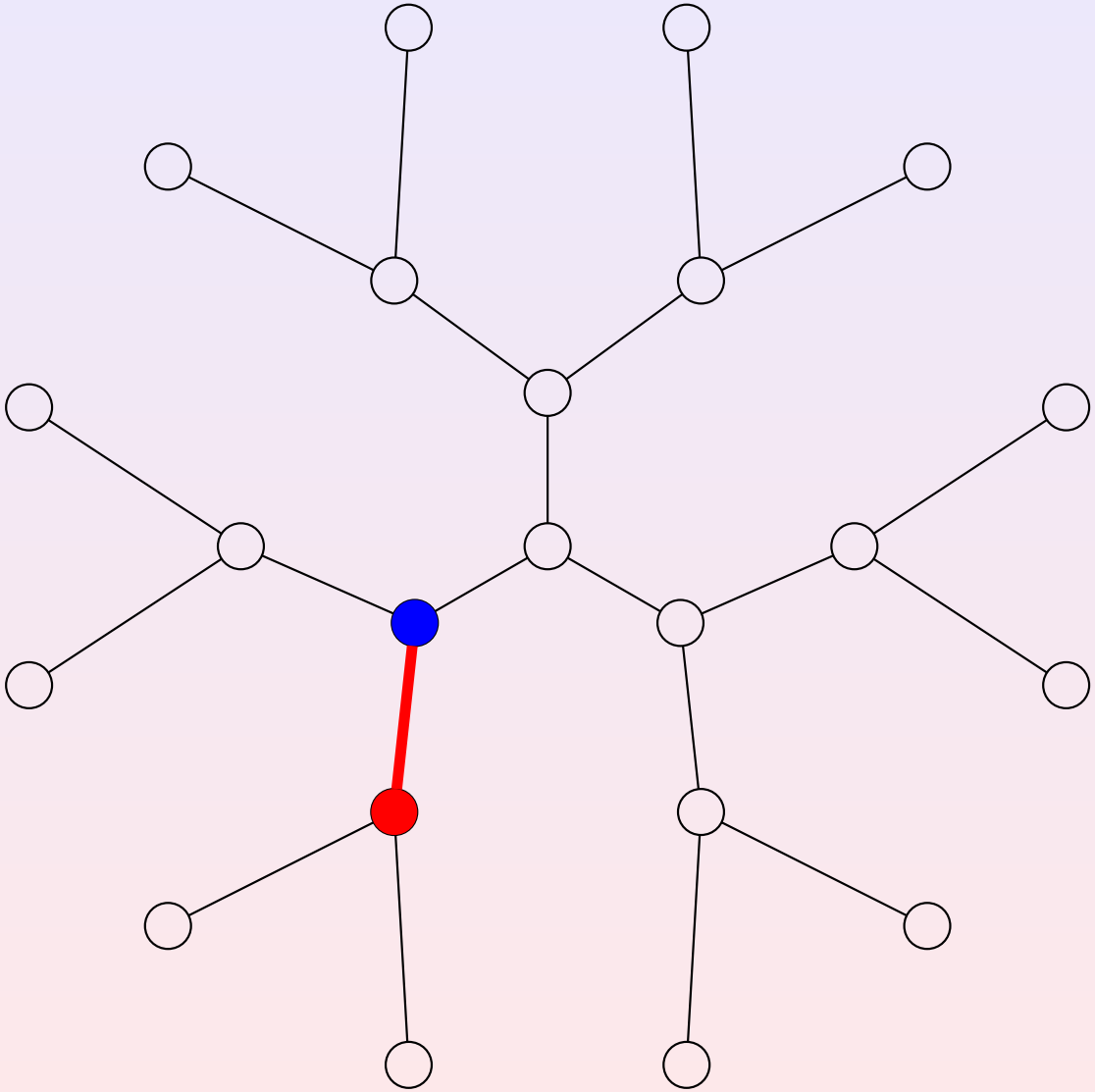
A simple example



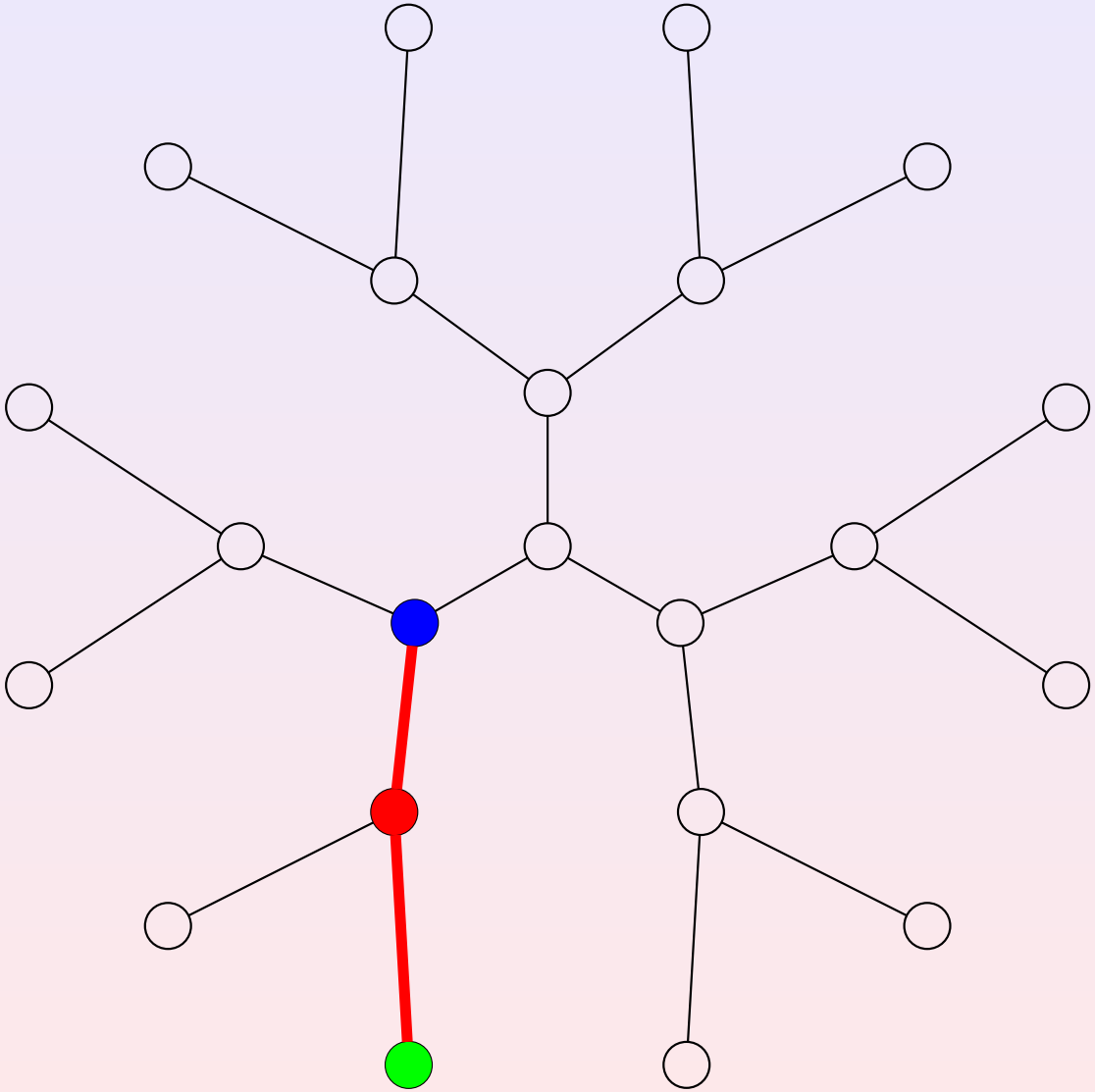
A simple example



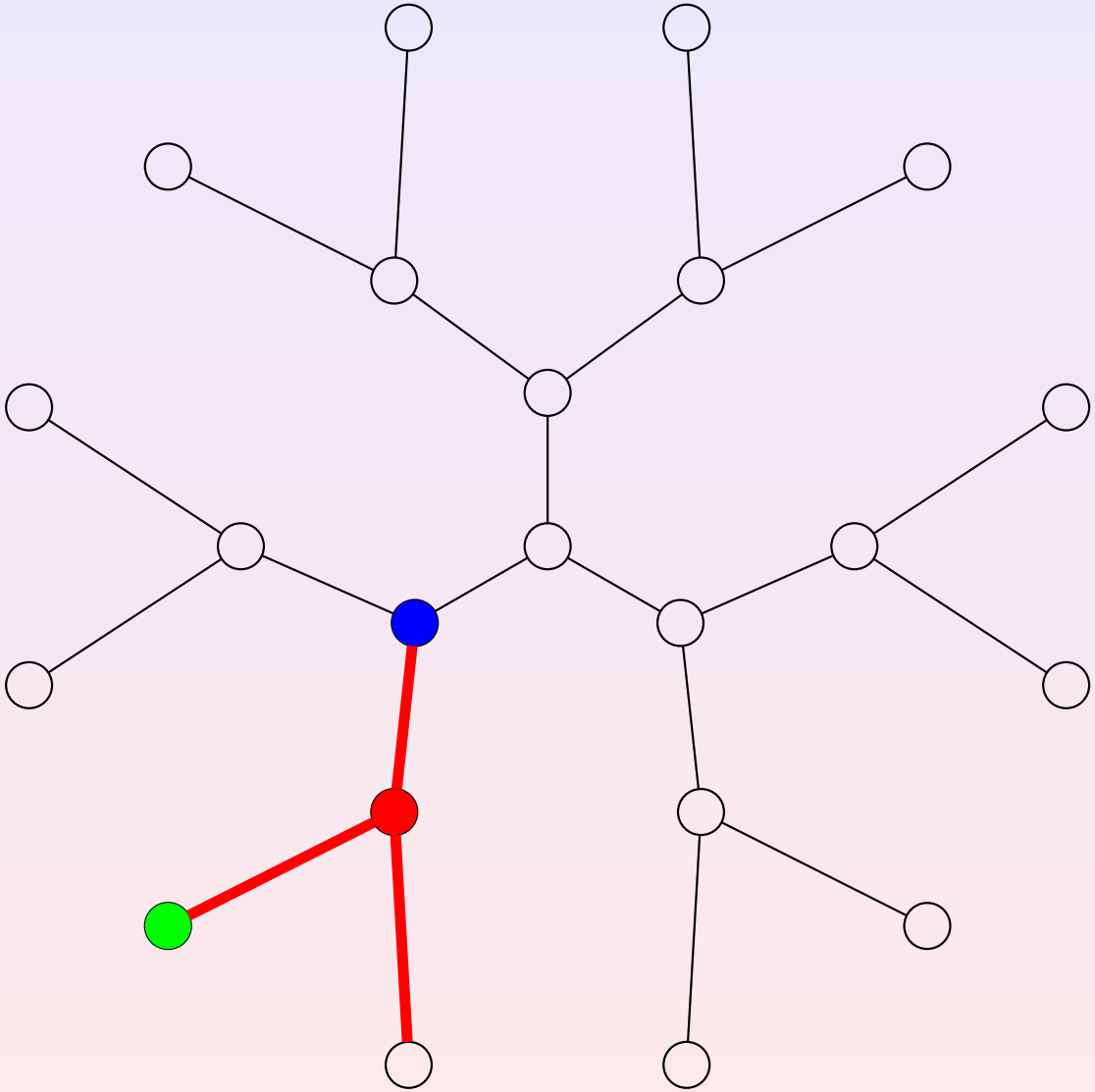
A simple example



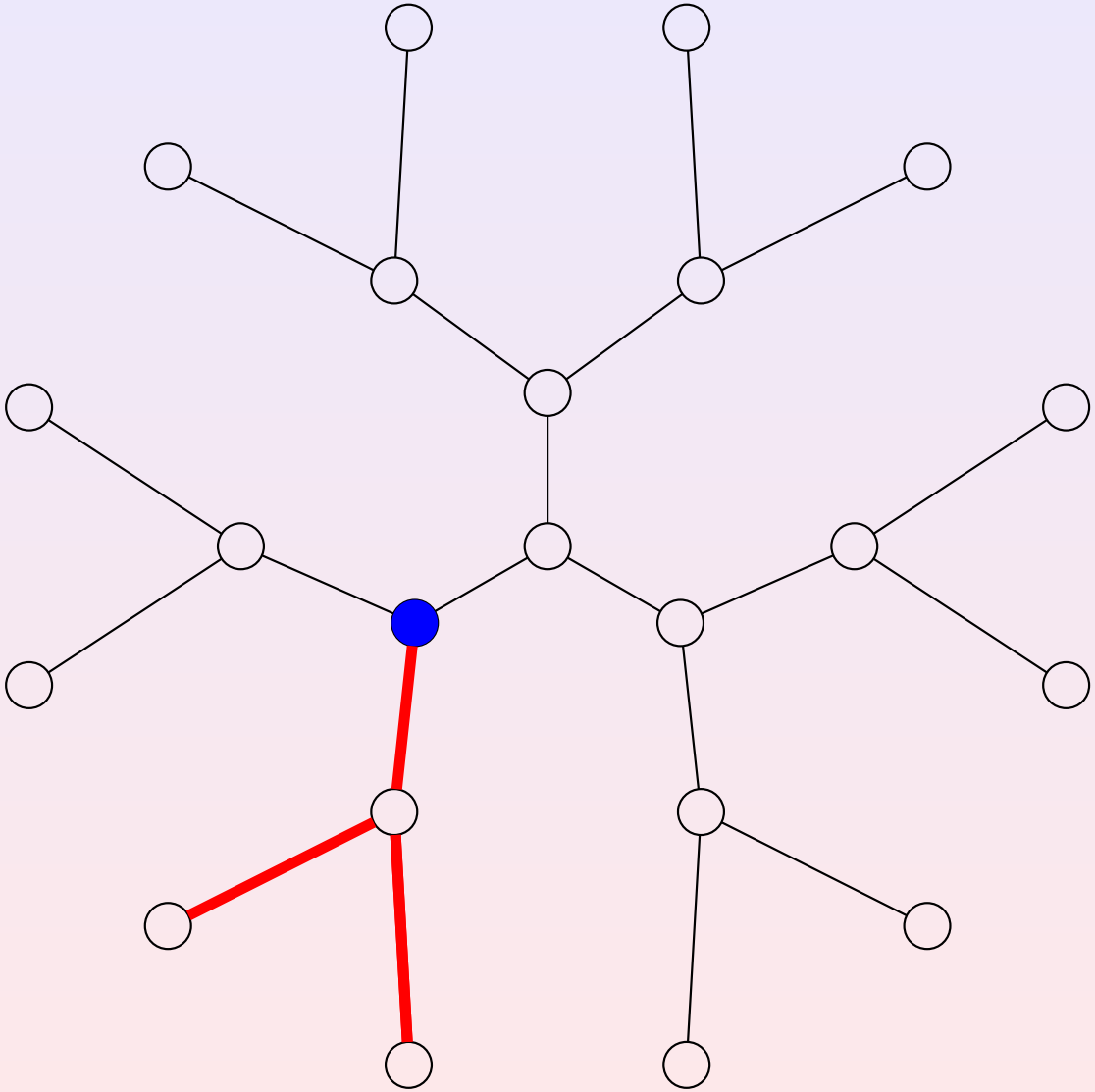
A simple example



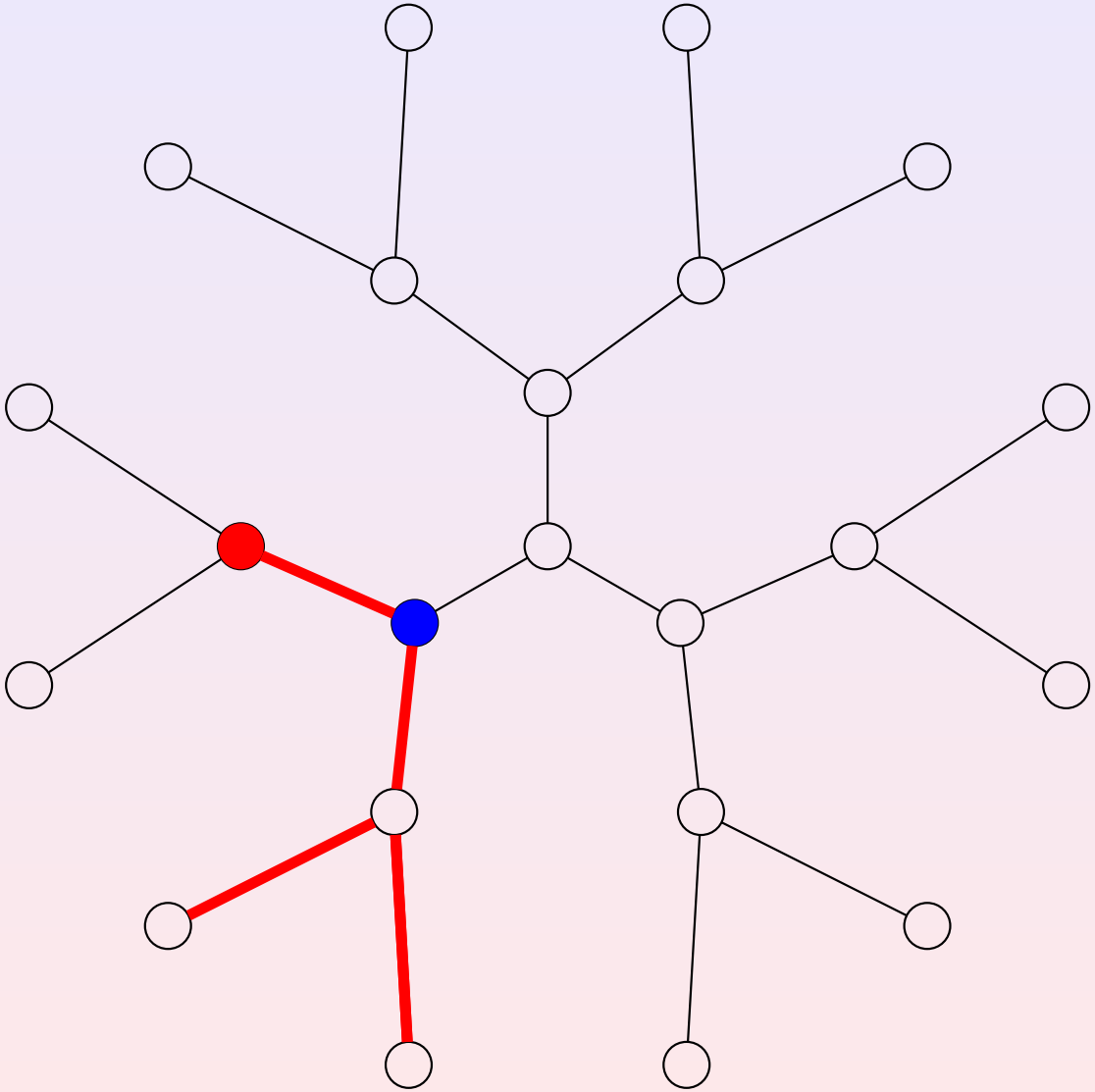
A simple example



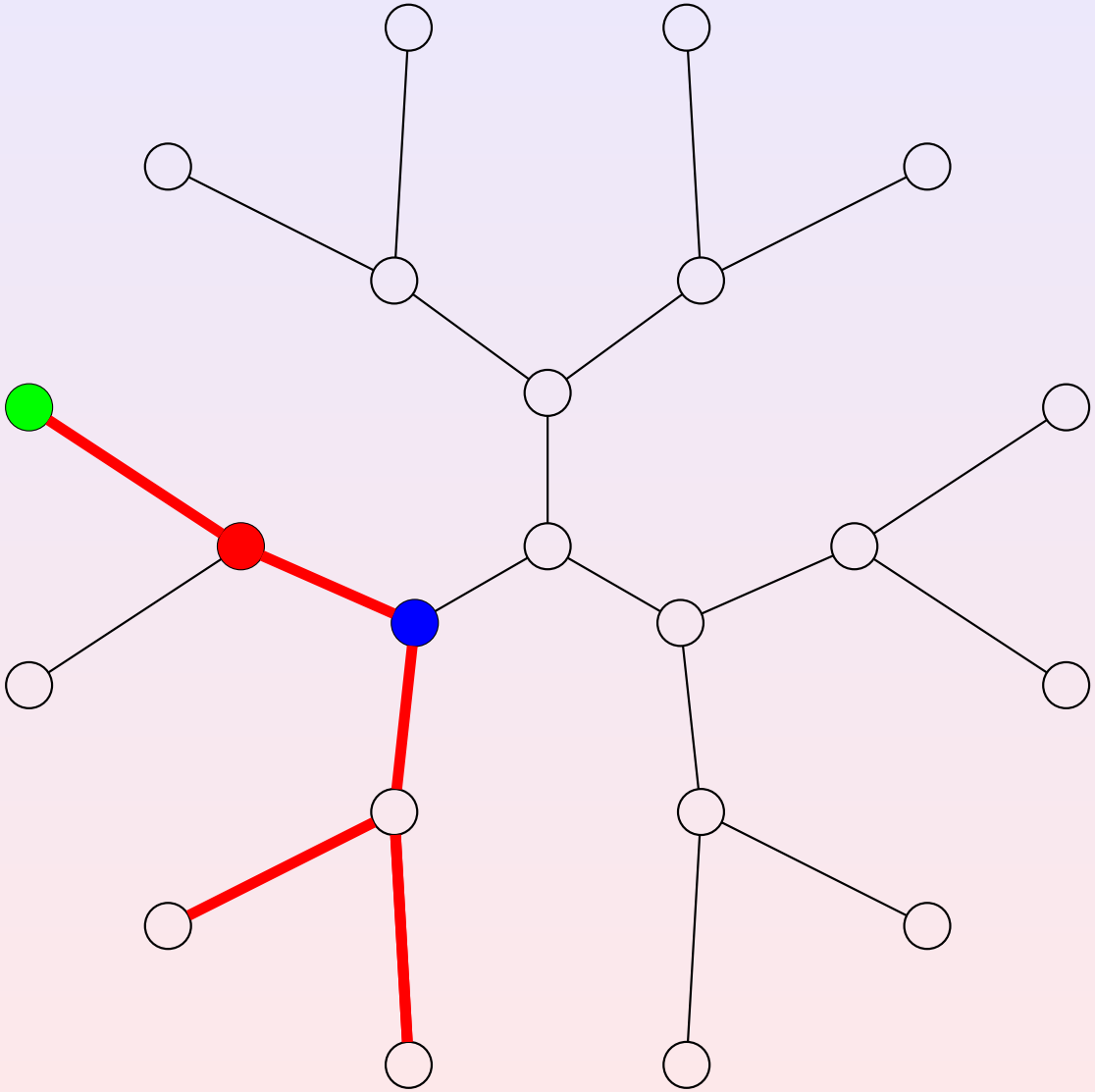
A simple example



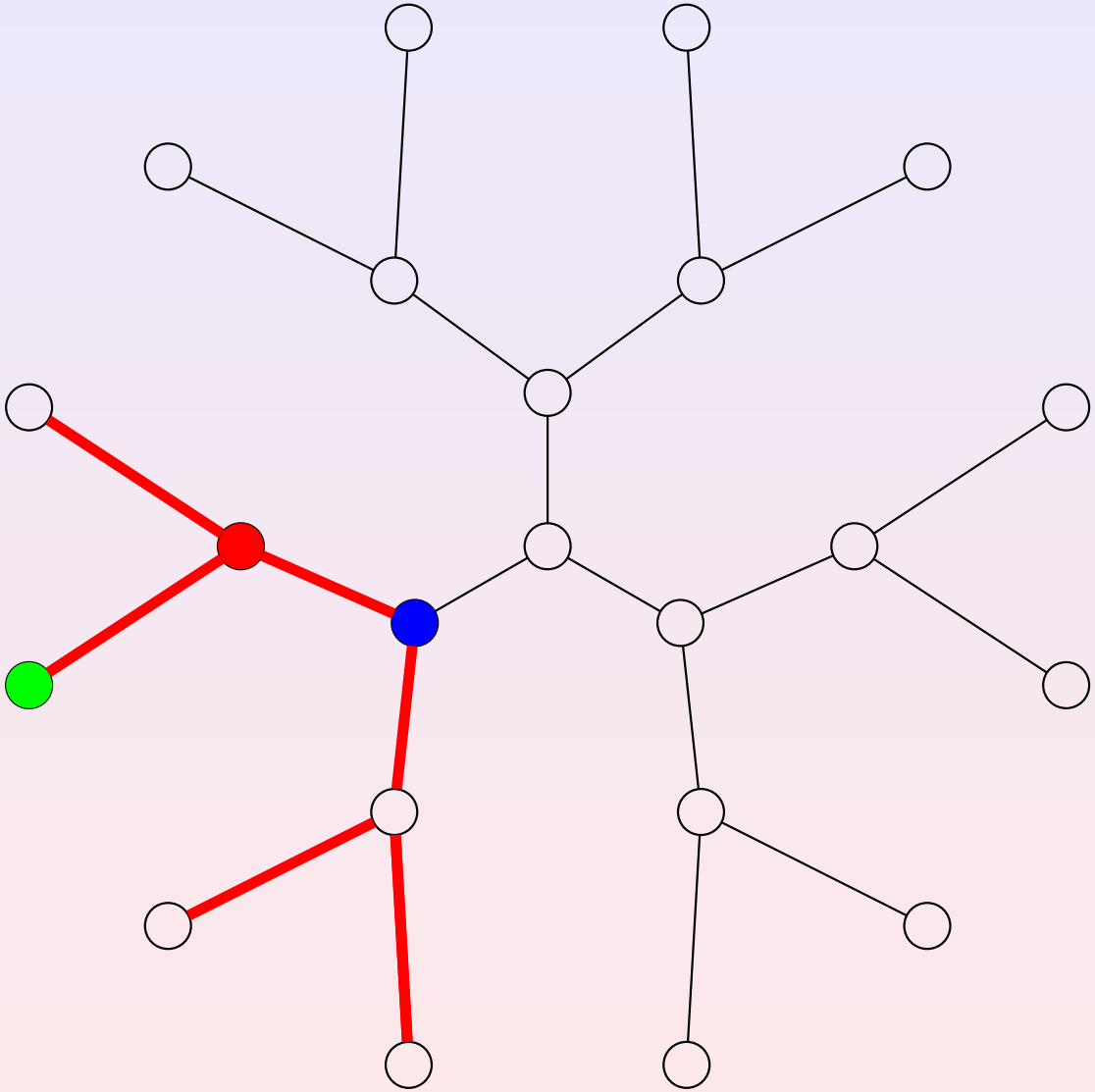
A simple example



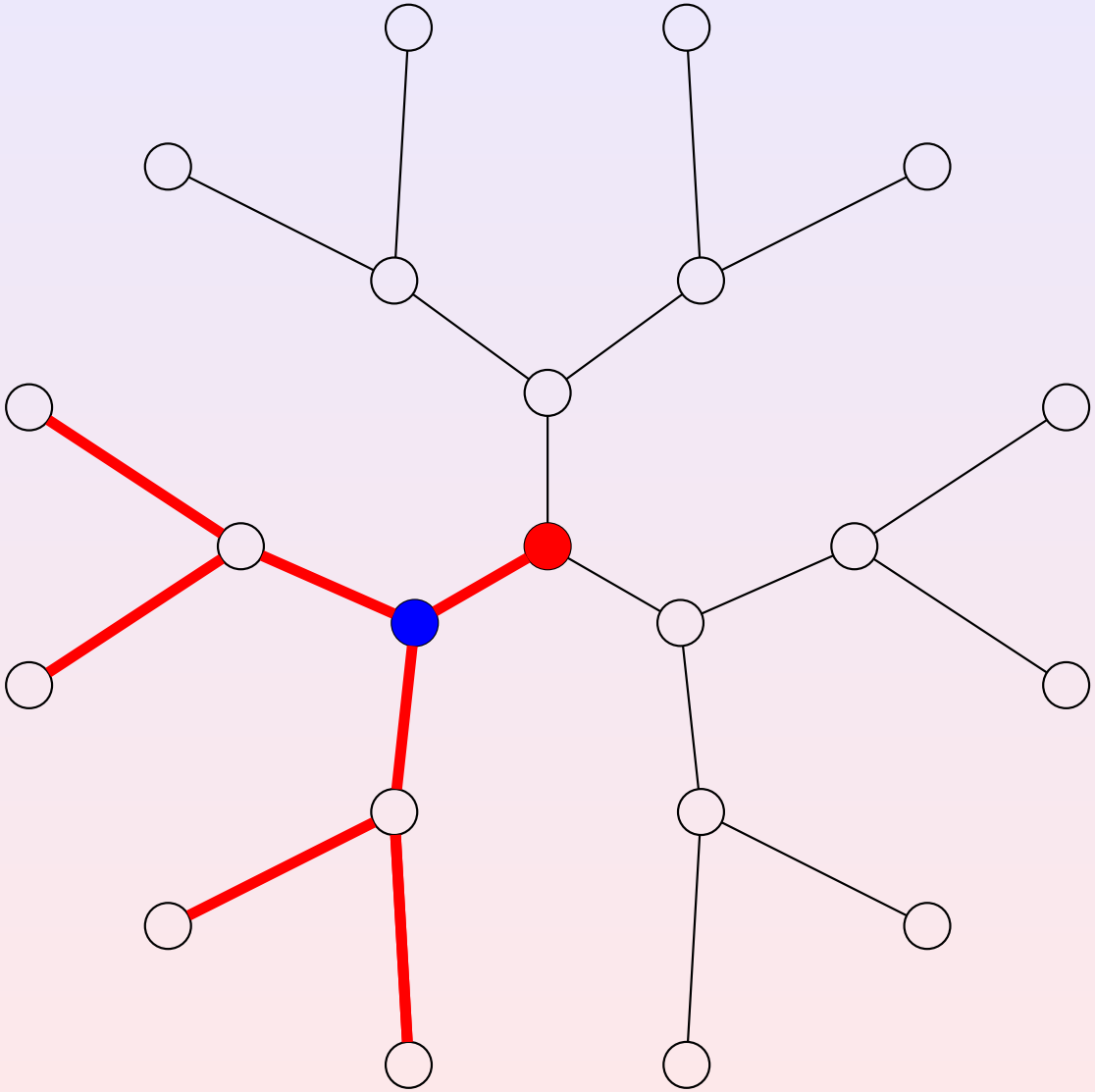
A simple example



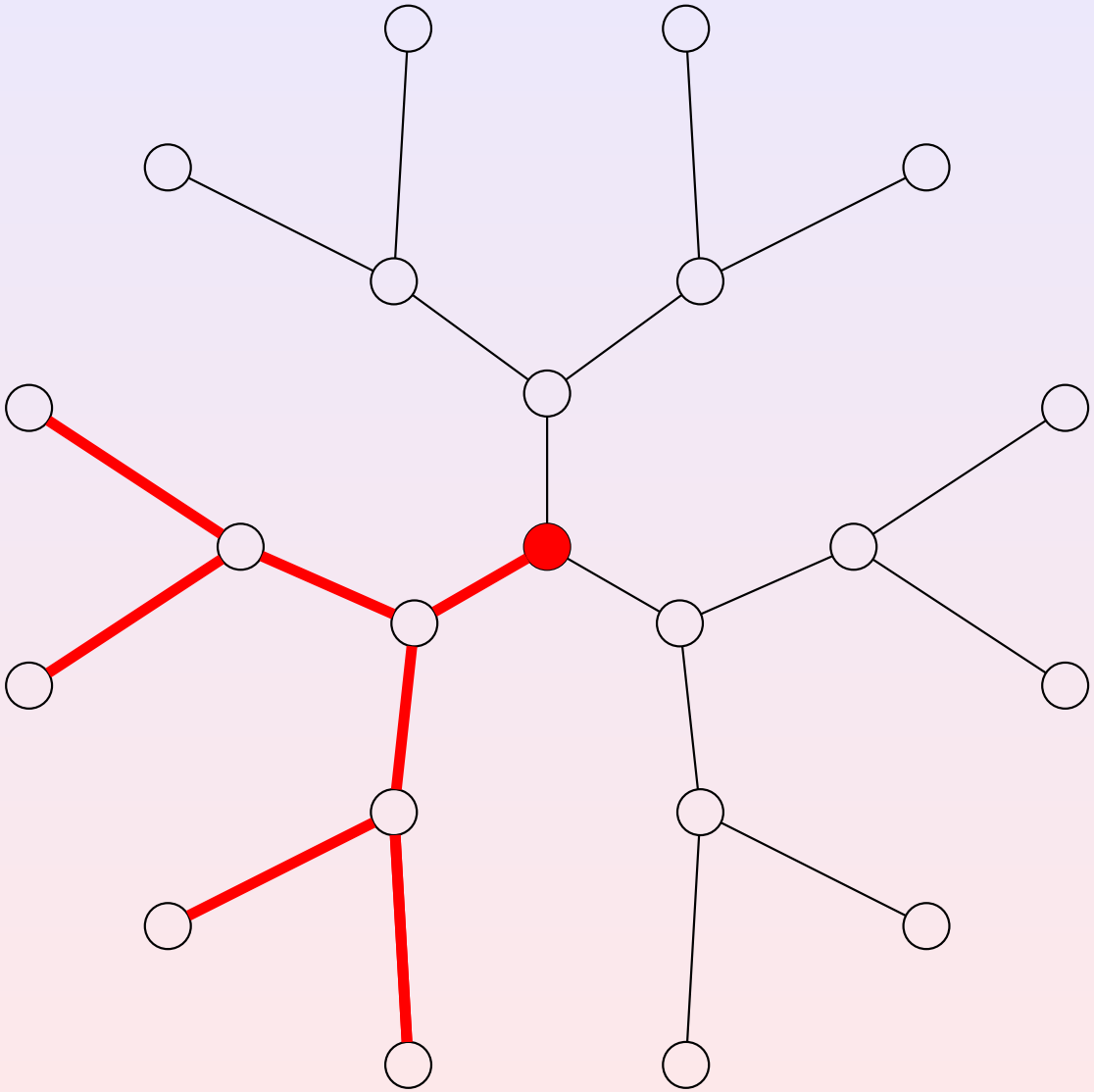
A simple example



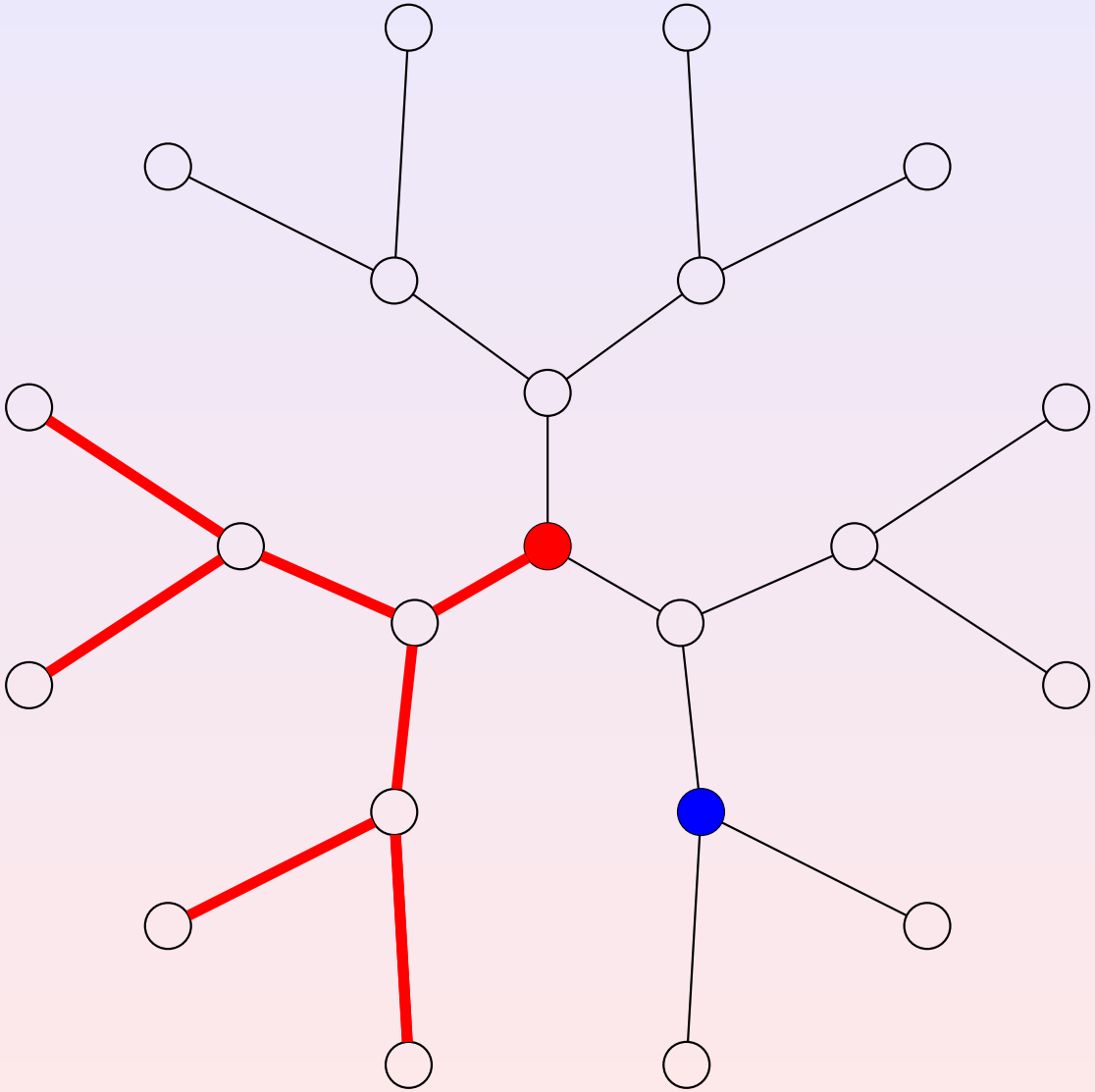
A simple example



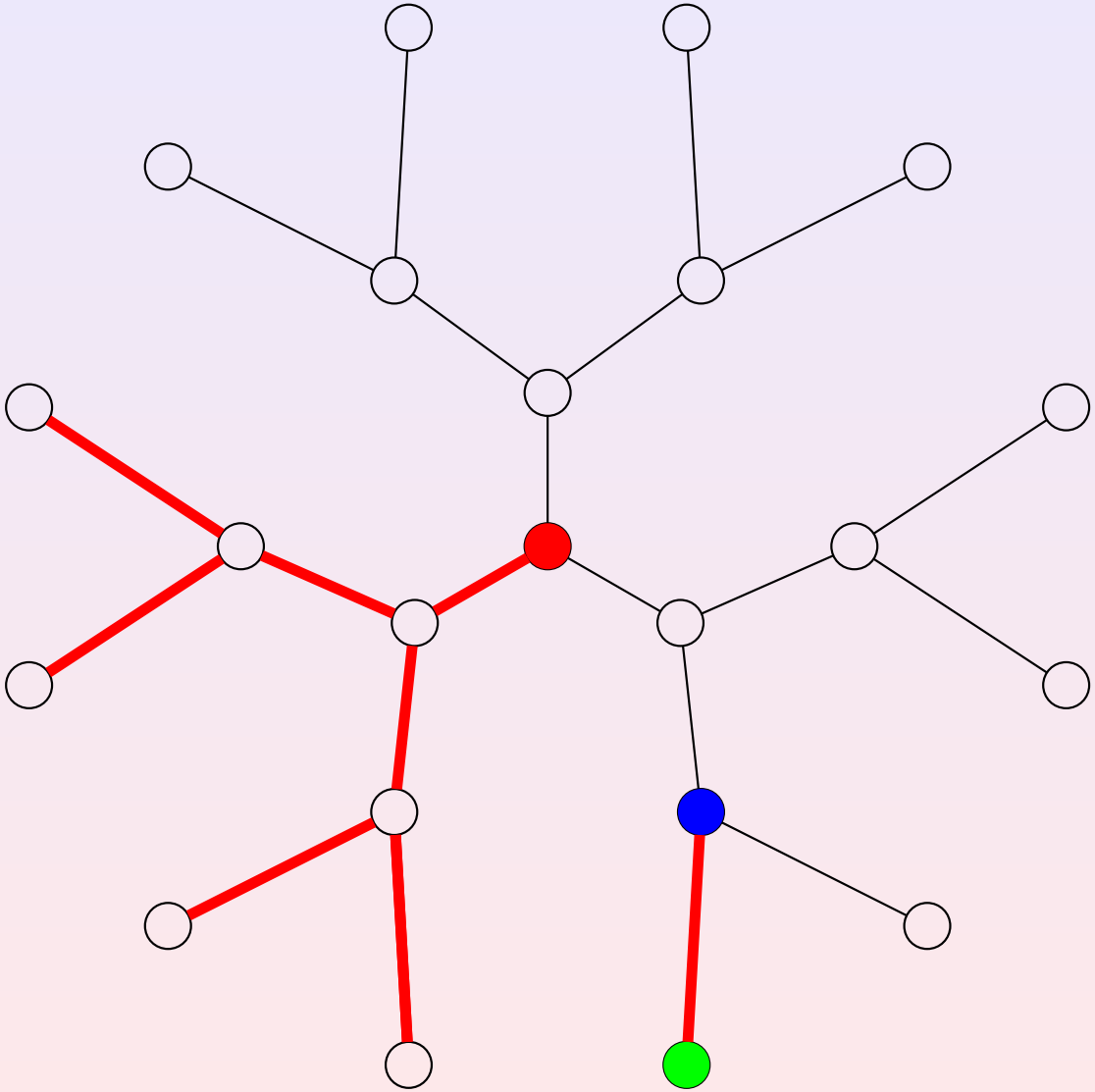
A simple example



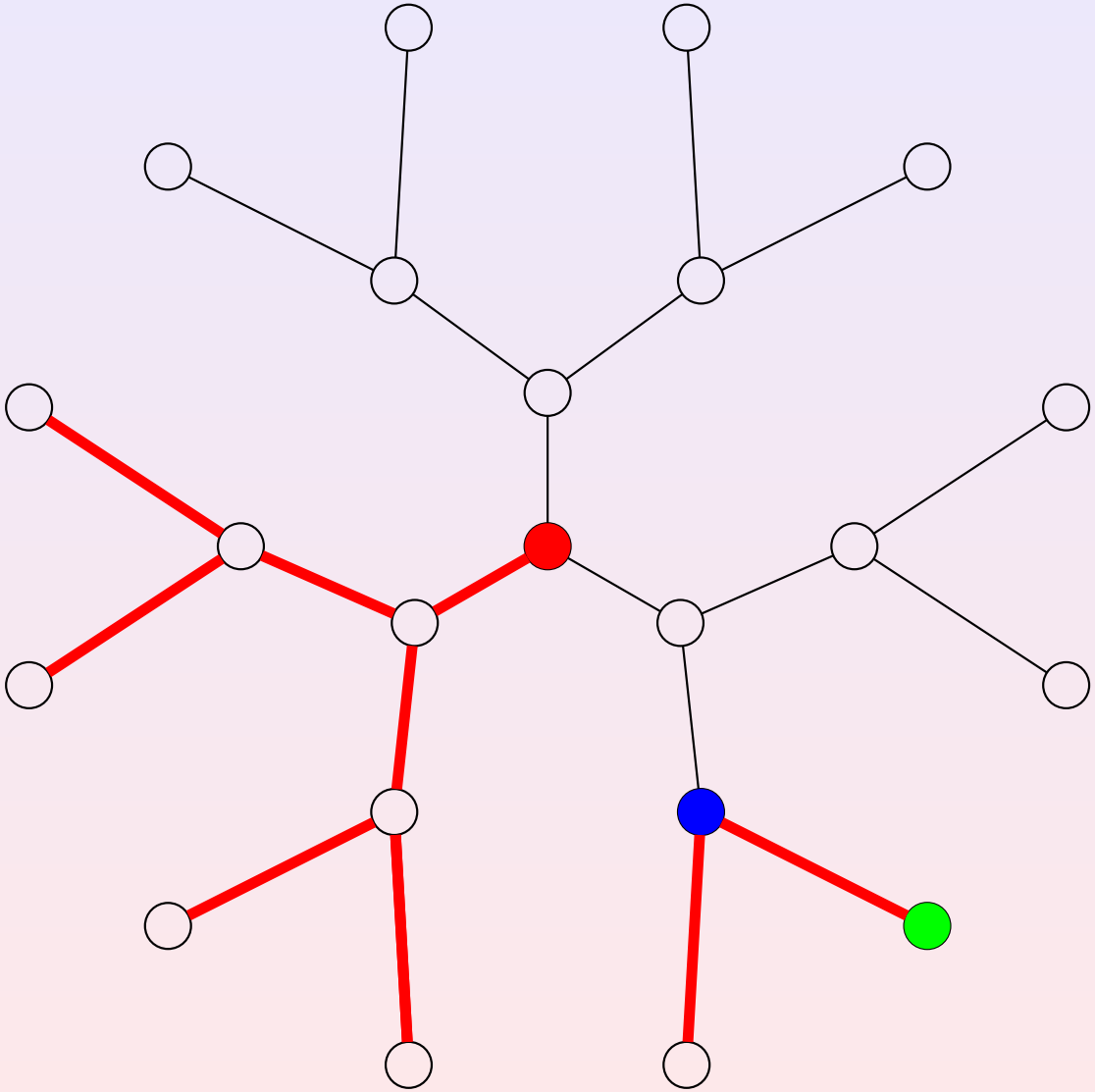
A simple example



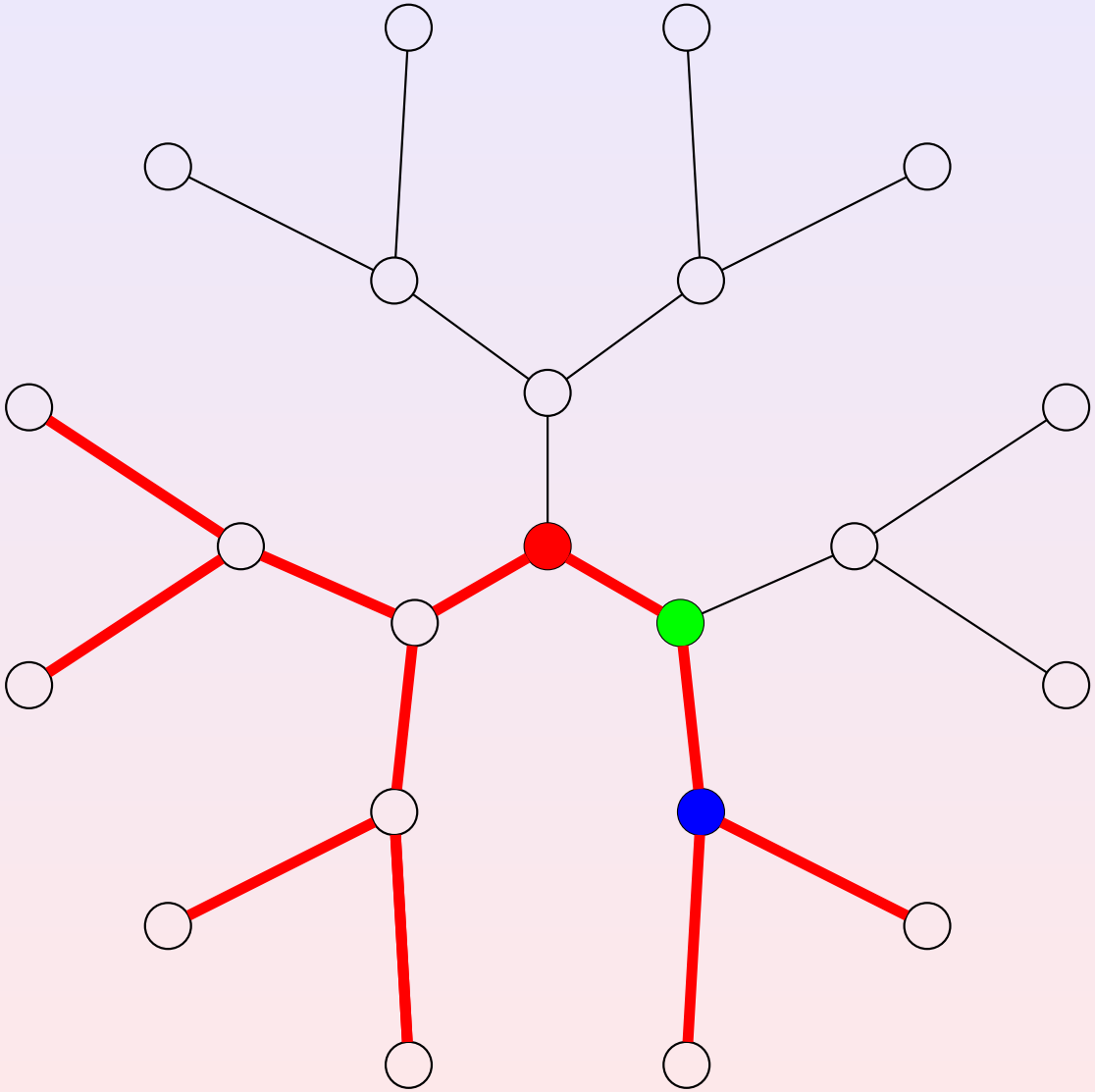
A simple example



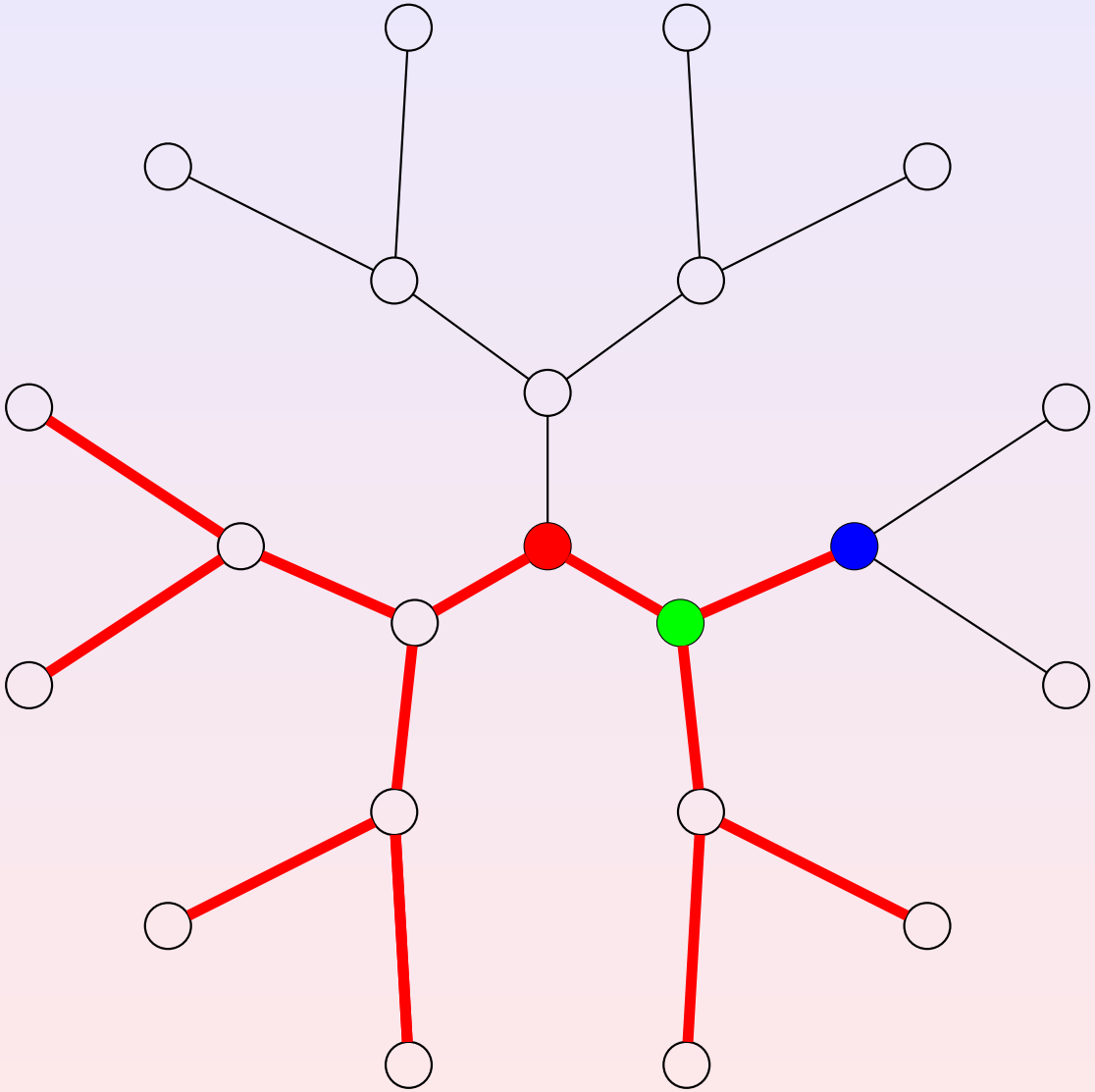
A simple example



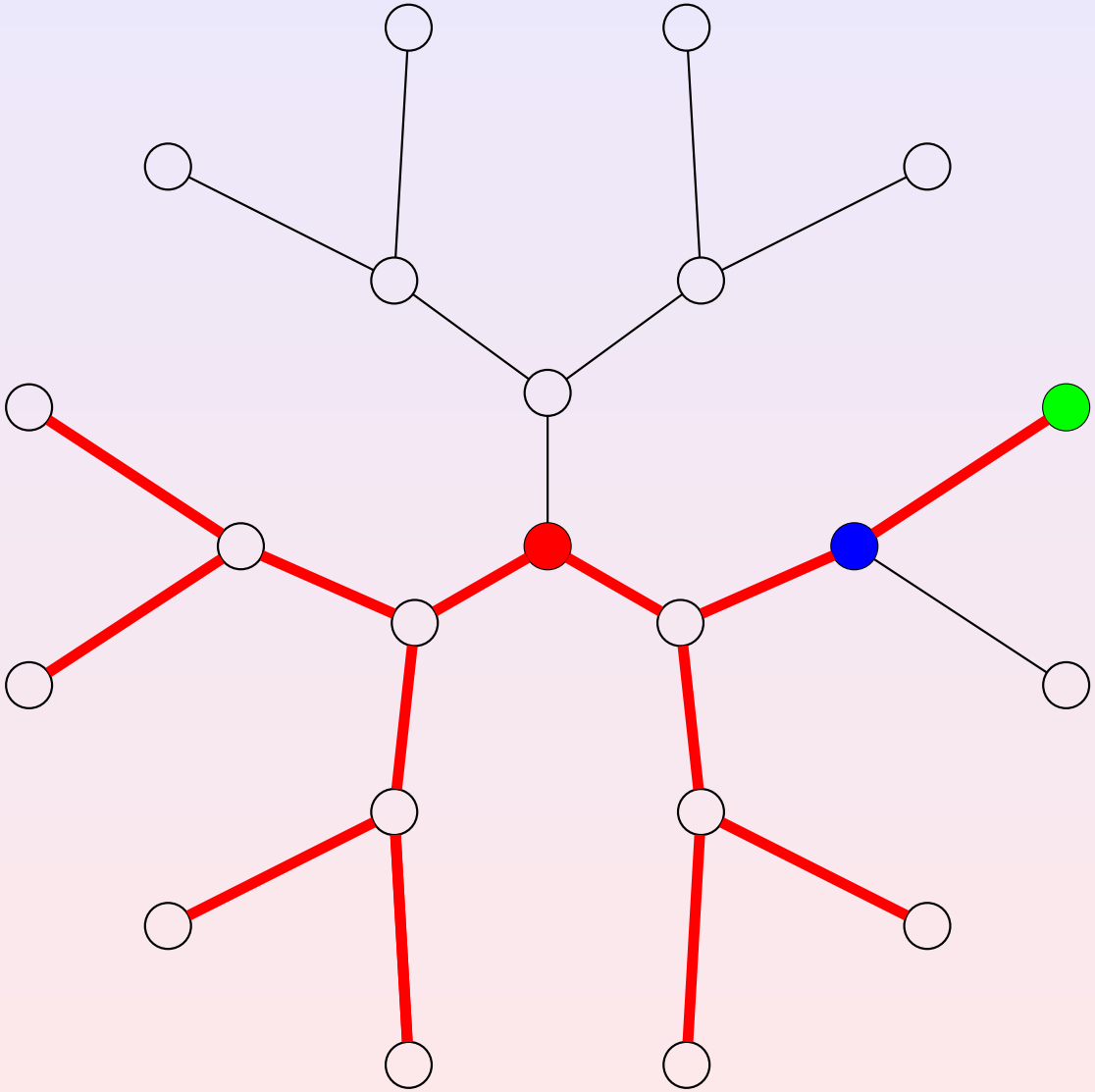
A simple example



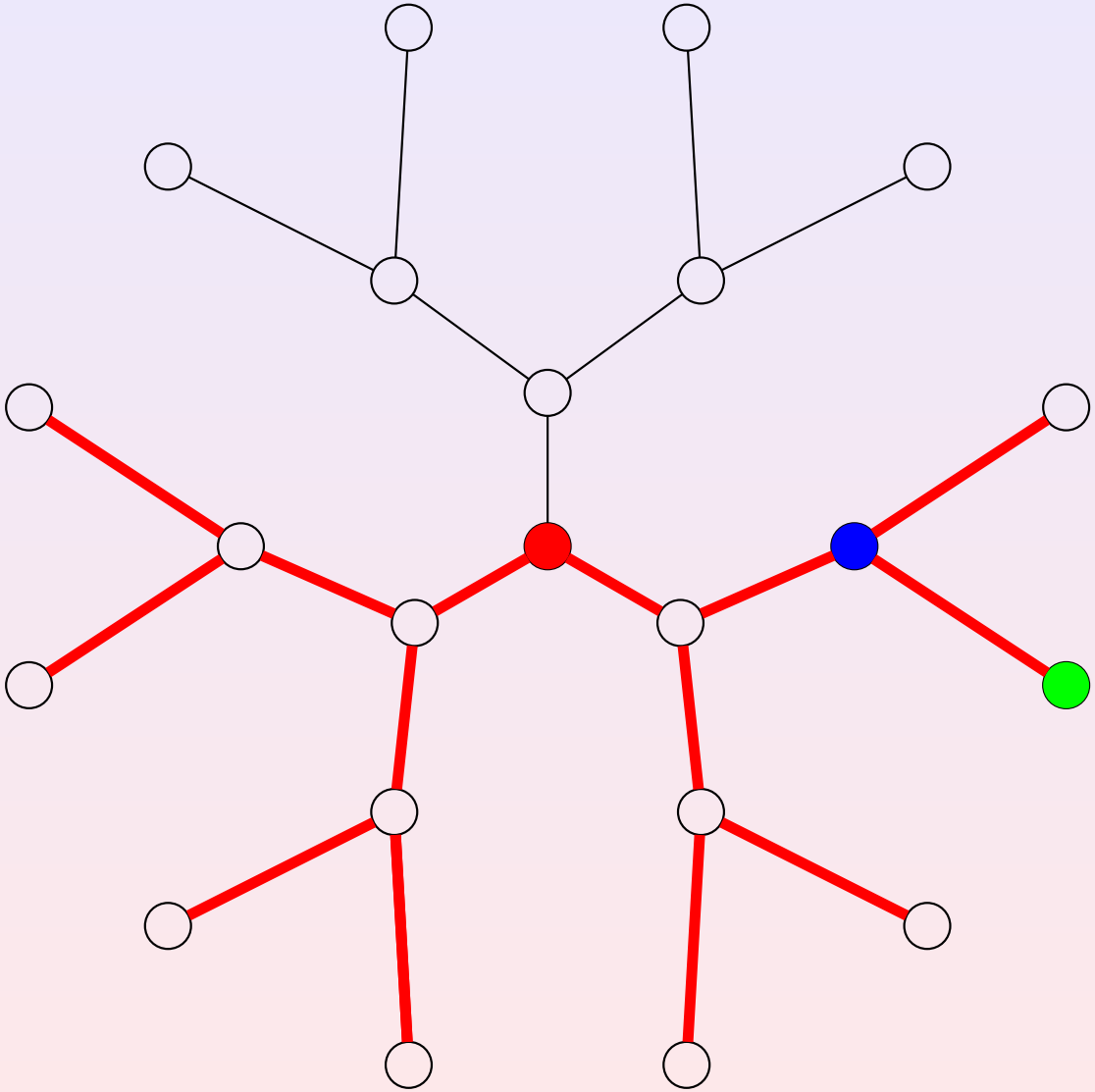
A simple example



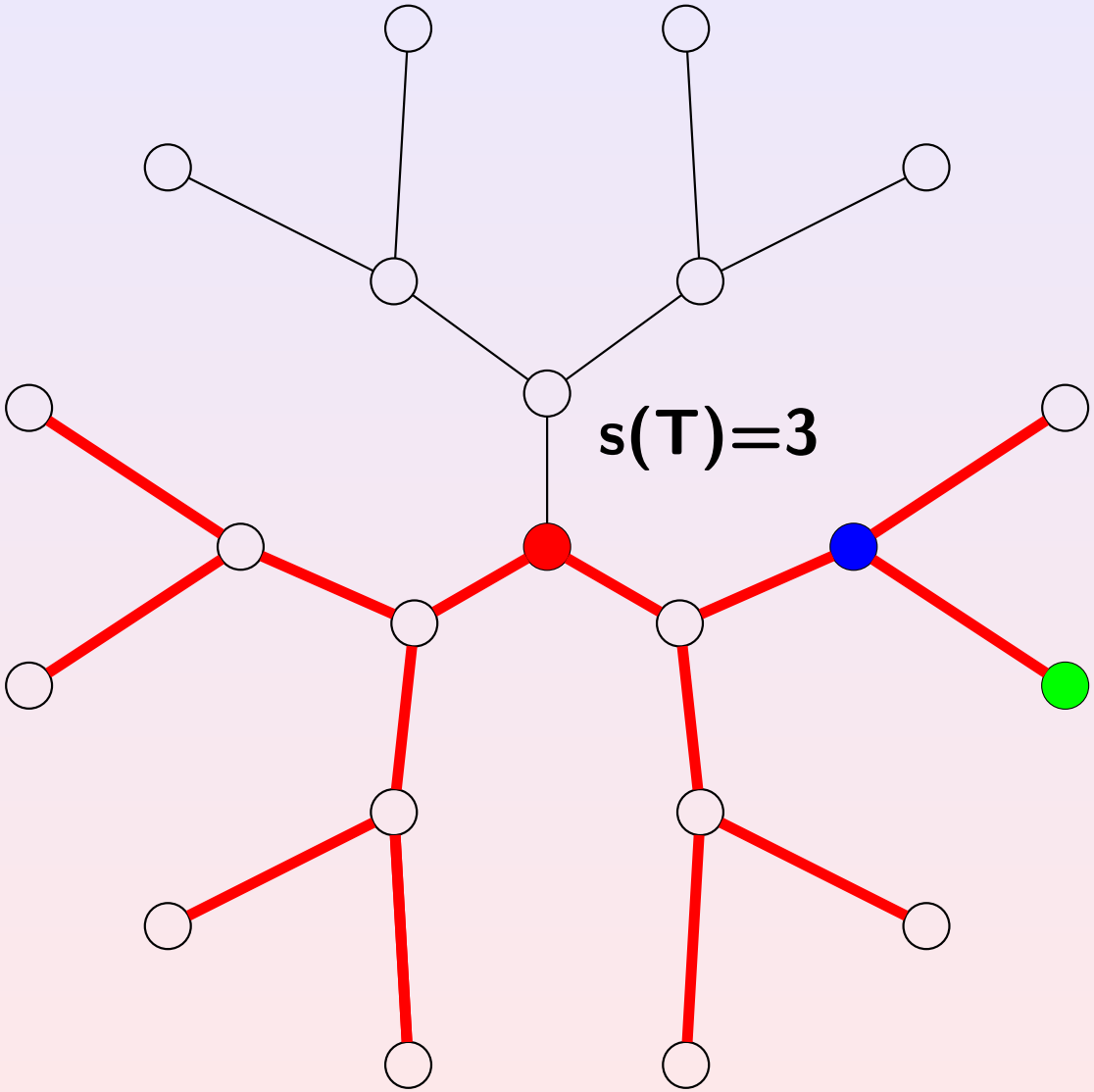
A simple example



A simple example



A simple example

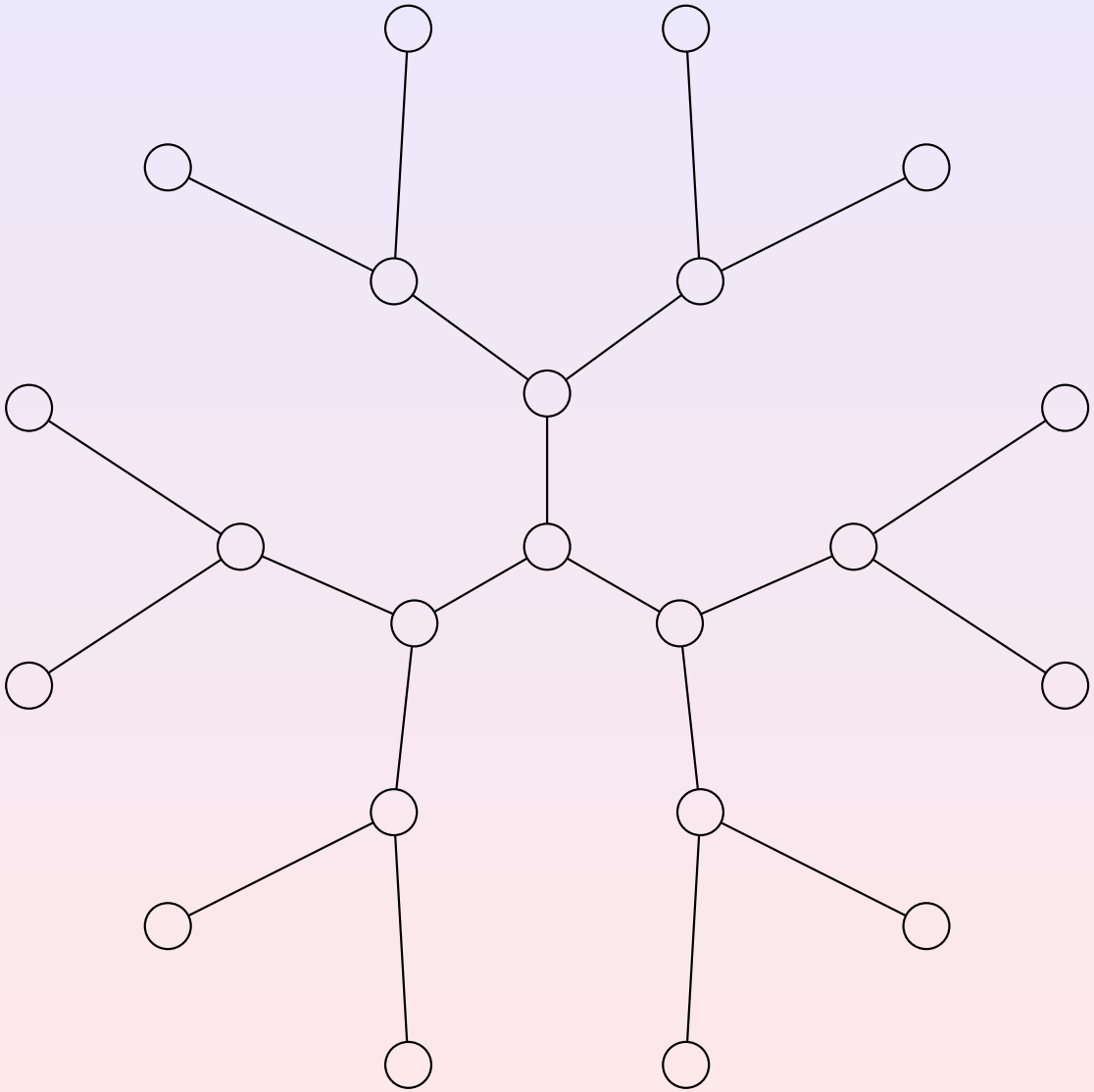


Visible Graph Searching

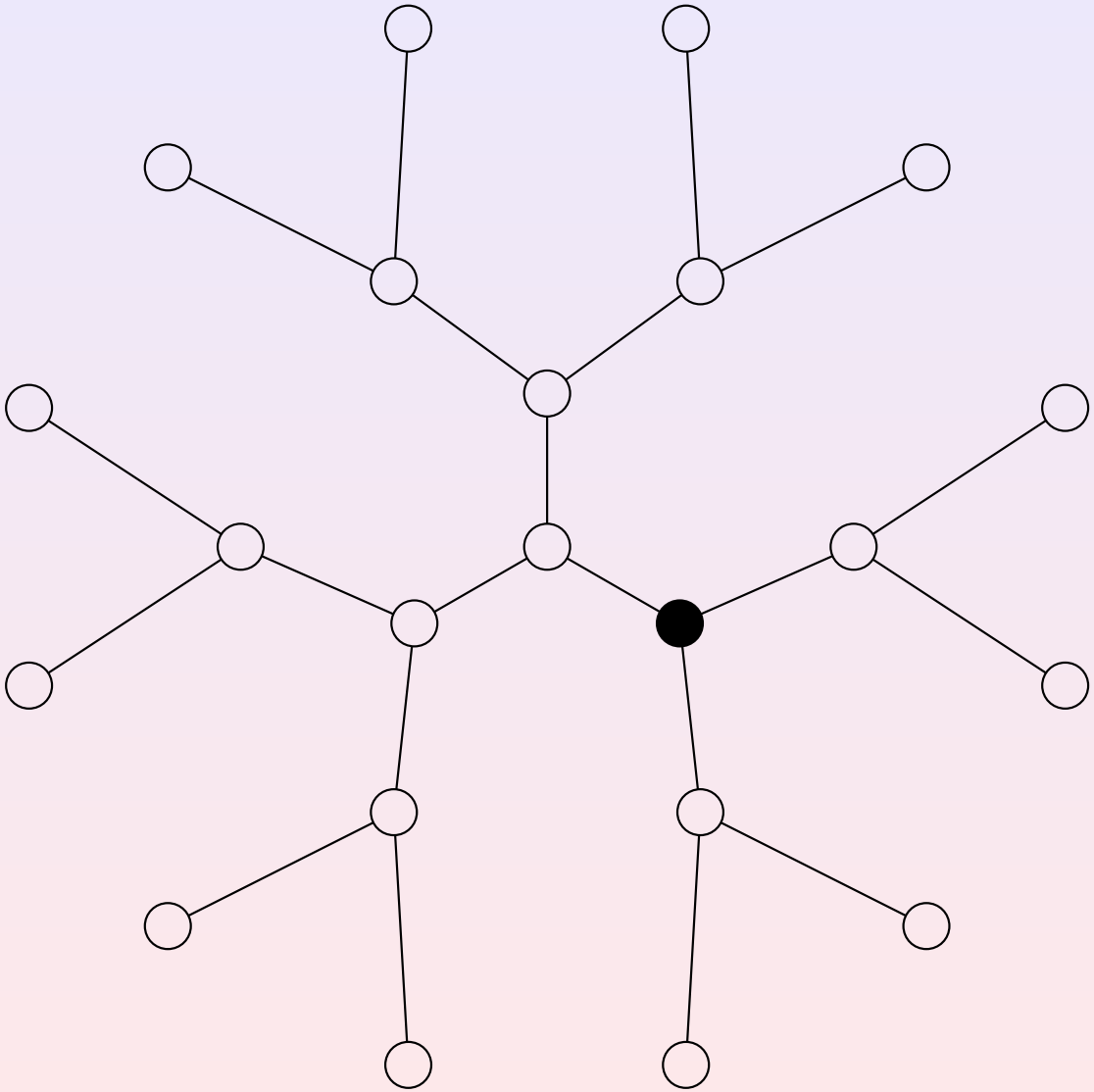
Visible fugitive

The fugitive is visible if, at every step, searchers know its position, i.e. the connected component of the contaminated part where the fugitive stands.
Let $vs(G)$ be the visible search number of the graph G.

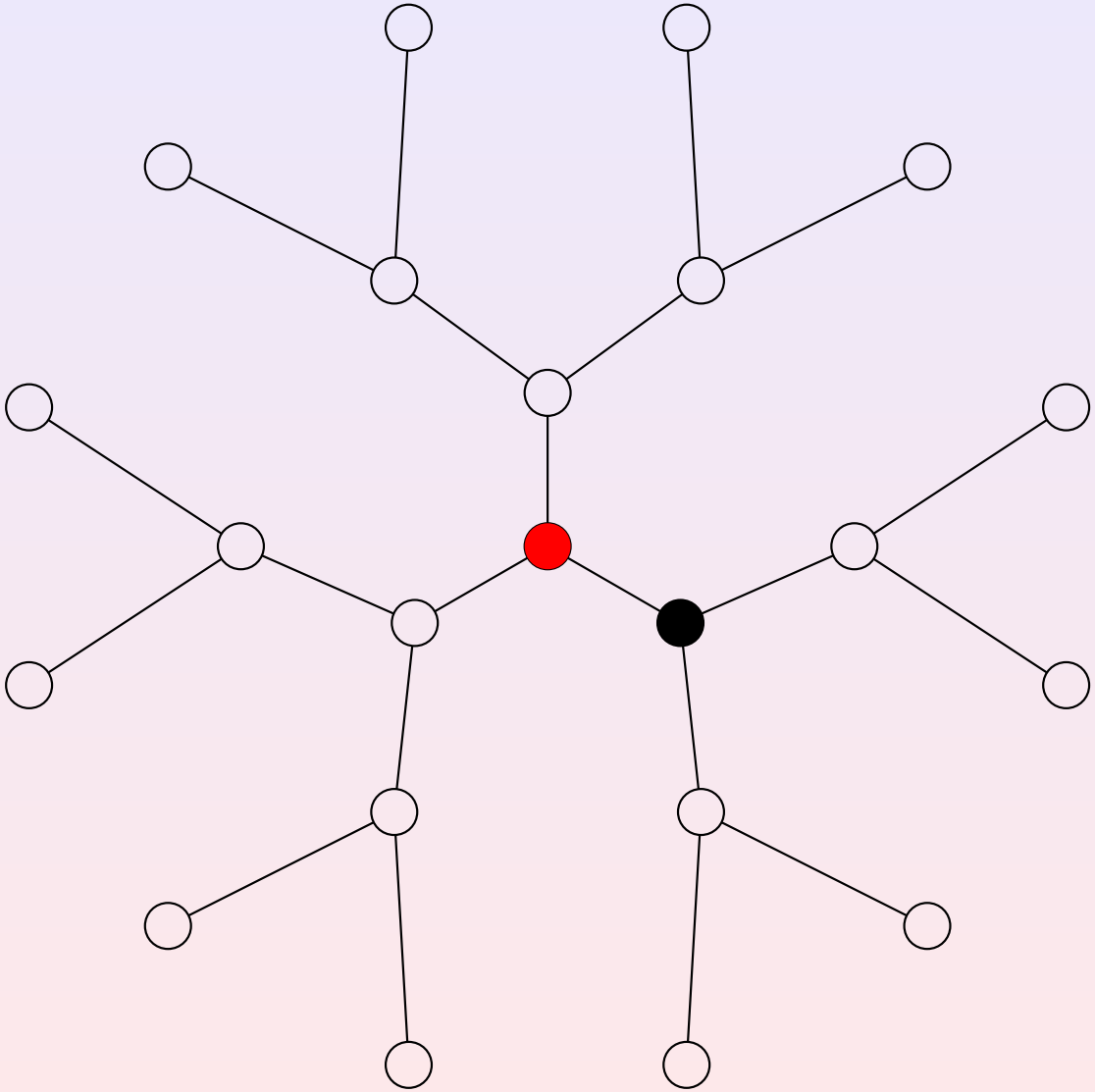
Visible graph searching in a tree



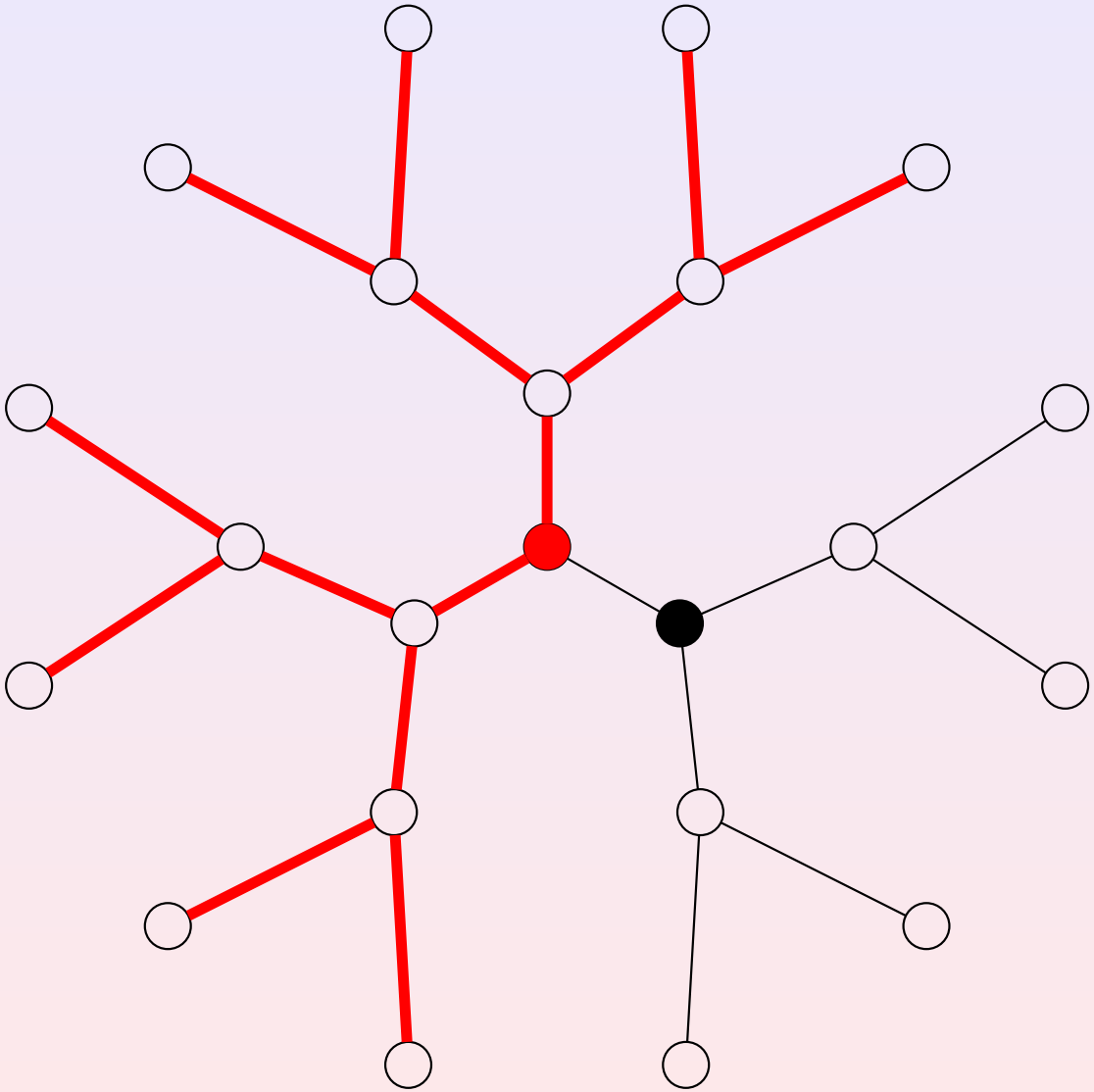
Visible graph searching in a tree



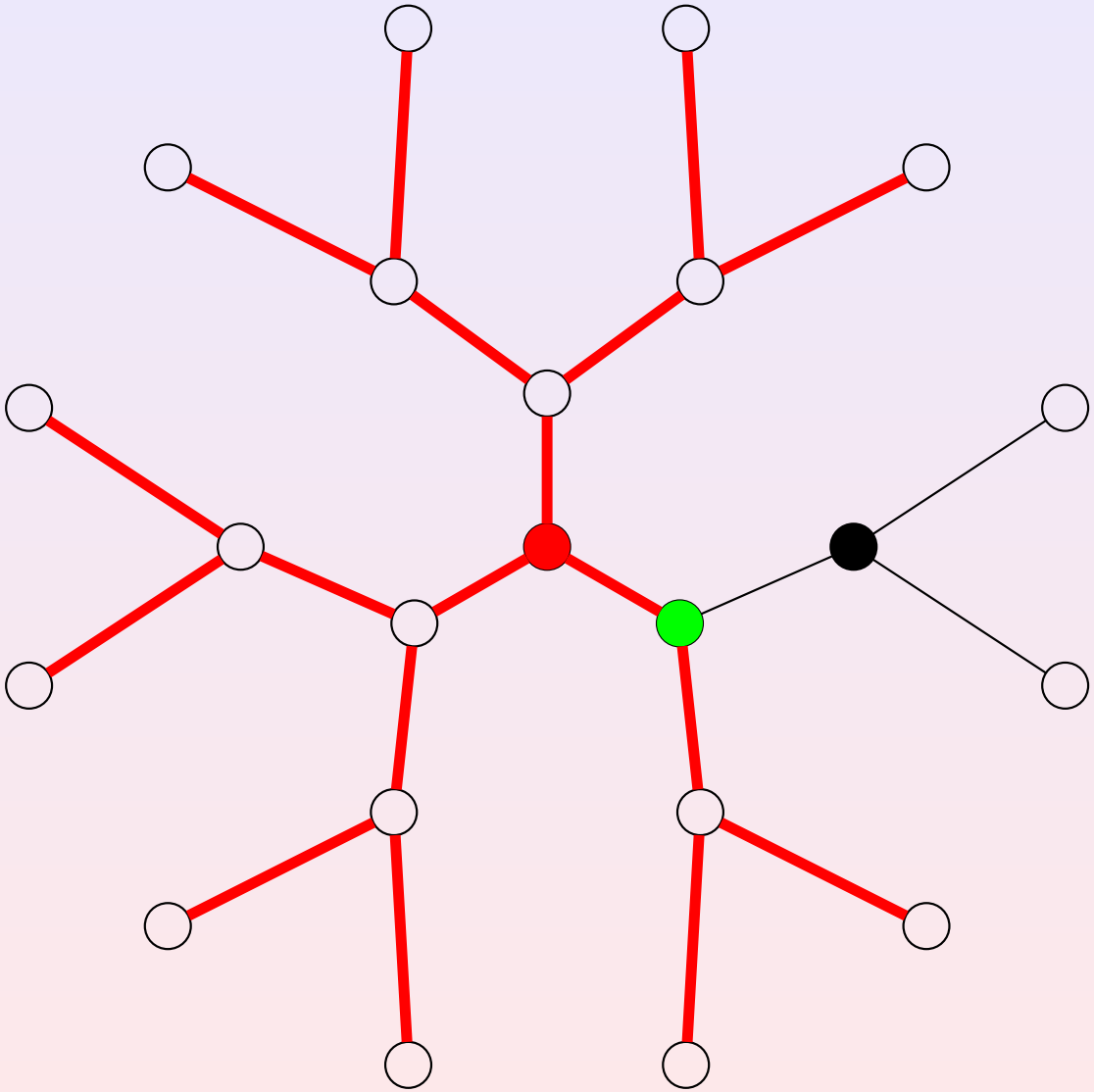
Visible graph searching in a tree



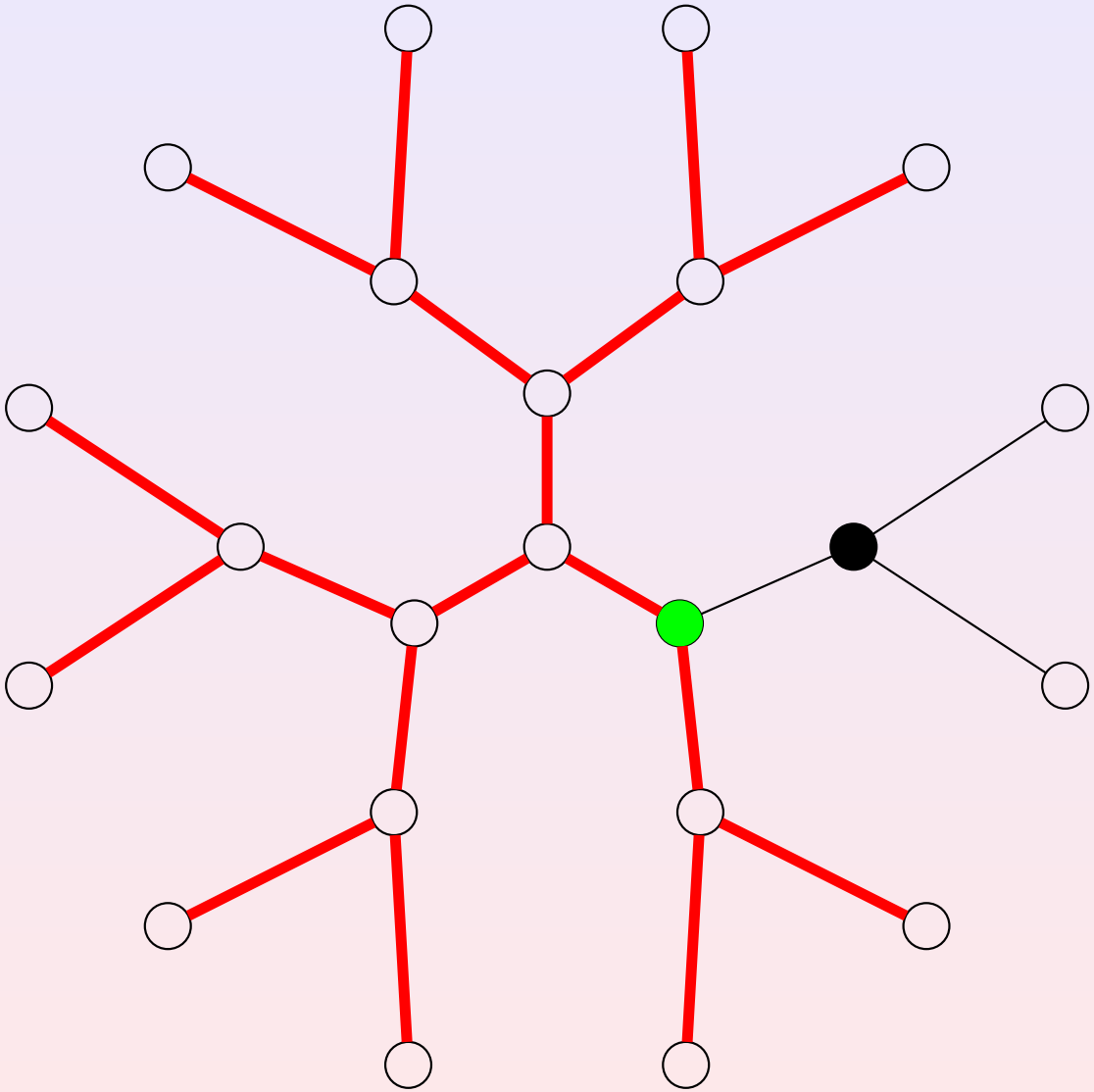
Visible graph searching in a tree



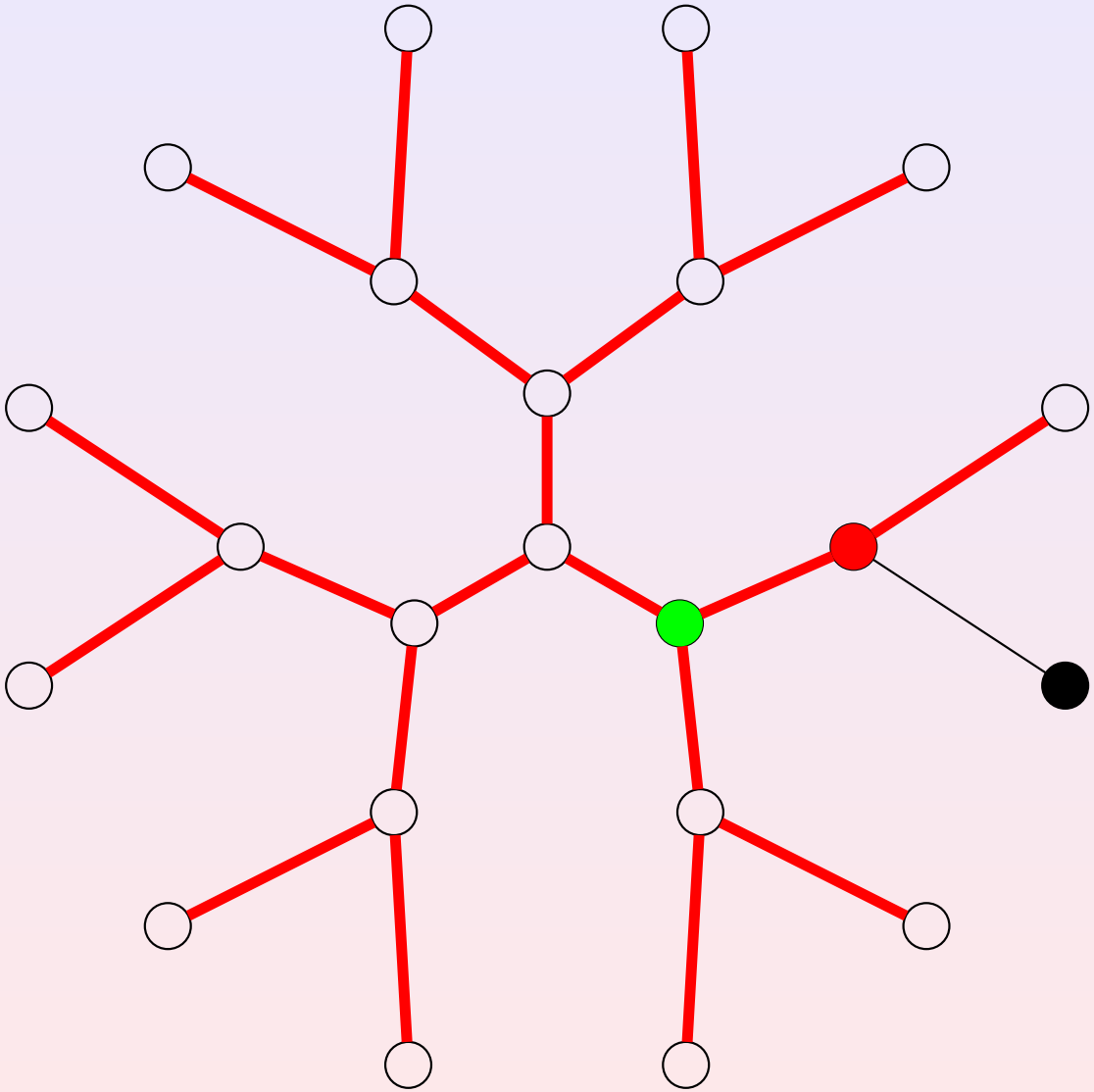
Visible graph searching in a tree



Visible graph searching in a tree

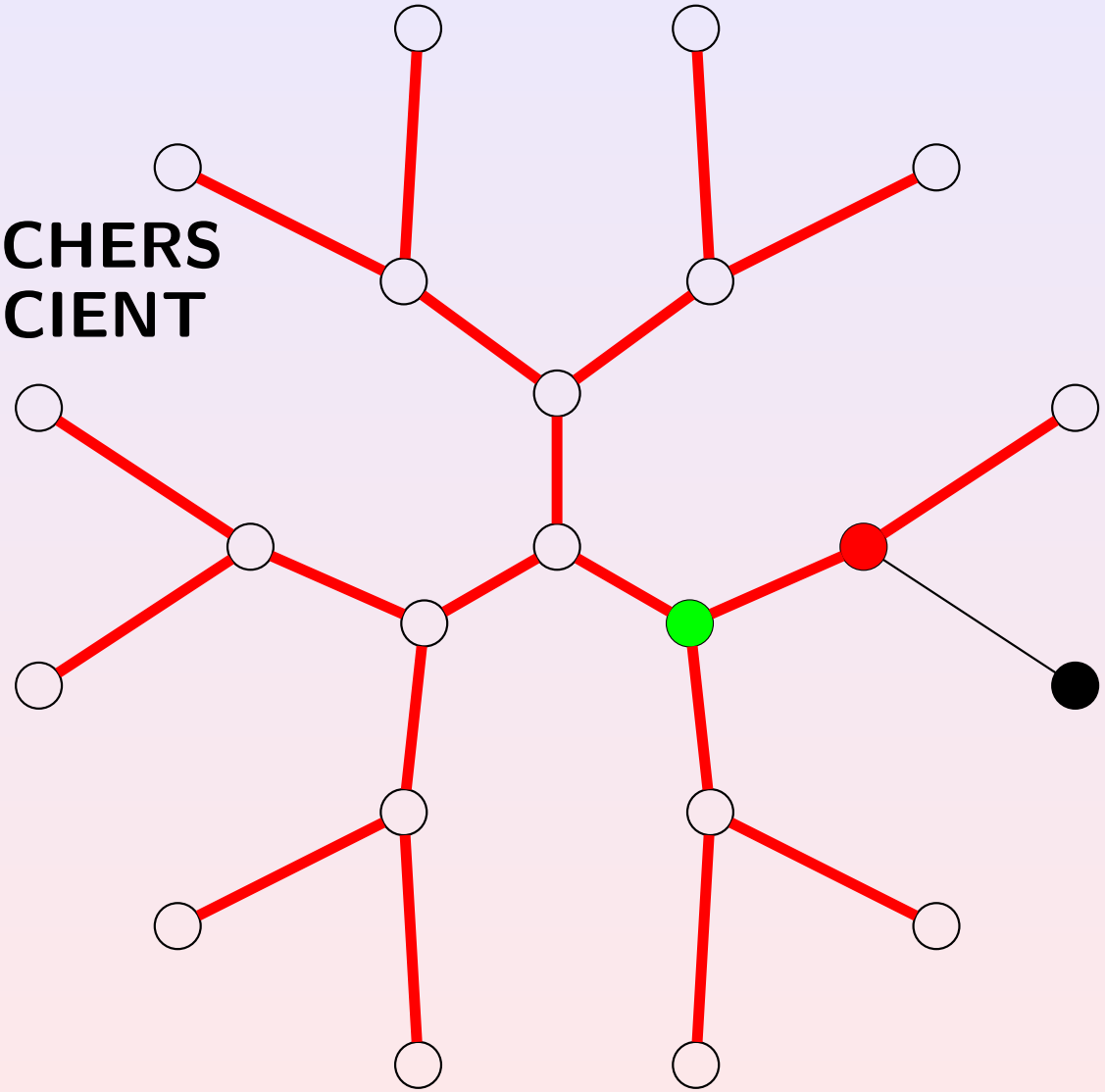


Visible graph searching in a tree



Visible graph searching in a tree

TWO SEARCHERS ARE SUFFICIENT



Monotony and Recontamination

Recontamination

A vertex v is recontaminated during a search strategy S if it has been cleared at a step s of S and the fugitive can access v at step $s' > s$.

Monotony

A search strategy is monotone if no recontamination ever occurs. That is, a cleared vertex remains clear until the end of the strategy.
Let ms(G) be the monotone search number of the graph G.

Monotony and Recontamination

Recontamination

A vertex v is recontaminated during a search strategy S if it has been cleared at a step s of S and the fugitive can access v at step $s' > s$.

Monotony

A search strategy is monotone if no recontamination ever occurs. That is, a cleared vertex remains clear until the end of the strategy.
Let mvs(G) be the monotone visible search number of G.

Monotony and Recontamination

Does Recontamination help?

Is it more difficult to catch a fugitive in a monotone way? In other words, do we have $\mathbf{ms}(G) > \mathbf{s}(G)$?

Recontamination does not help: $s(G) = ms(G)$

- **Bienstock and Seymour**, J.of Alg., 1991
  Monotonicity in graph searching.
- **LaPaugh**, J.of ACM, 1993
  Recontamination does not help to search a graph.

Case of a visible fugitive: $vs(G) = mvs(G)$

- **Seymour and Thomas**, J. of Comb. Th., 1993.
  Graph searching and a min-max theorem for tree-width

Monotony and Recontamination

Does Recontamination help?

Is it more difficult to catch a fugitive in a monotone way? In other words, do we have $\mathbf{ms}(G) > \mathbf{s}(G)$?

Recontamination does not help: $s(G) = ms(G)$

- **Bienstock and Seymour**, J.of Alg., 1991
  Monotonicity in graph searching.
- **LaPaugh**, J.of ACM, 1993
  Recontamination does not help to search a graph.

Case of a visible fugitive: $vs(G) = mvs(G)$

- **Seymour and Thomas**, J. of Comb. Th., 1993.
  Graph searching and a min-max theorem for tree-width

Recontamination does not help

In other words...

For any graph G, there exists a monotone search strategy using at most $\mathbf{s}(G)$ searchers (resp. $\mathbf{vs}(G)$ searchers in the visible case).

The corresponding decision problems are in NP

A monotone search strategy terminates after a linear number of steps. It gives a certificate checkable in polynomial time.

Another important consequence

We can restrict the analysis to monotone search strategies.

Recontamination does not help

In other words...

For any graph G, there exists a monotone search strategy using at most $\mathbf{s}(G)$ searchers (resp. $\mathbf{vs}(G)$ searchers in the visible case).

The corresponding decision problems are in NP

A monotone search strategy terminates after a linear number of steps. It gives a certificate checkable in polynomial time.

Another important consequence

We can restrict the analysis to monotone search strategies.

Recontamination does not help

In other words...

For any graph G, there exists a monotone search strategy using at most $\mathbf{s}(G)$ searchers (resp. $\mathbf{vs}(G)$ searchers in the visible case).

The corresponding decision problems are in NP

A monotone search strategy terminates after a linear number of steps. It gives a certificate checkable in polynomial time.

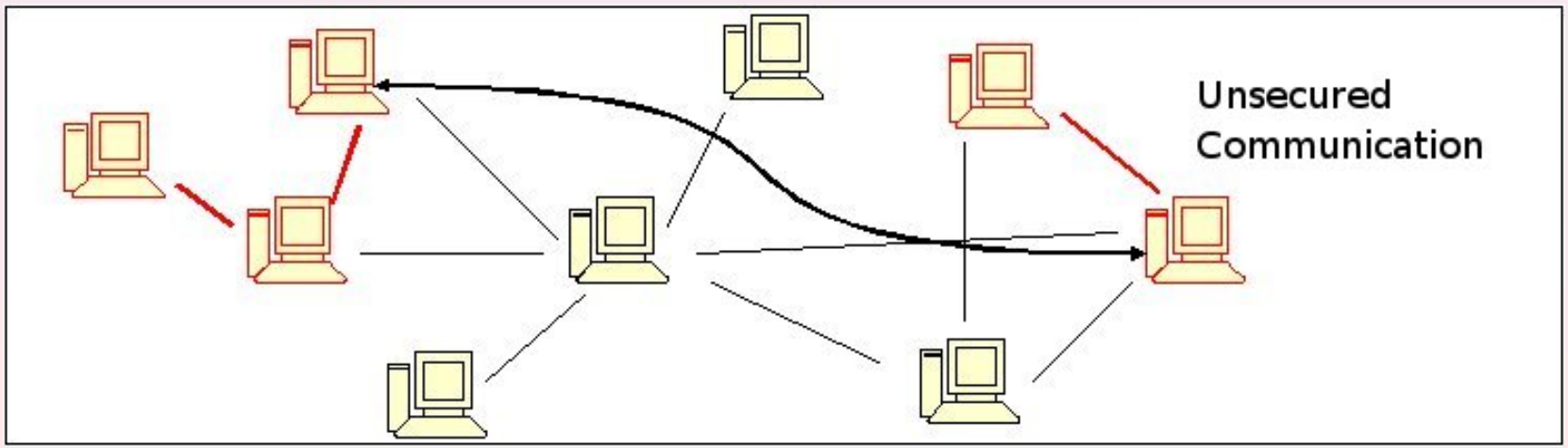
Another important consequence

We can restrict the analysis to monotone search strategies.

Connected Graph Searching

Limits of Parson's model

- Searchers cannot move at will in a real network;
- It would be better to let the searchers grouped.



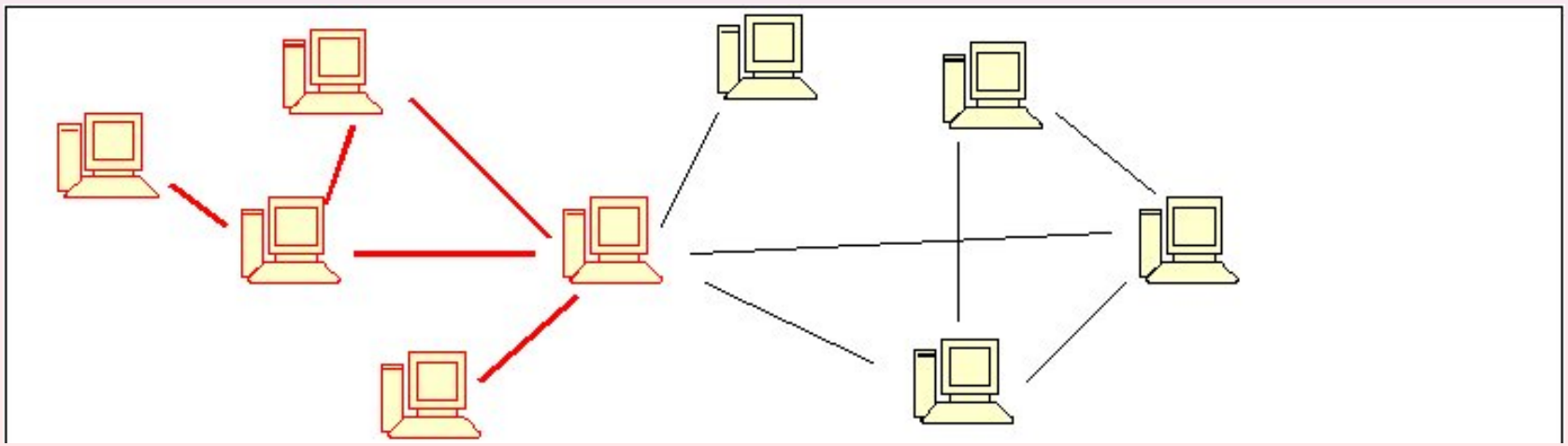
Connected Graph Searching

Limits of Parson's model

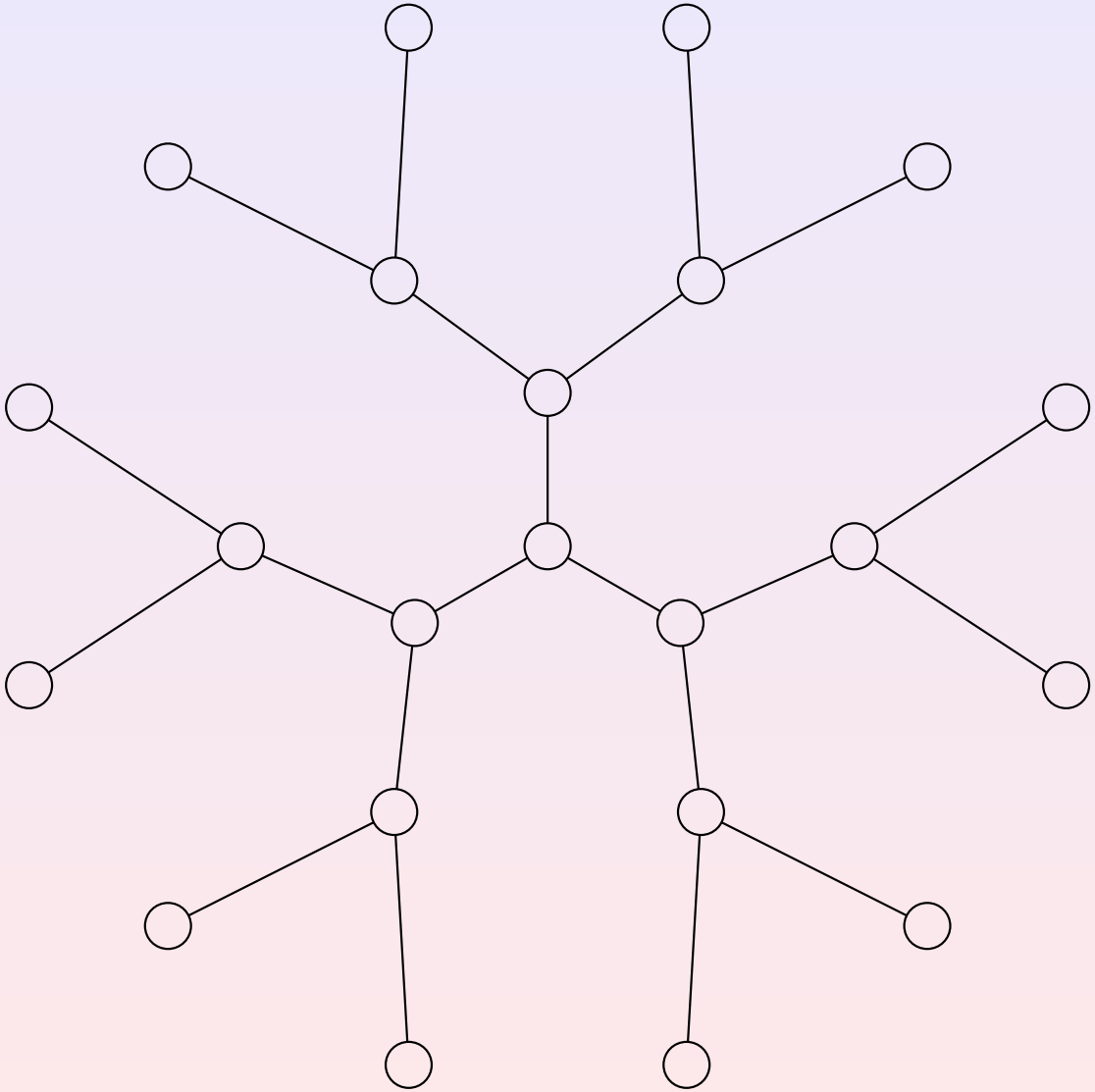
- Searchers cannot move at will in a real network;
- It would be better to let the searchers grouped.

Connected Search Strategy

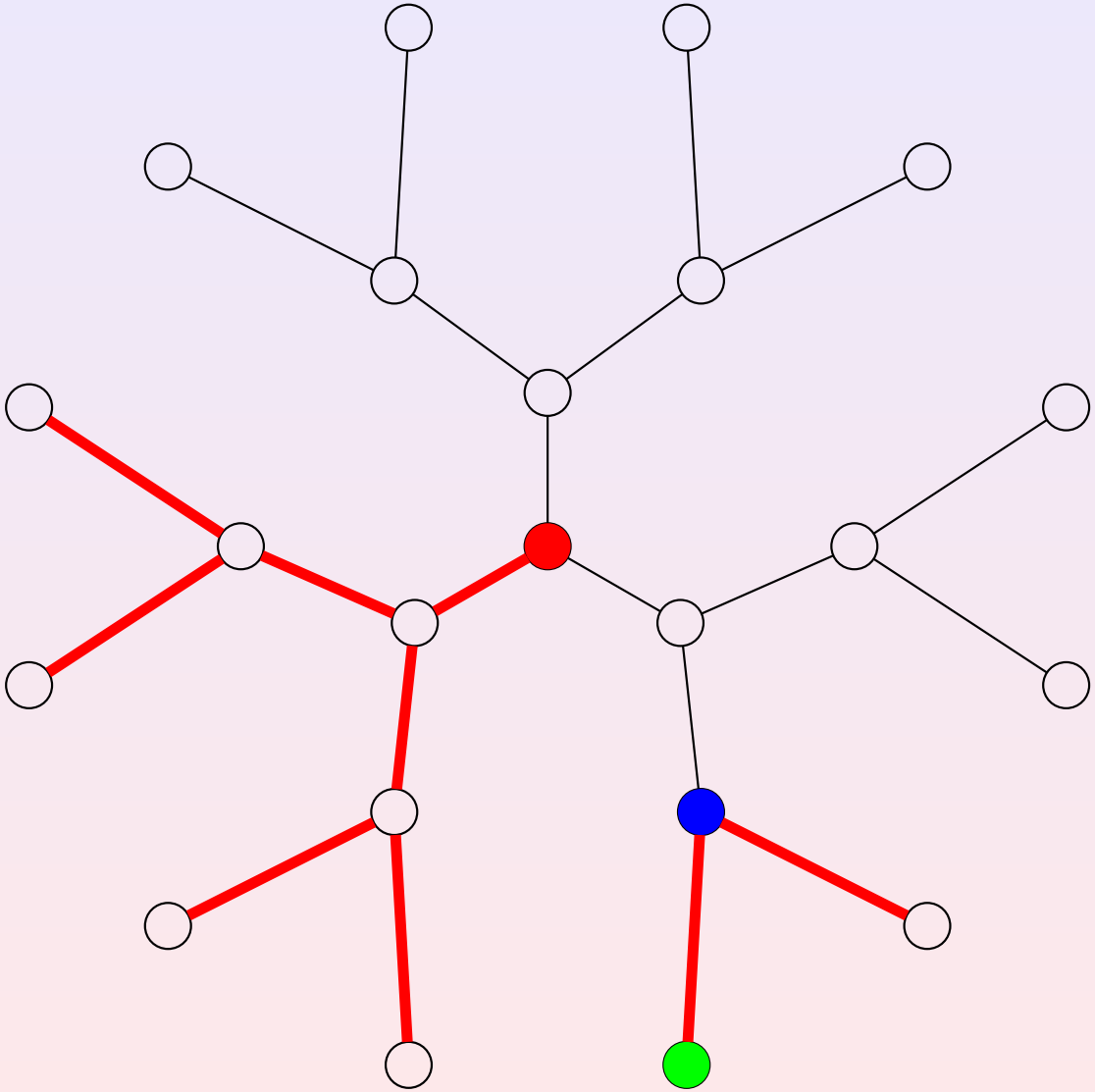
At any step, the cleared part of the graph must induce a connected subgraph.
Let $cs(G)$ be the connected search number of the graph G.



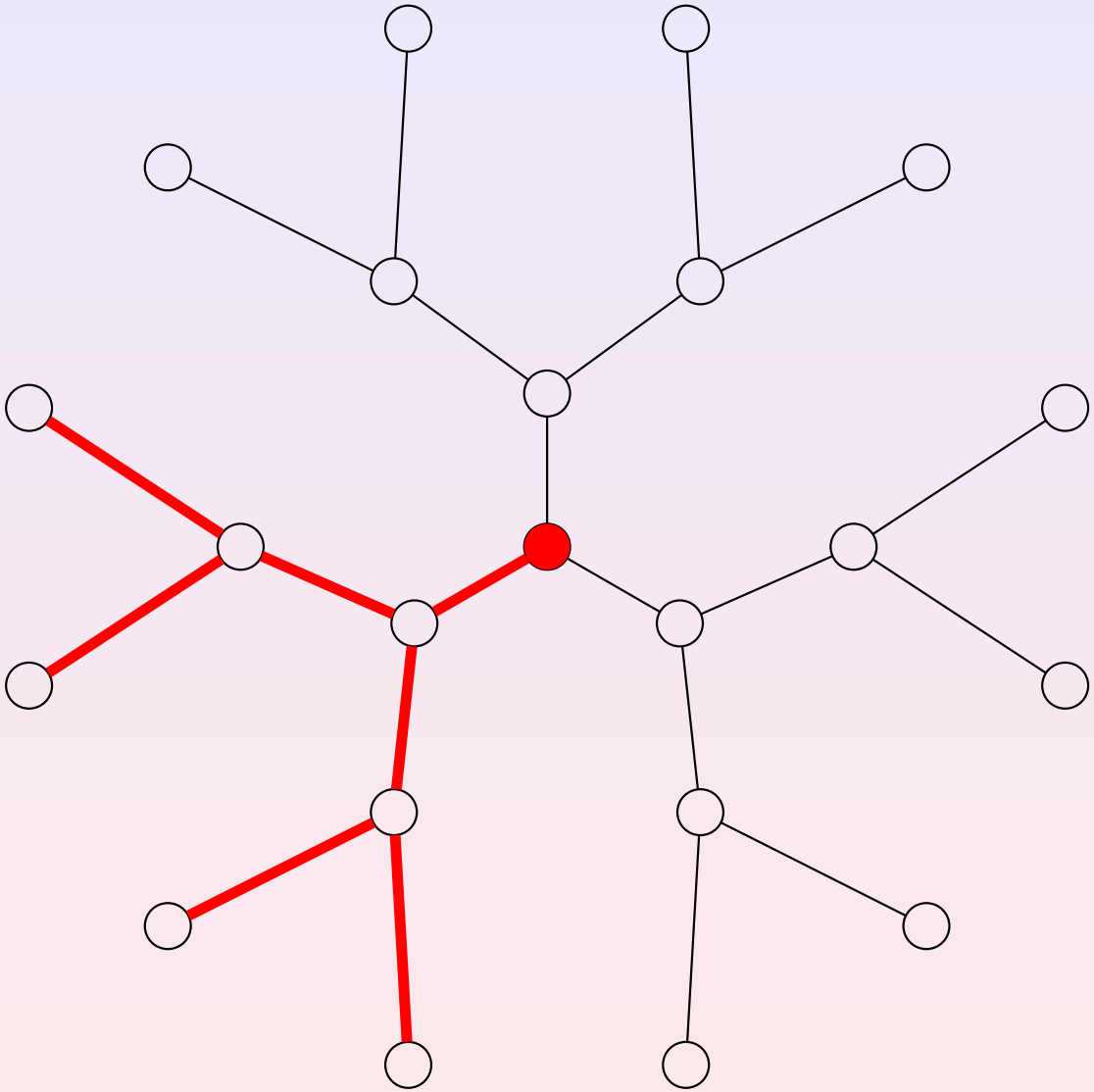
Back to a simple example



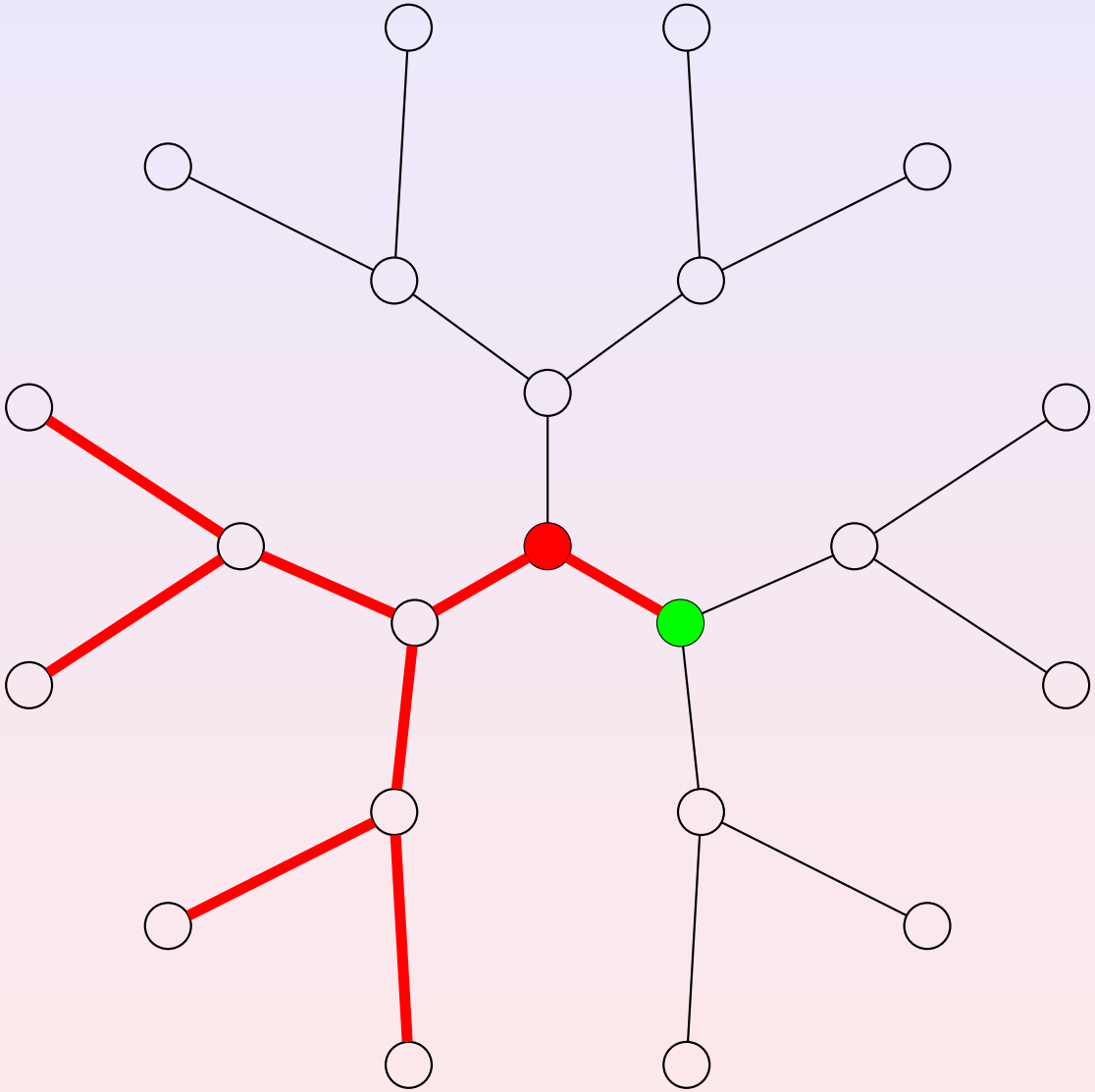
Back to a simple example



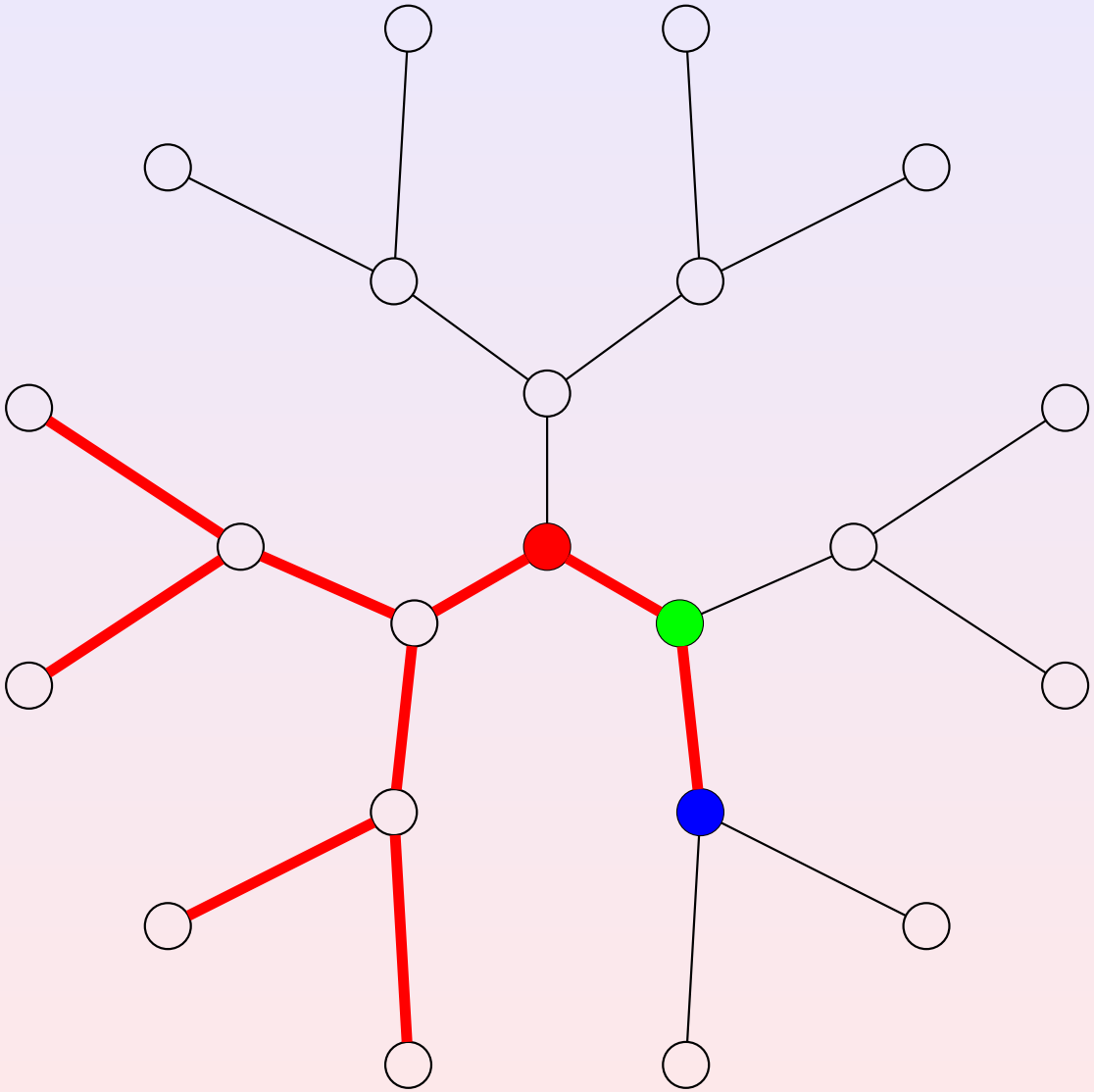
Back to a simple example



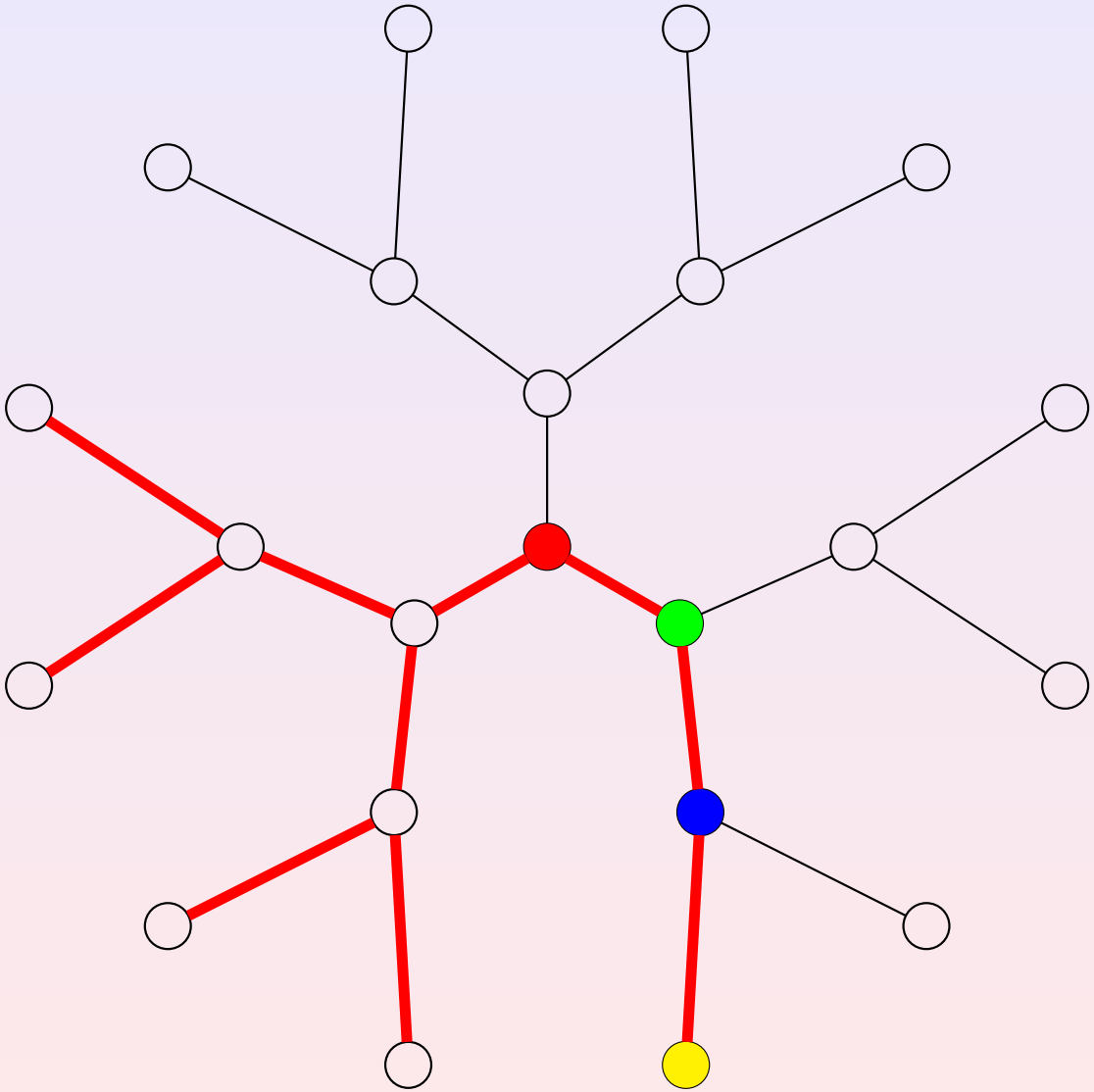
Back to a simple example



Back to a simple example

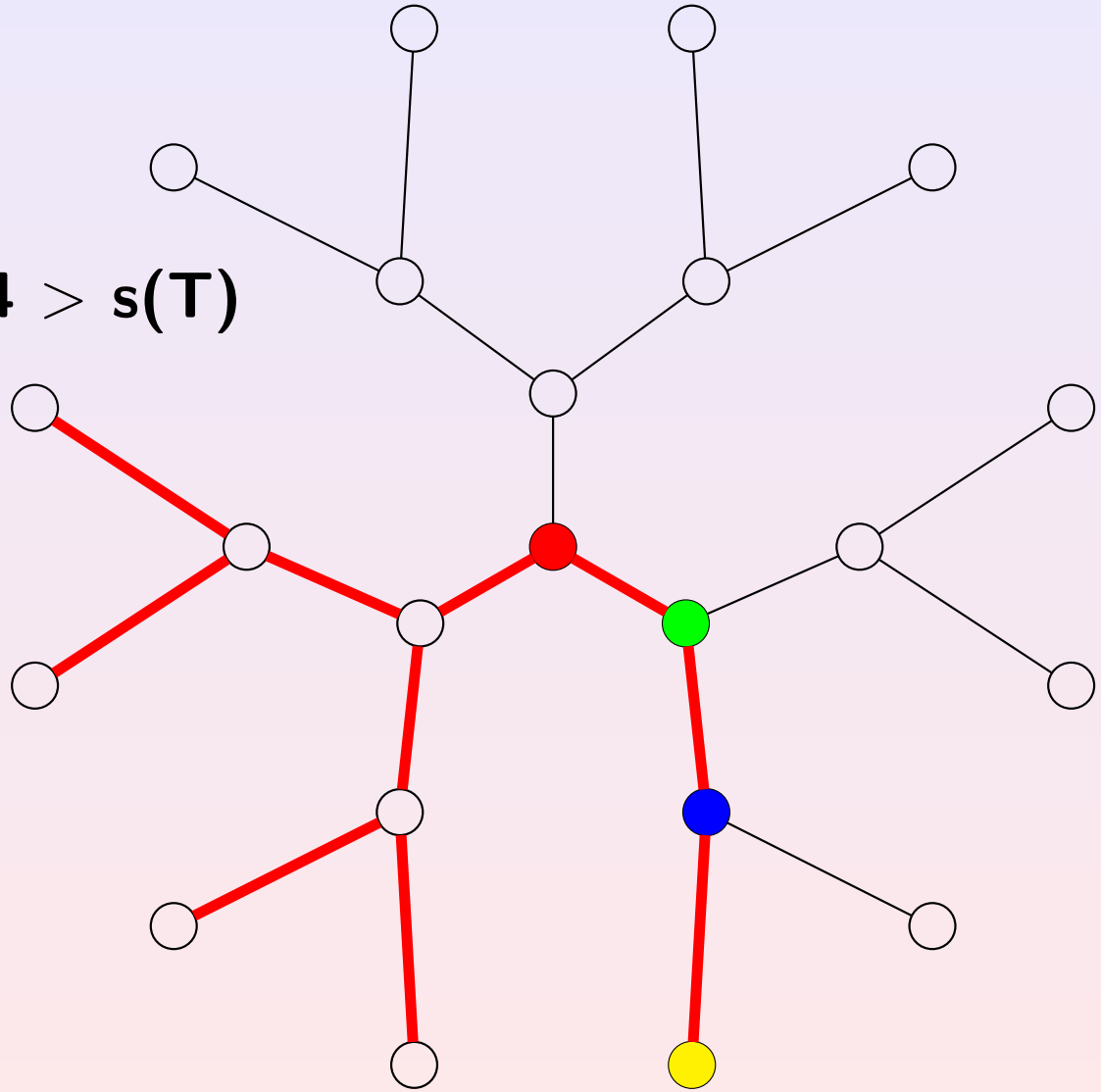


Back to a simple example



Back to a simple example

$\mathbf{cs(T) \geq 4 > s(T)}$



Price of connectedness

Barrière, Fraigniaud, Santoro and Thilikos. [WG, 2003]

For any tree T, $s(T) \leq cs(T) \leq 2\ s(T) - 2$.
Moreover, these bounds are tight.

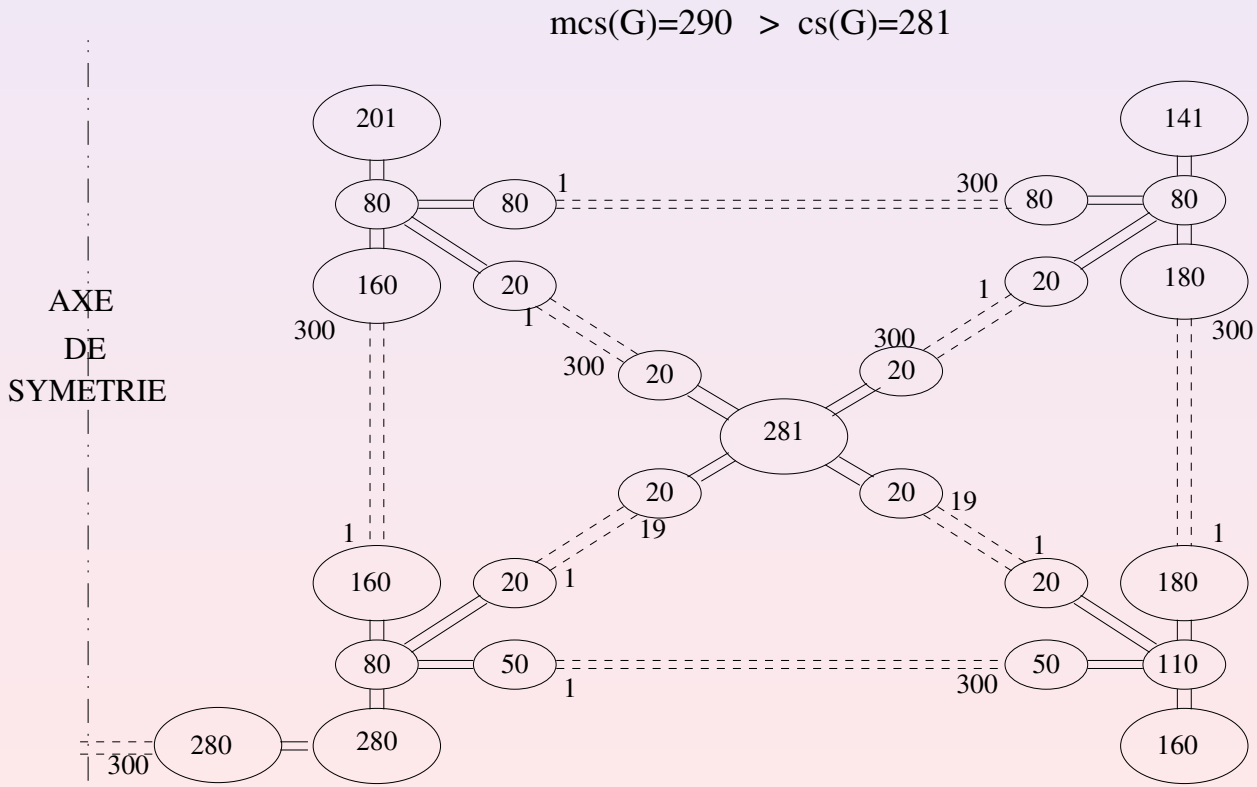
Fraigniaud and Nisse [LATIN, 2006]

For any n-node graph G, $s(G) \leq cs(G) \leq (1 + \log n)\ s(G)$.

Price of connectedness

In the connected case, Recontamination helps

Yang, Dyer, Alspach. Sweeping Graphs and Digraphs. [ISAAC04]



Related Work

Ratio monotone connected search over search

	in trees	in arbitrary n-node graphs
invisible fugitive	2(tight) [BFF^+03]	$O(\log n)$ [$FN06$]
visible fugitive	1 [trivial]	

An overview of monotony in graph searching

	search in arbitrary graphs	connected search in trees	connected search in arbitrary graphs
invisible fugitive	monotone [$BS91, L93$]	monotone [BFF^+02]	not monotone [$YDA04$]
visible fugitive	monotone [$ST93$]	monotone [trivial]	

Related Work

Ratio monotone connected search over search

	in trees	in arbitrary n-node graphs
invisible fugitive	2(tight) [BFF^+03]	$O(\log n)$ [$FN06$]
visible fugitive	1 [trivial]	$\Theta(\log n)$ **this paper**

An overview of monotony in graph searching

	search in arbitrary graphs	connected search in trees	connected search in arbitrary graphs
invisible fugitive	monotone [$BS91, L93$]	monotone [BFF^+02]	not monotone [$YDA04$]
visible fugitive	monotone [$ST93$]	monotone [trivial]	

Related Work

Ratio monotone connected search over search

	in trees	in arbitrary n-node graphs
invisible fugitive	2(tight) [BFF^+03]	$O(\log n)$ [$FN06$]
visible fugitive	1 [trivial]	$\Theta(\log n)$ **this paper**

An overview of monotony in graph searching

	search in arbitrary graphs	connected search in trees	connected search in arbitrary graphs
invisible fugitive	monotone [$BS91, L93$]	monotone [BFF^+02]	not monotone [$YDA04$]
visible fugitive	monotone [$ST93$]	monotone [trivial]	not monotone **this paper**

Visible Connected Graph Searching

Theorem 1

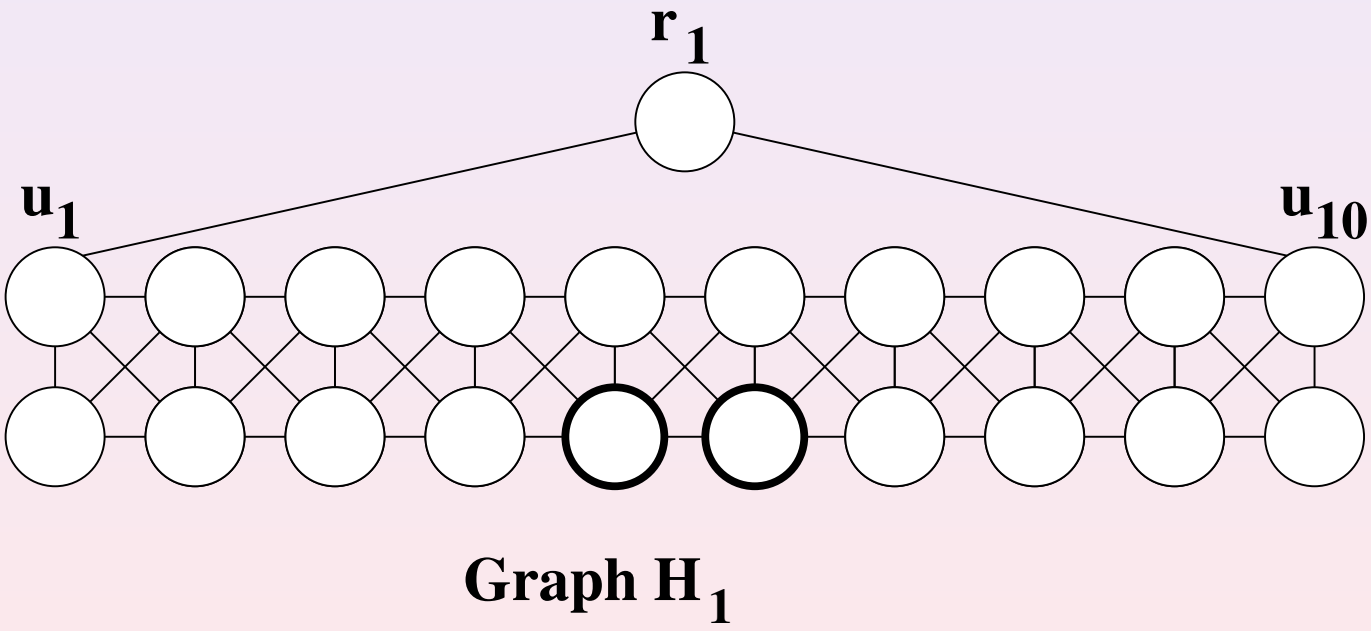
For any n_0, there is $n \geq n_0$ and an n-node graph G such that any monotone connected visible search for G uses at least $\Omega(\mathbf{vs}(G) \cdot \log n)$ searchers.

More precisely

We propose a sequence of n_i-node graph $(\mathcal{G}_i)_{i \in N}$, such that, the sequence $(n_i)_{i \in N}$ is strictly increasing, and, for any $i > 1$, $\mathbf{vs}(\mathcal{G}_i) \leq 5$ and $\mathbf{mcvs}(\mathcal{G}_i) \geq \Omega(\log n_i)$.

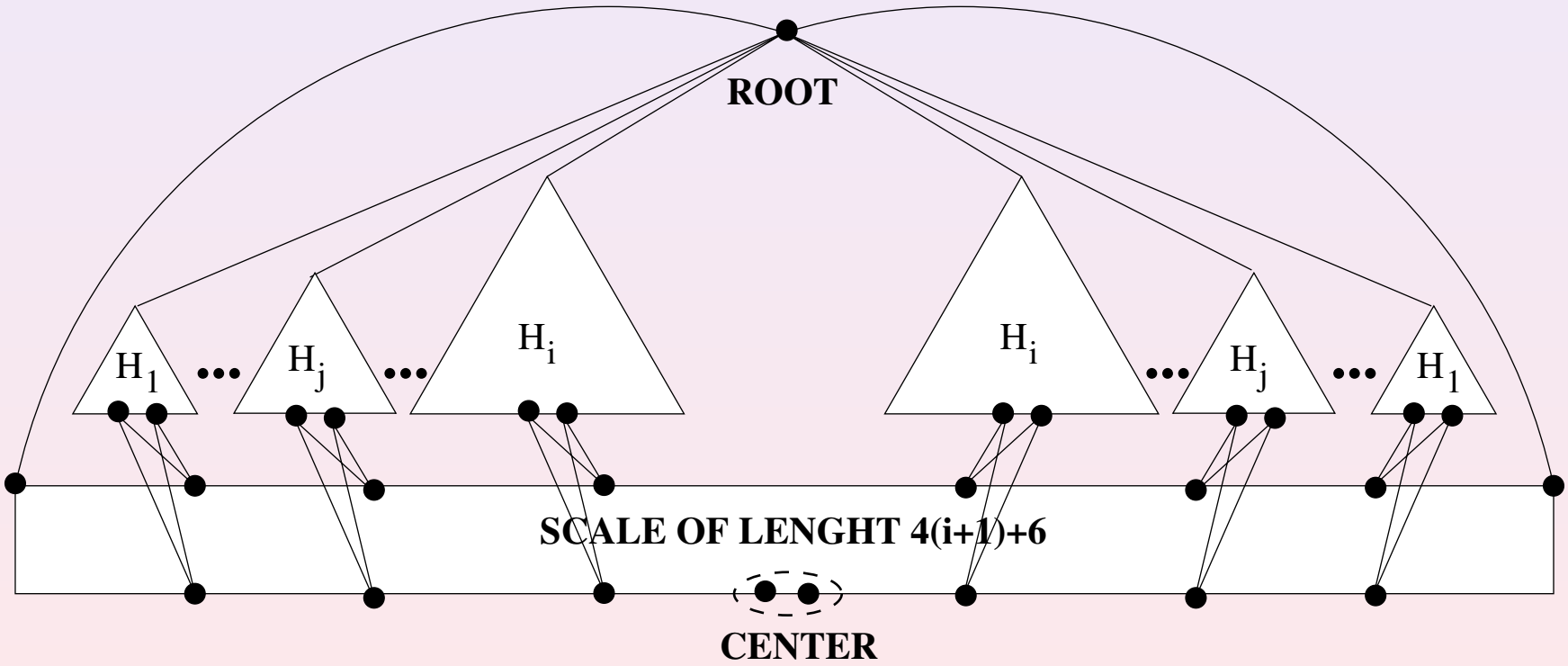
Squetch of the proof by induction on i

The sequence of graphs $(H_i)_{i \in N}$ is defined recursively.
H_1 consists of a scale of length 10 plus a root r_1.



Squetch of the proof by induction on i

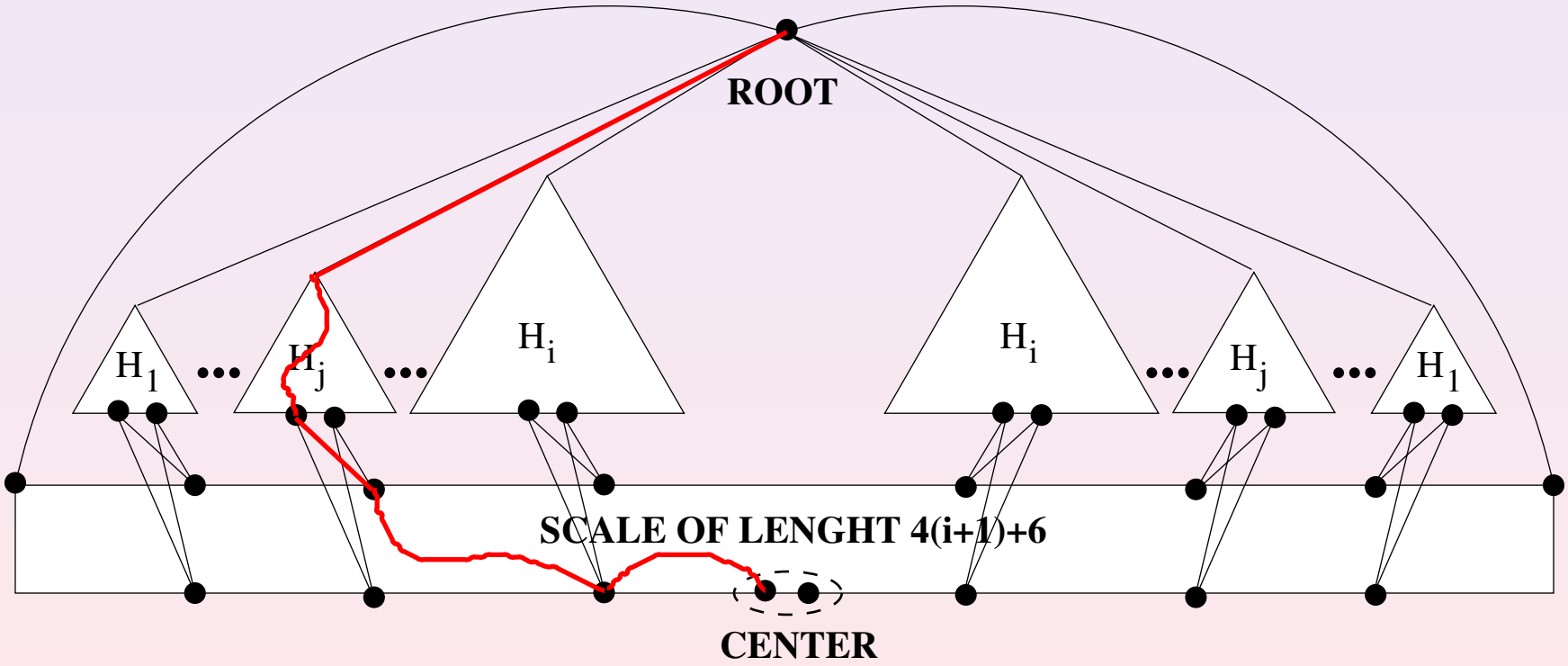
For any $i > 1$, the graph H_{i+1} is built
using 2 copies of $H_1, \cdots, H_i$ and a scale of length $4(i+1)+6$.



Graph H_{i+1}

Squetch of the proof by induction on i

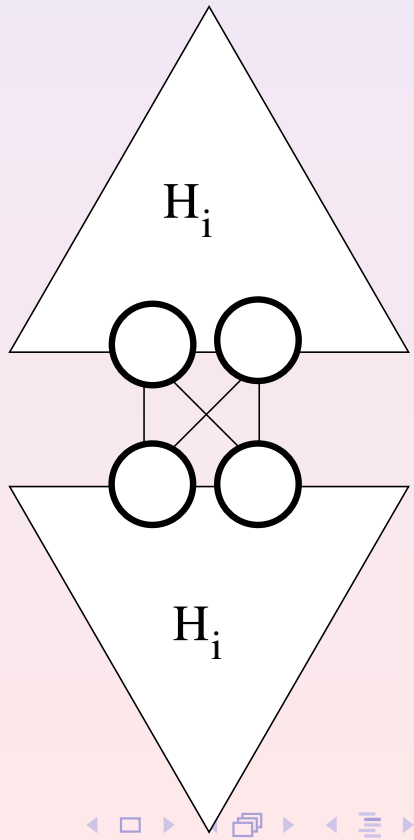
Lemma: Catching the fugitive in the root, starting from the center requires at least $2i$ searchers.



Graph H_{i+1}

Squetch of the proof by induction on i

For any $i \geq 1$, $\mathcal{G}_i$ consists of 2 copies of H_i linked by a 4-clique.

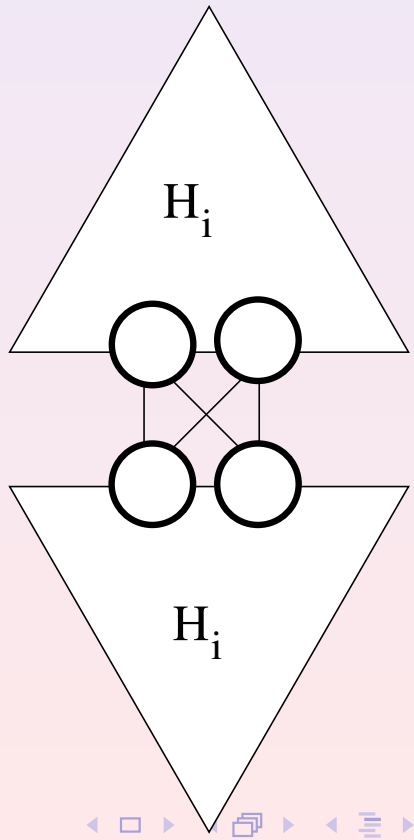


Squetch of the proof by induction on i

For any $i \geq 1$, $\mathcal{G}_i$ consists of 2 copies of H_i linked by a 4-clique.

Visible search of G_i

Using tree-decomposition, we prove by induction that $vs(G_i) \leq 5$.



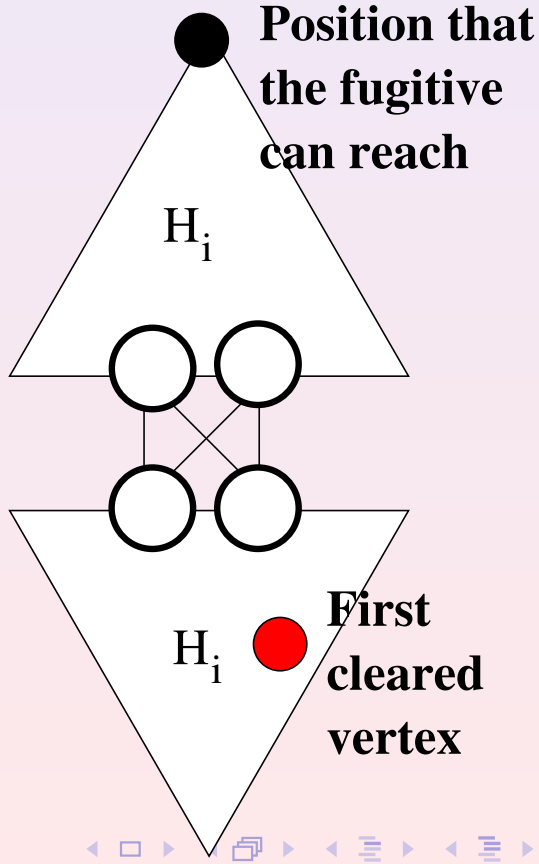
Squetch of the proof by induction on i

For any $i \geq 1$, $\mathcal{G}_i$ consists of 2 copies of H_i linked by a 4-clique.

Visible search of G_i

Using tree-decomposition, we prove by induction that $vs(G_i) \leq 5$.

Whatever the first cleared vertex is, the fugitive can flee in the other copy of H_i.



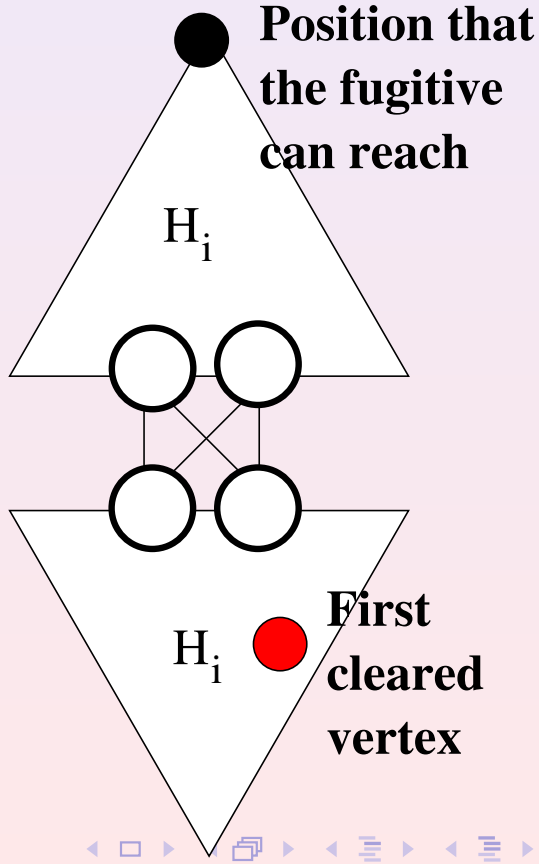
Squetch of the proof by induction on i

For any $i \geq 1$, $\mathcal{G}_i$ consists of 2 copies of H_i linked by a 4-clique.

Visible search of G_i

Using tree-decomposition, we prove by induction that $vs(G_i) \leq 5$.

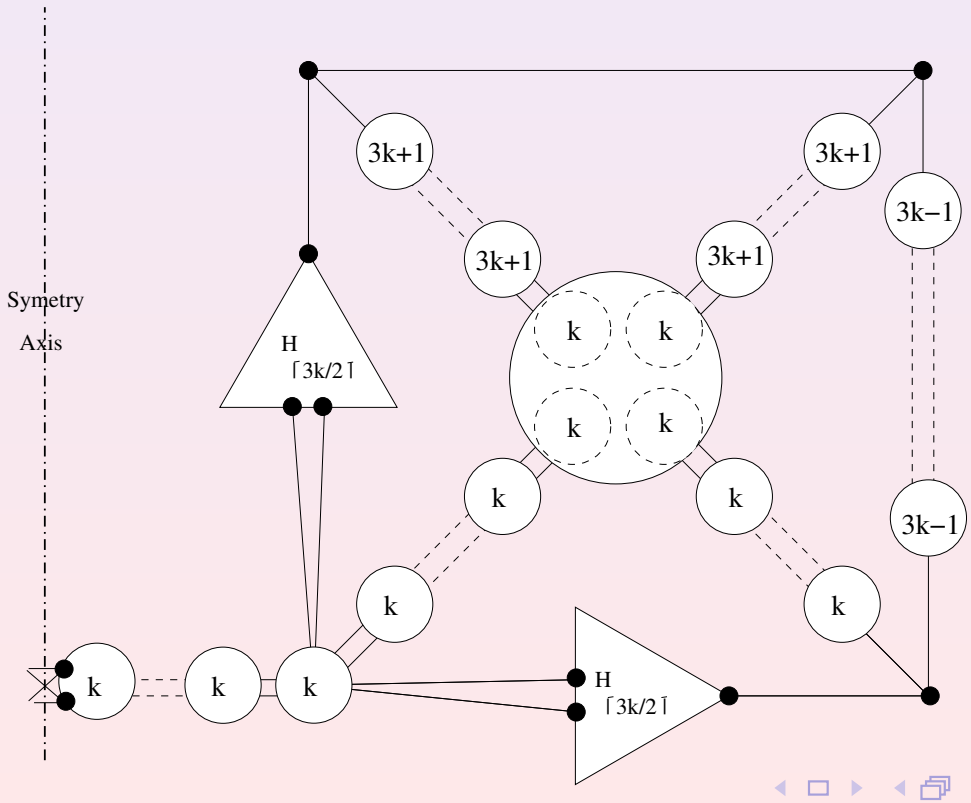
Whatever the first cleared vertex is, the fugitive can flee in the other copie of H_i.
Thus, $2i$ searchers are required, i.e. about $\mathbf{vs}(G_i) \log n$ searchers.



Monotony in Visible Connected Graph Searching

Theorem 2

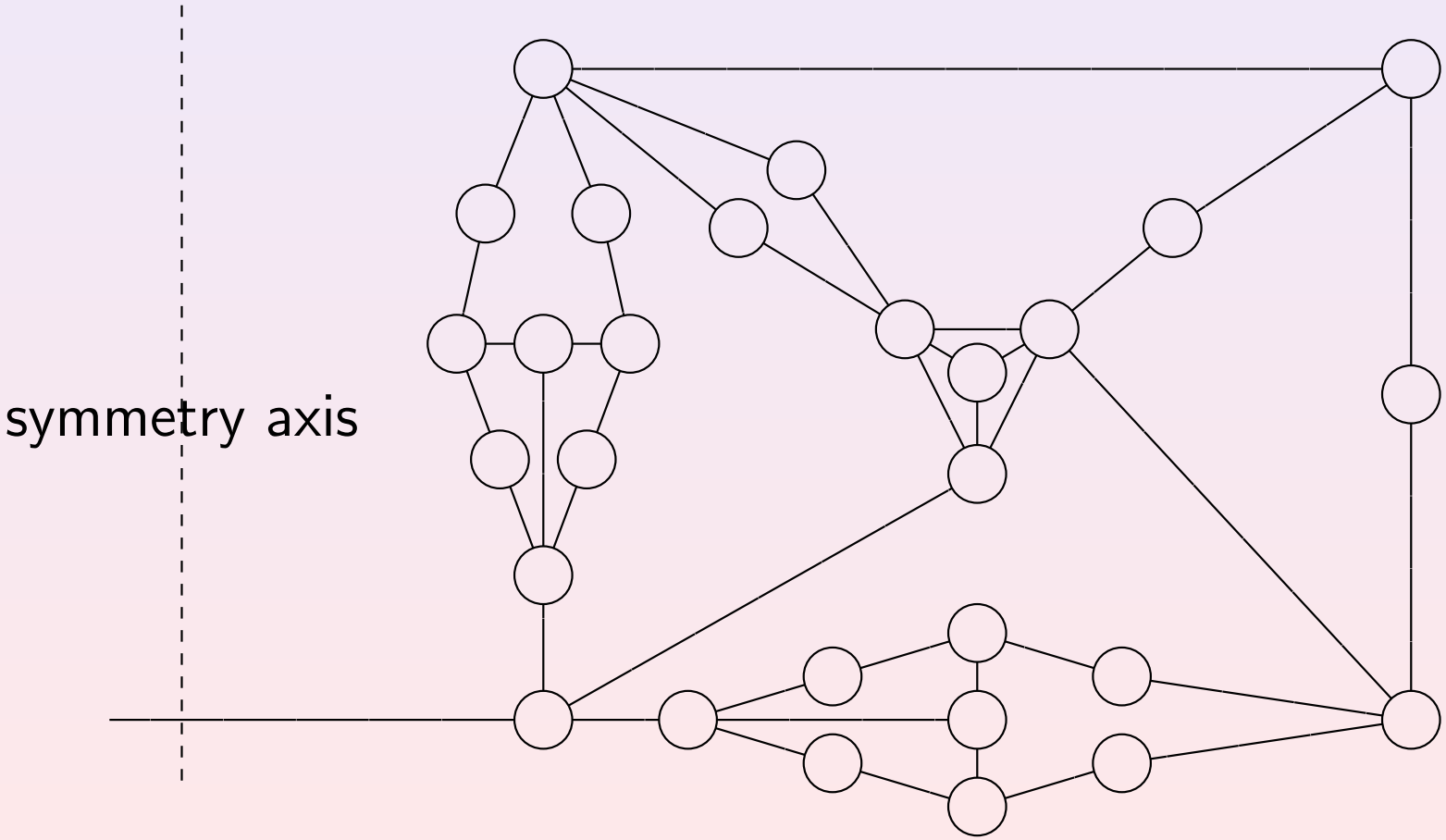
For any $k \geq 4$, there exists a graph G such that $\mathbf{cvs}(G) = 4k + 1$ and any monotone connected visible search strategy uses at least $4k + 2$ searchers.



Monotony in Visible Connected Graph Searching

Recontamination helps

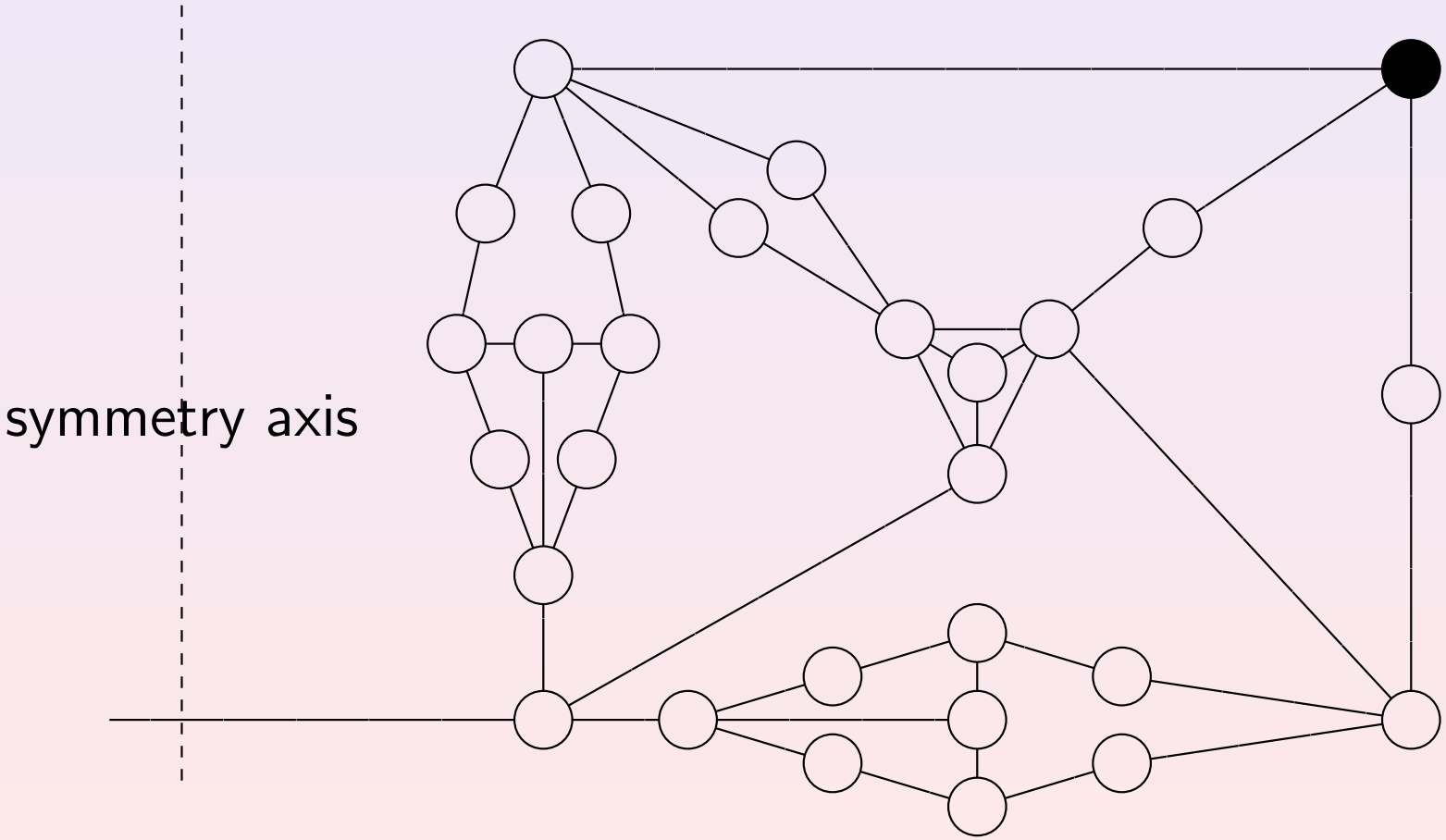
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

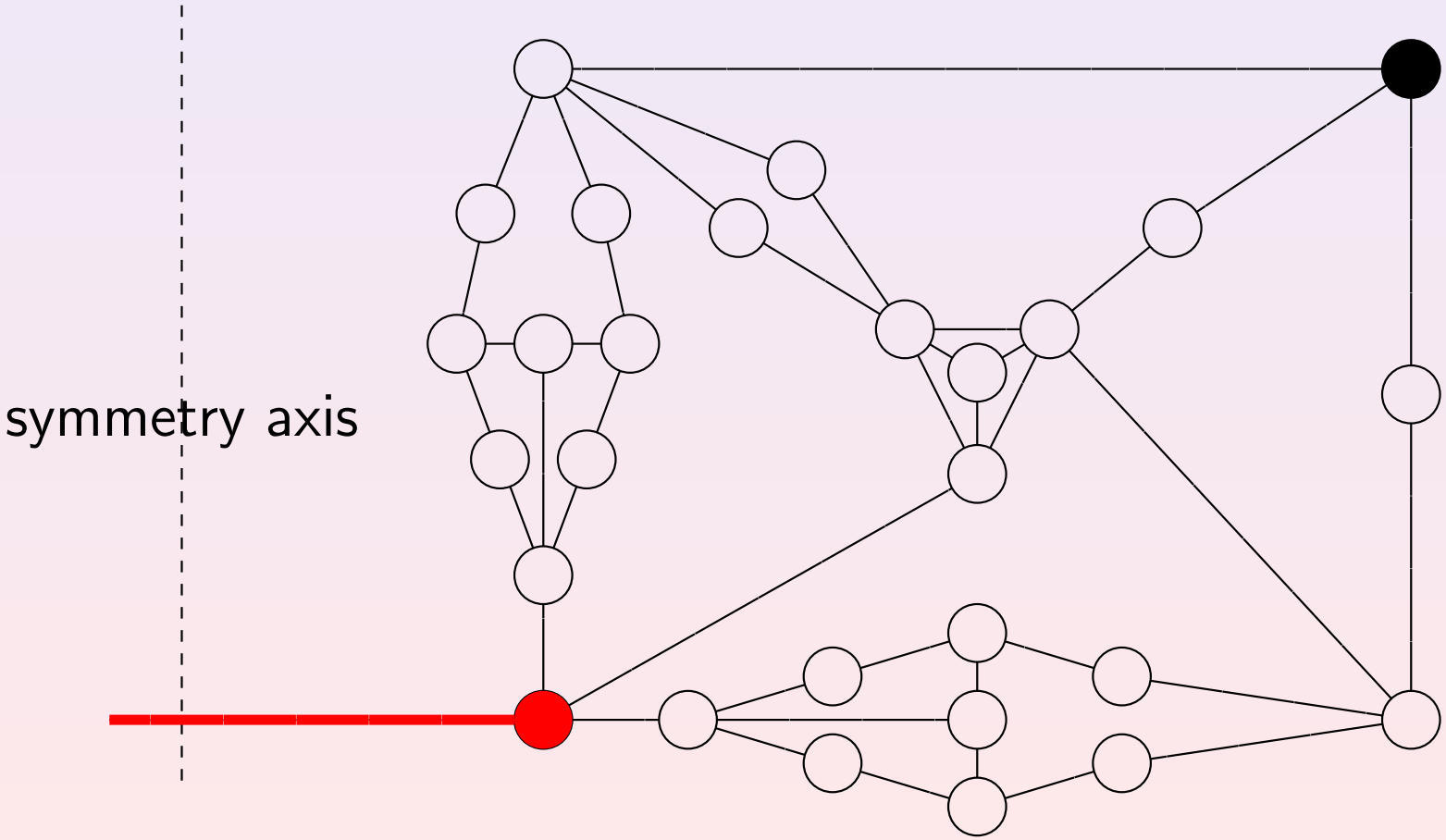
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

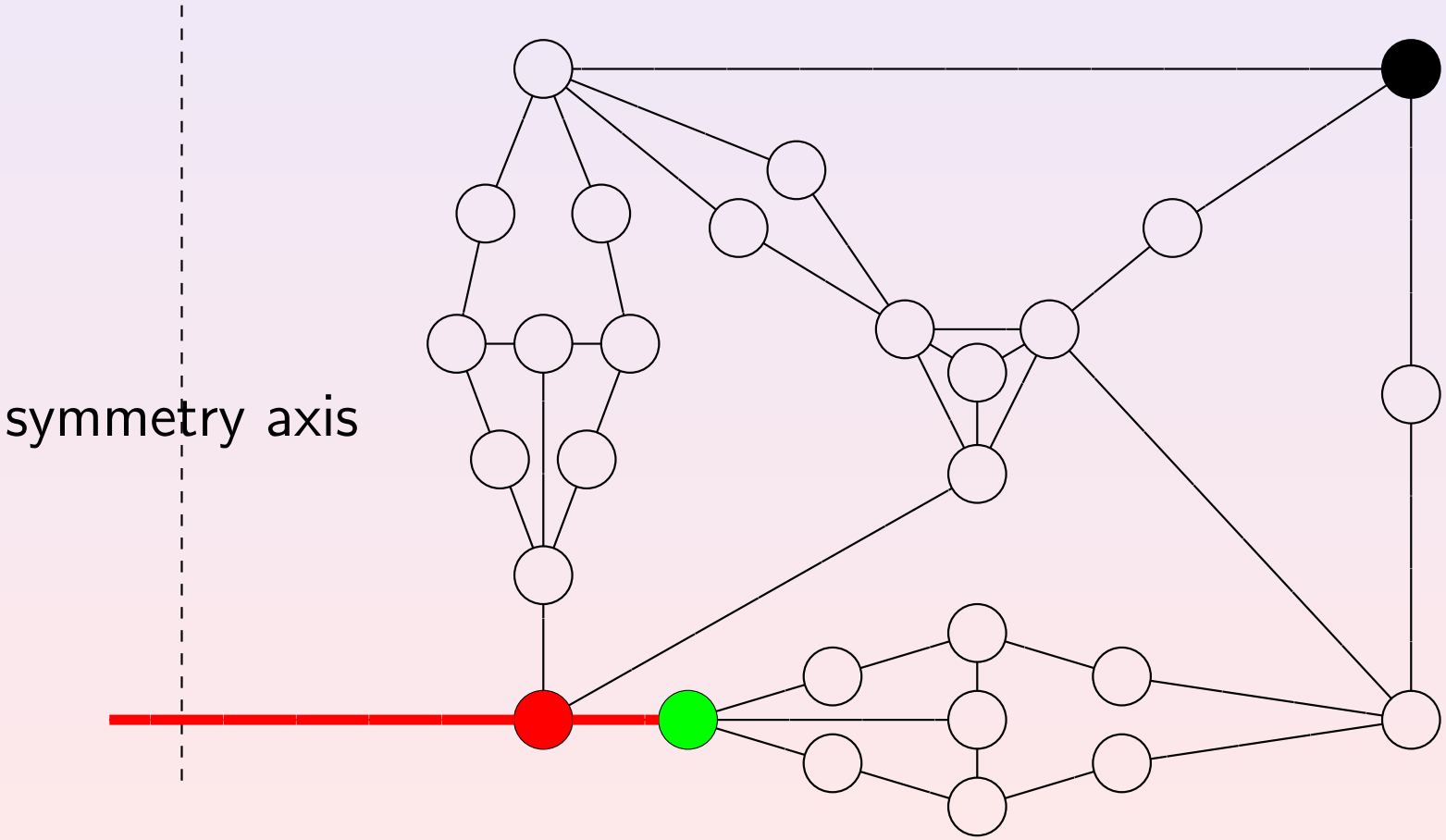
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

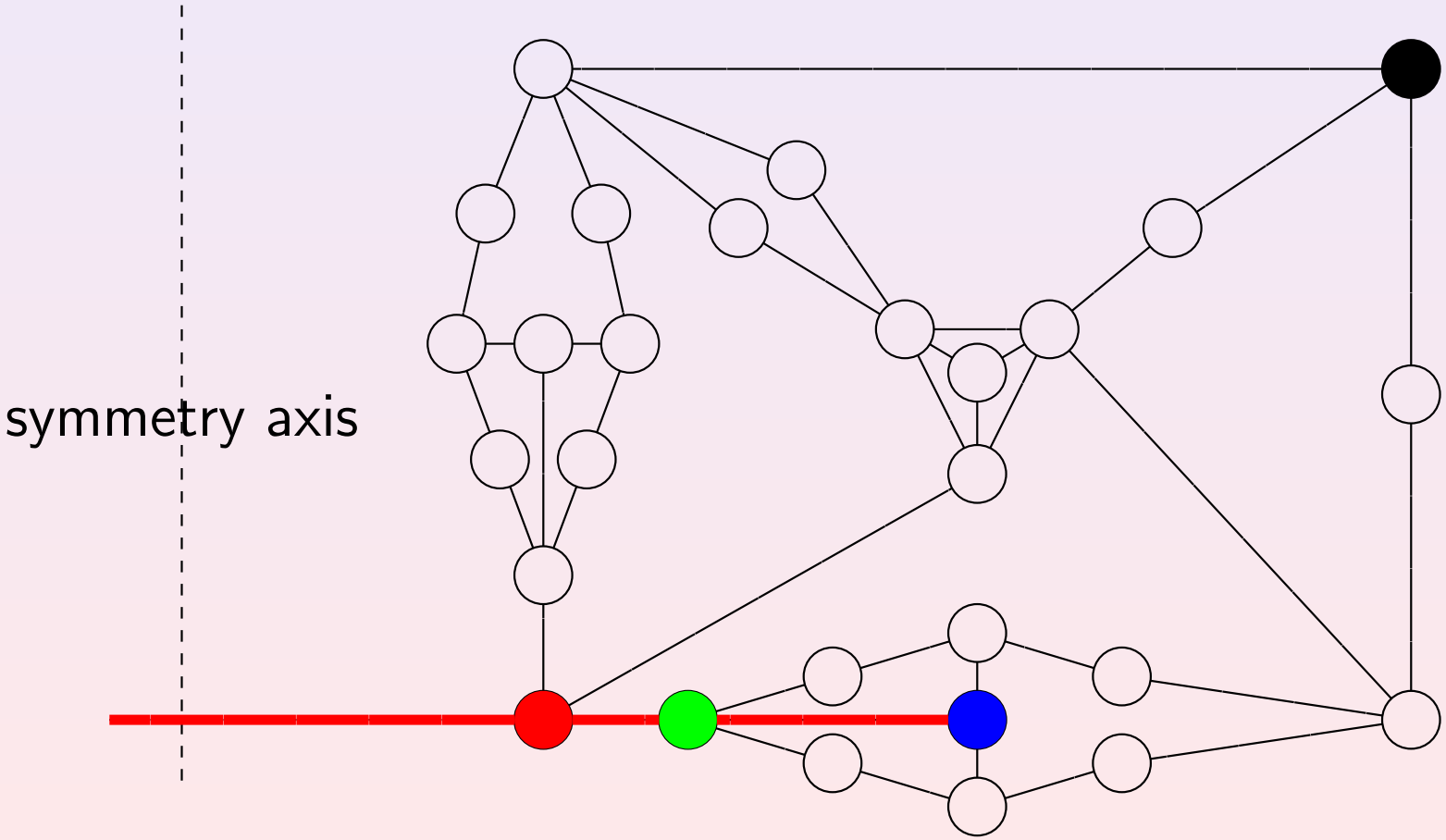
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

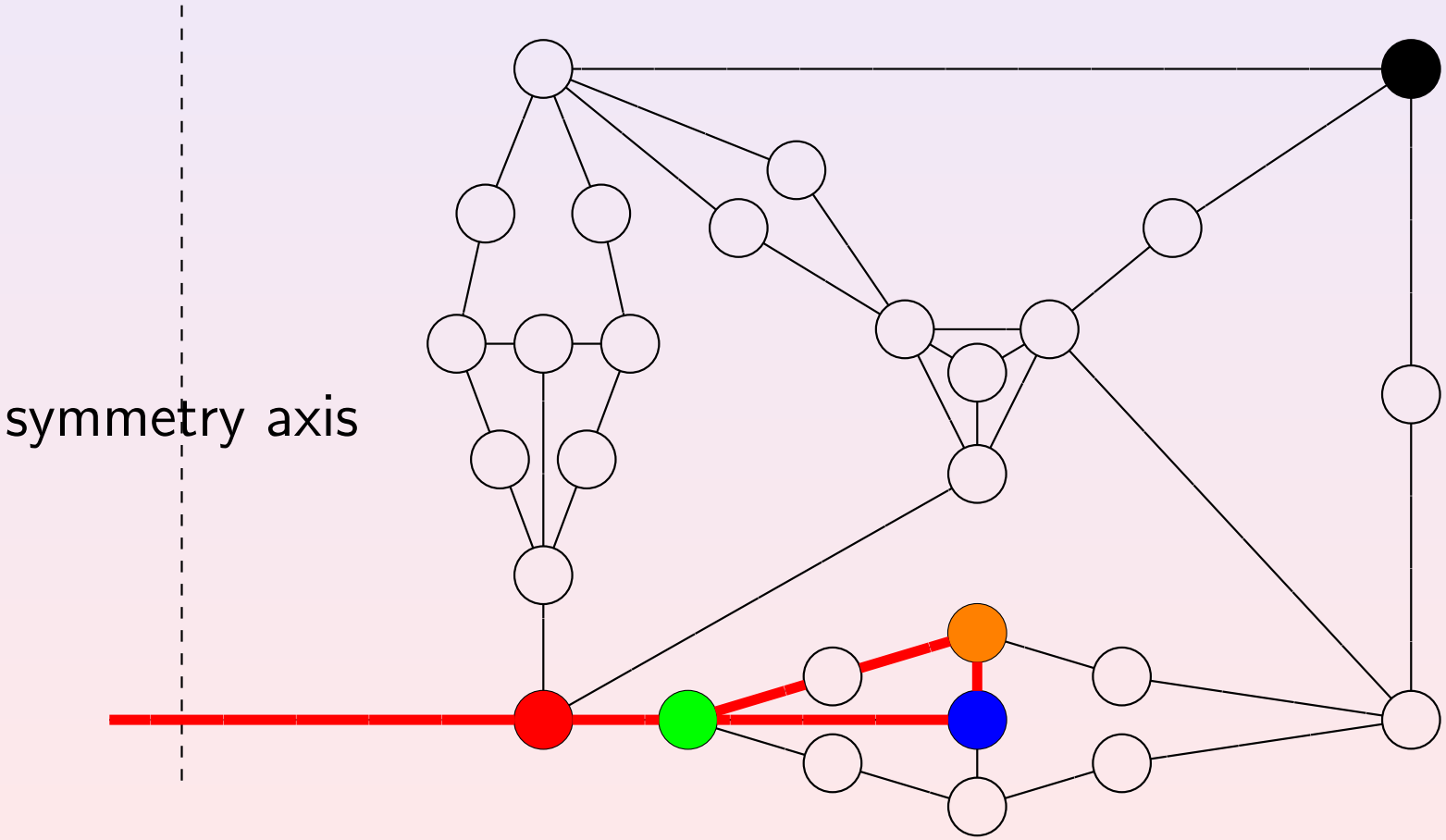
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

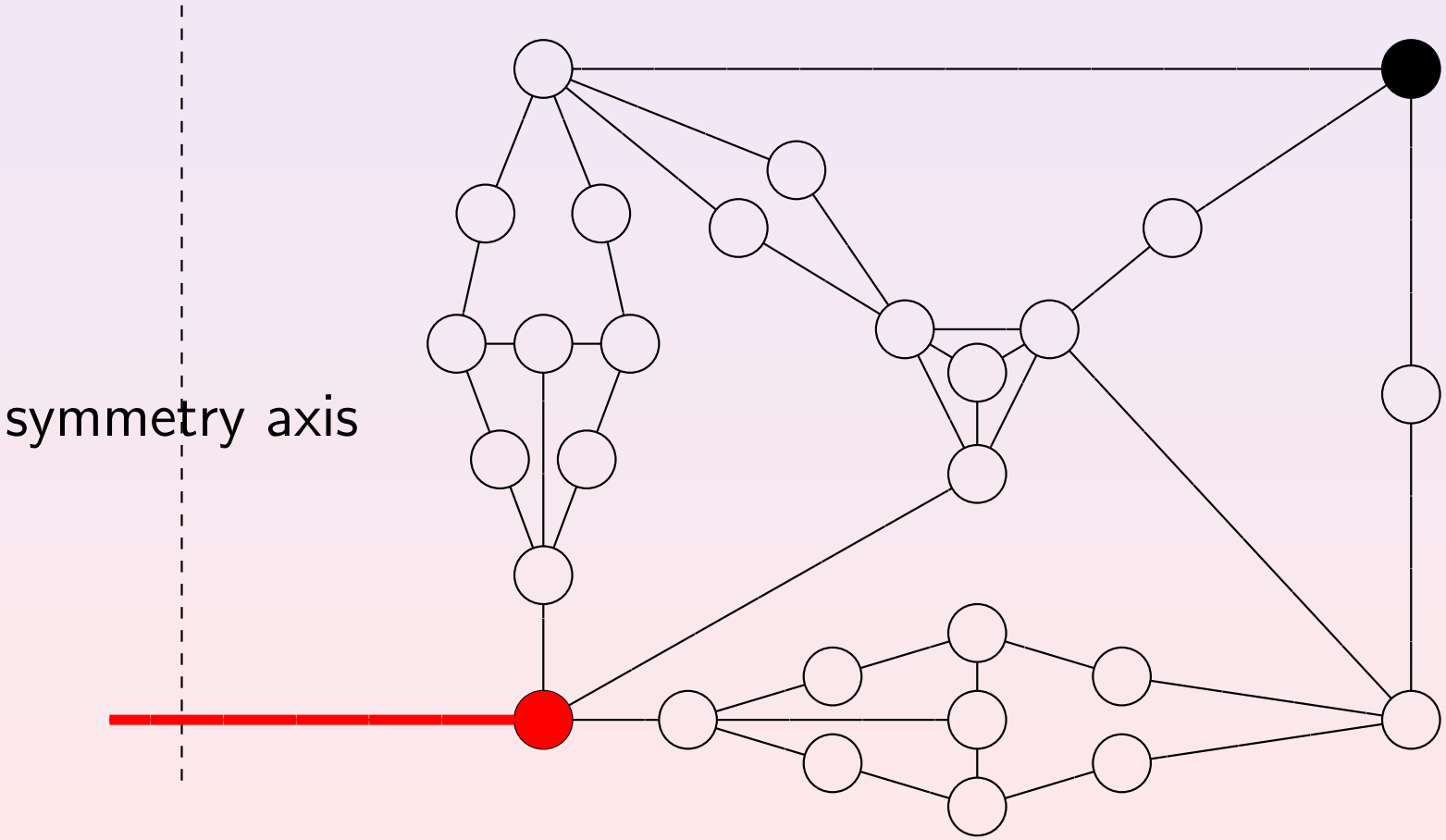
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

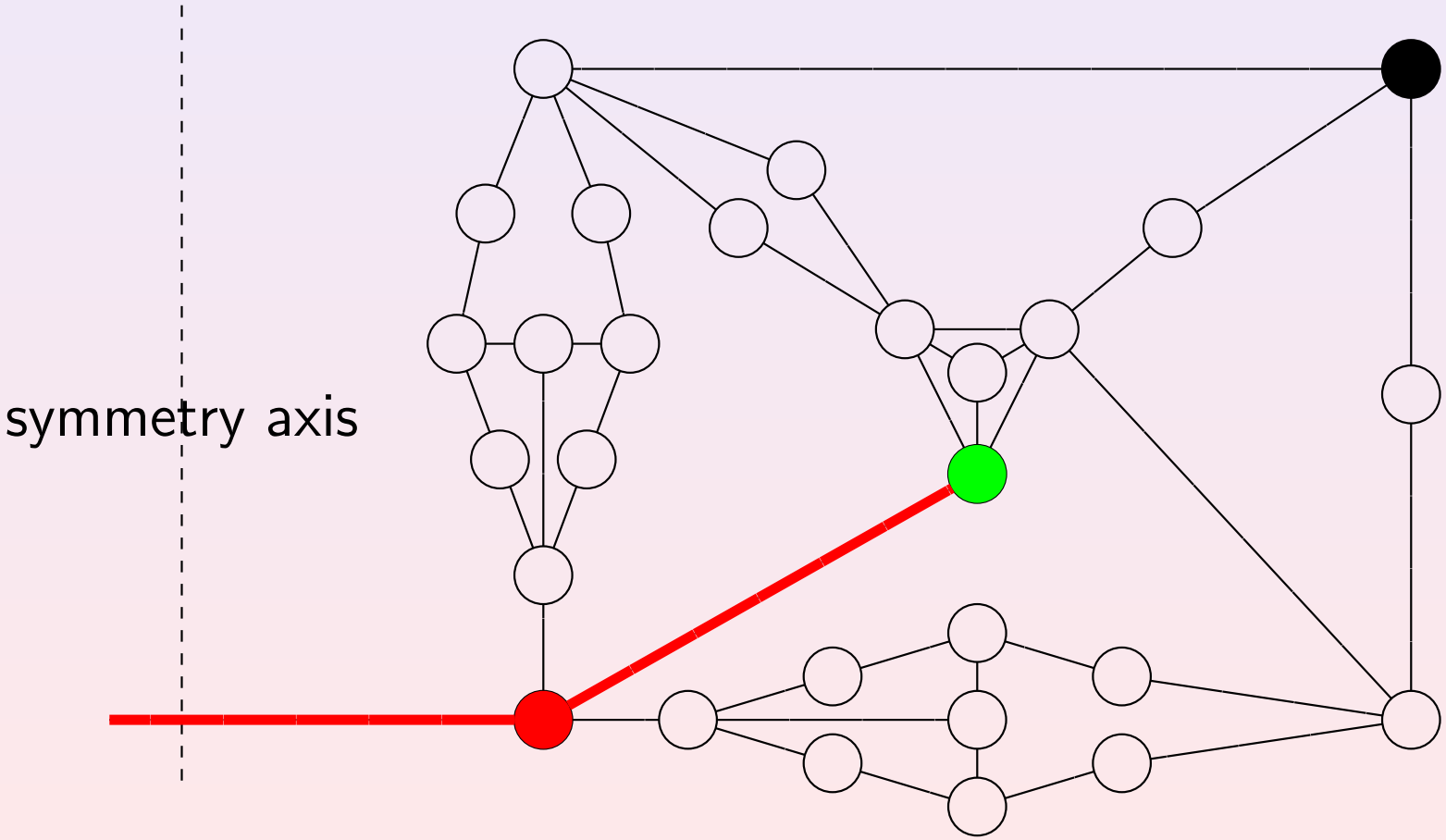
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

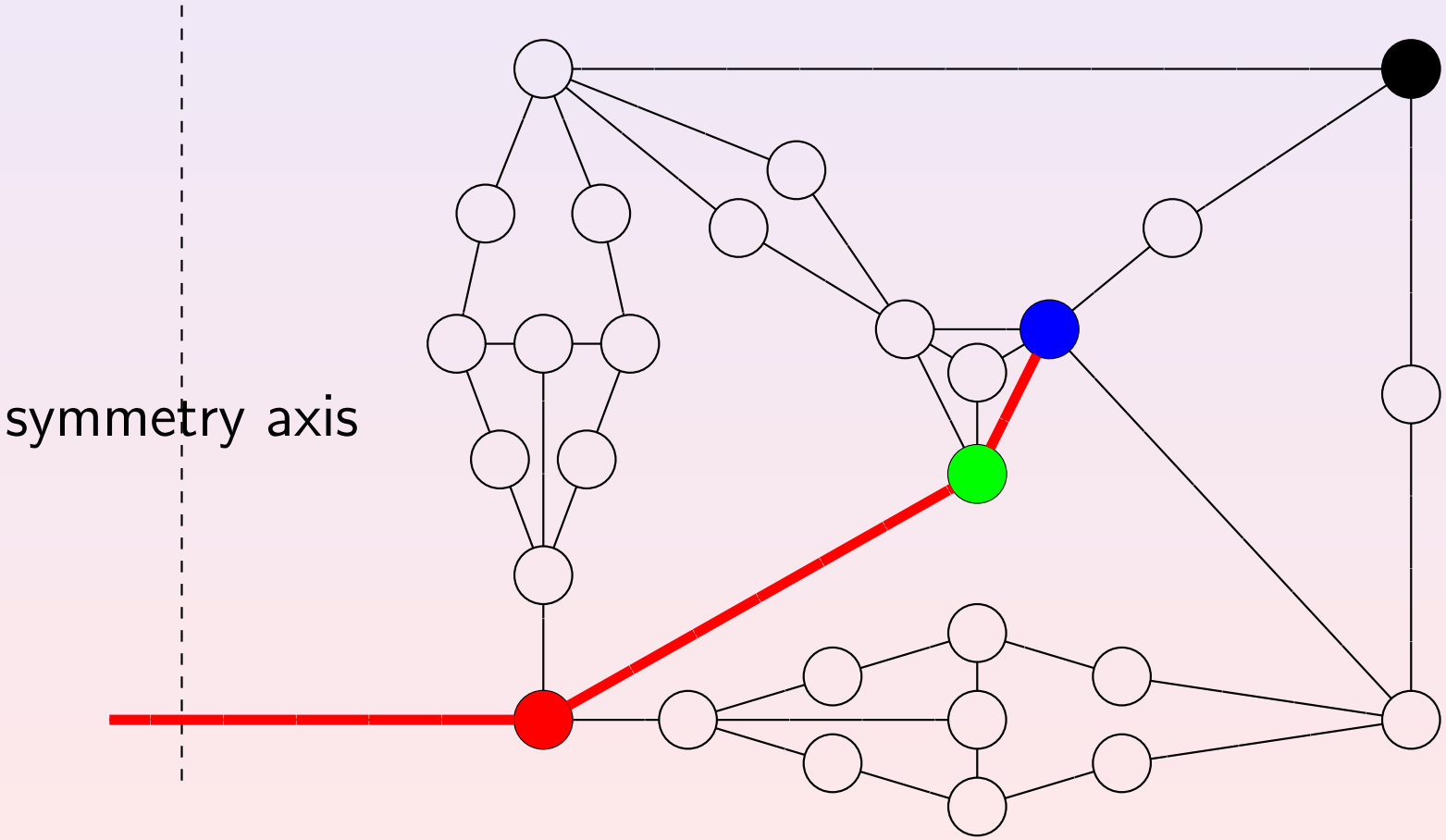
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

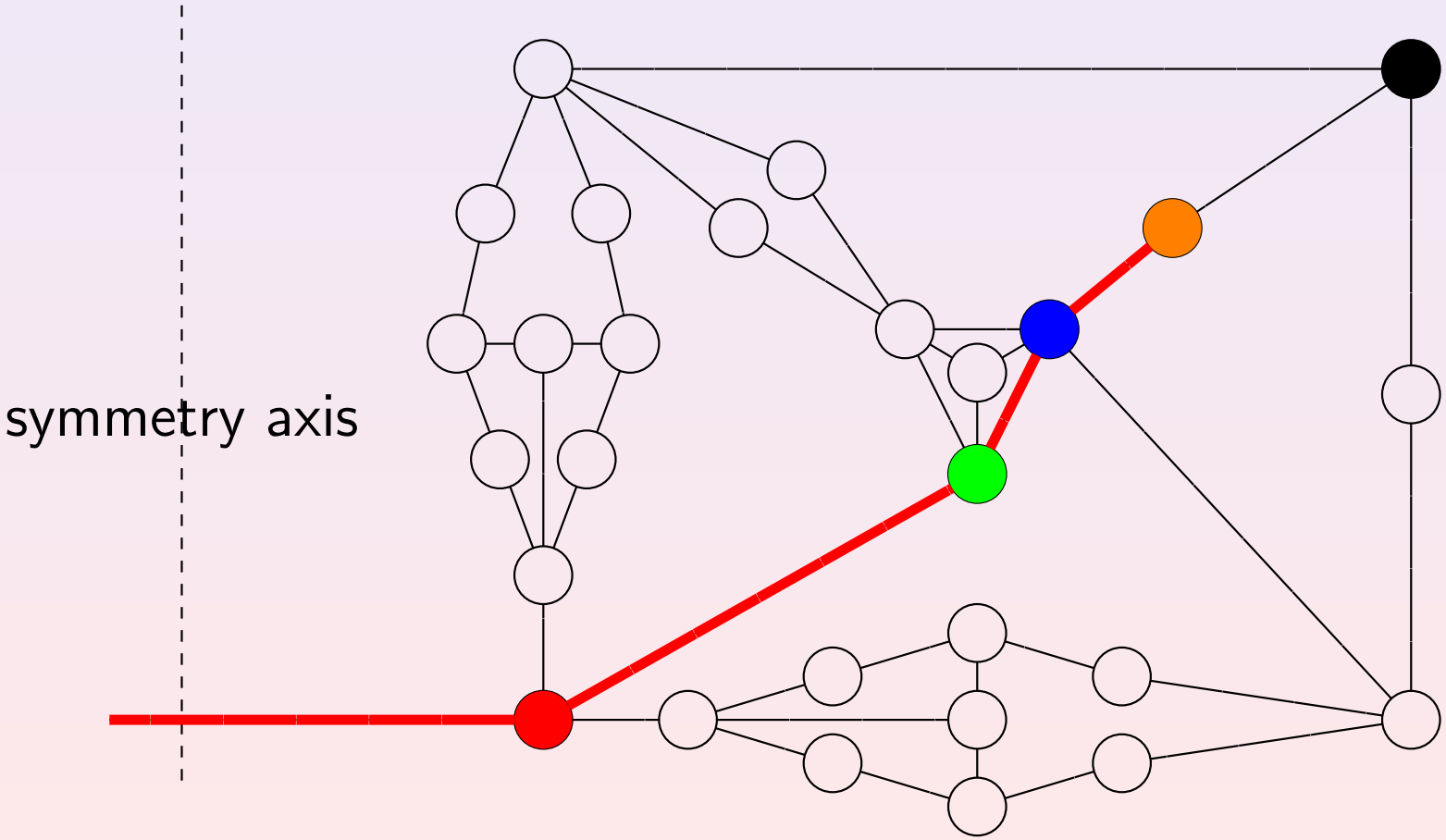
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

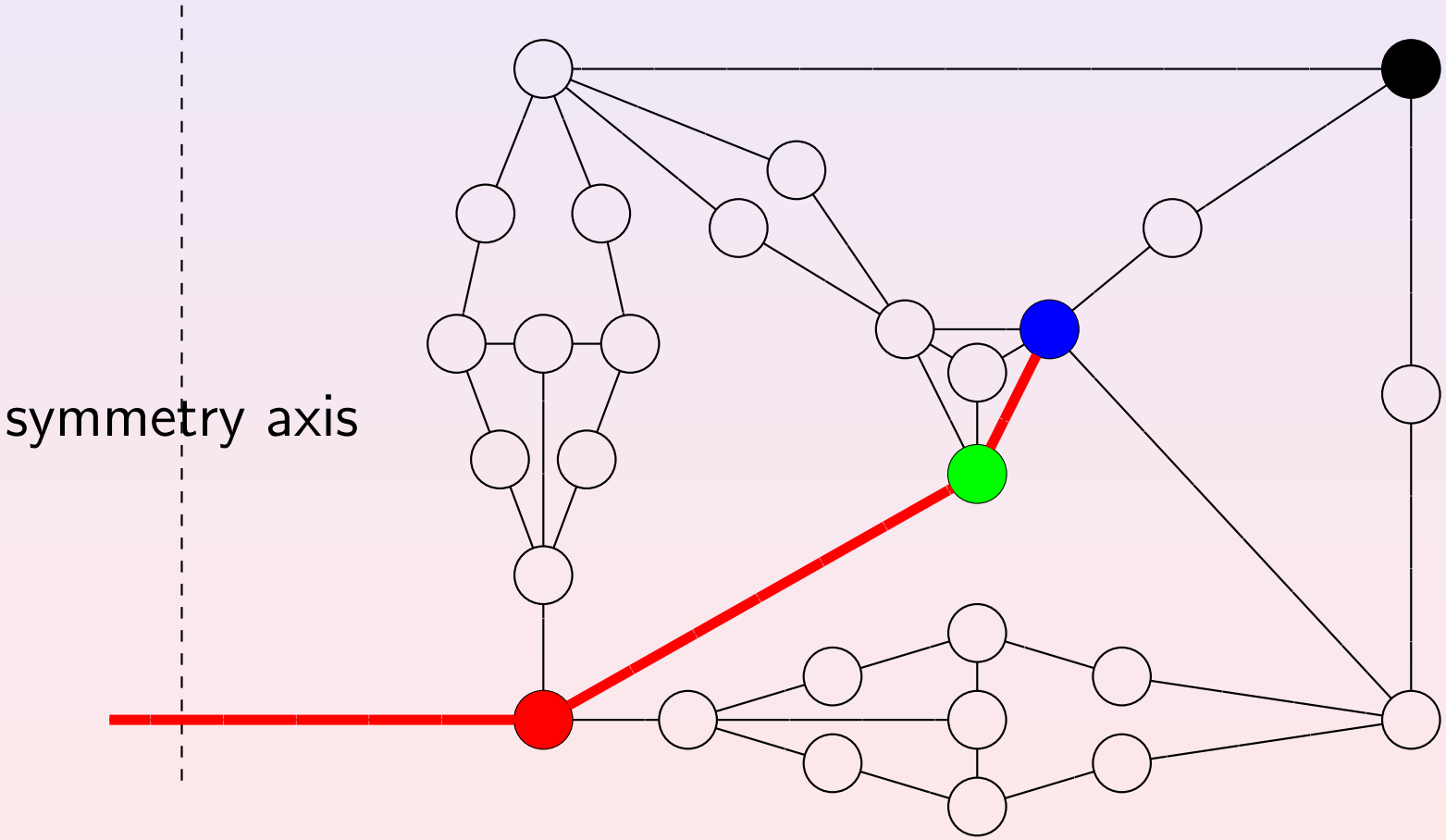
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

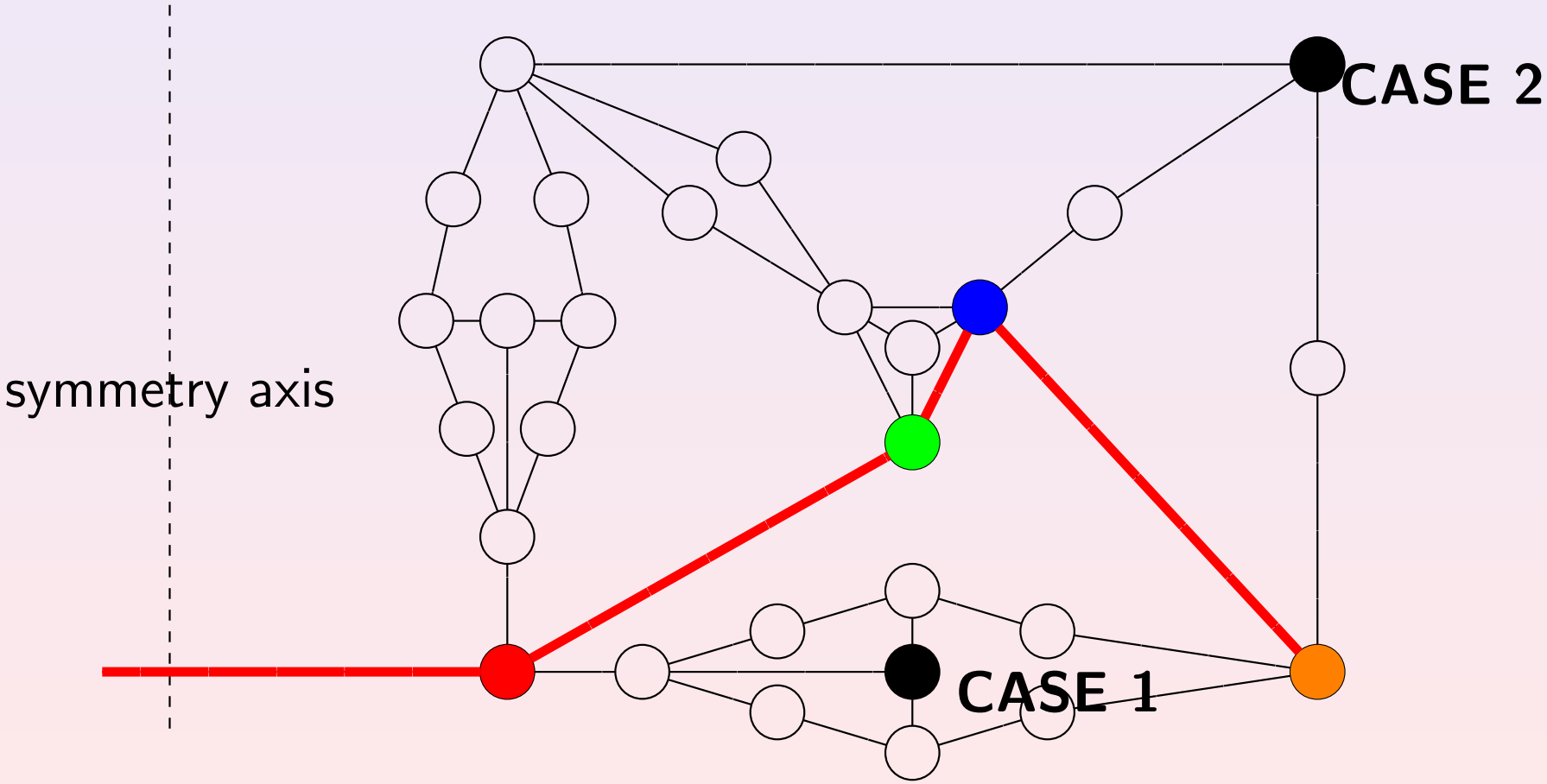
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

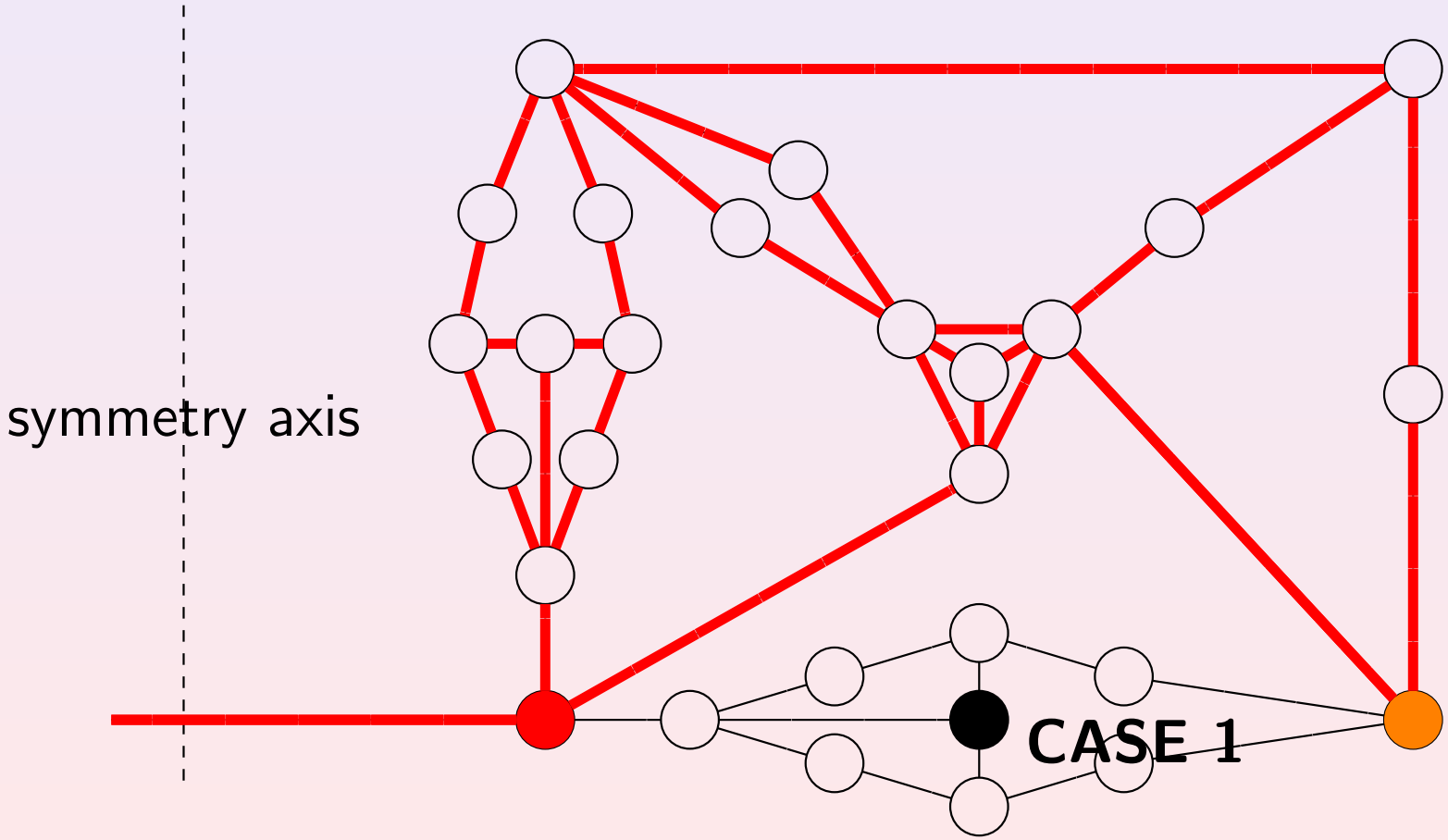
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

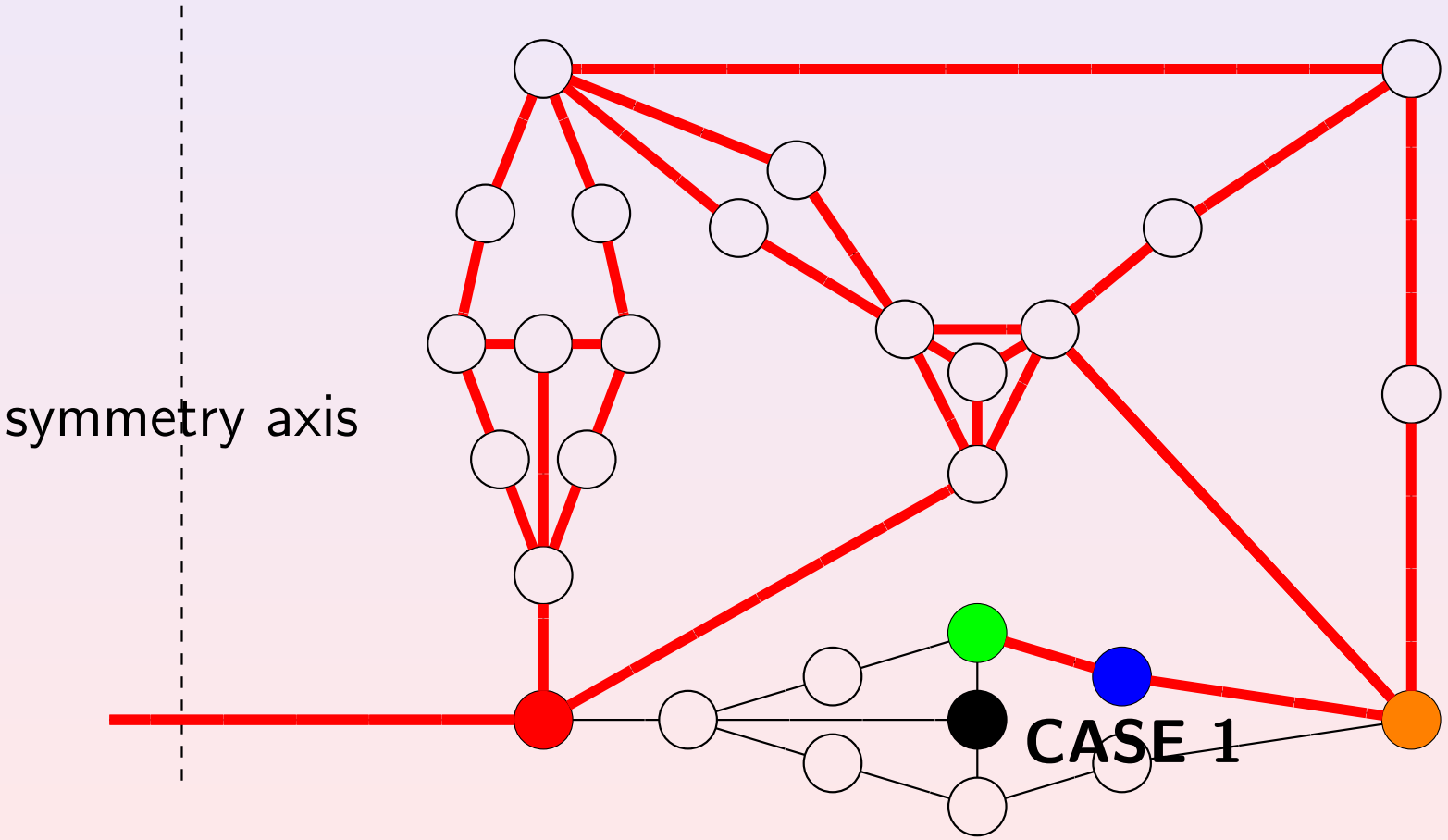
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

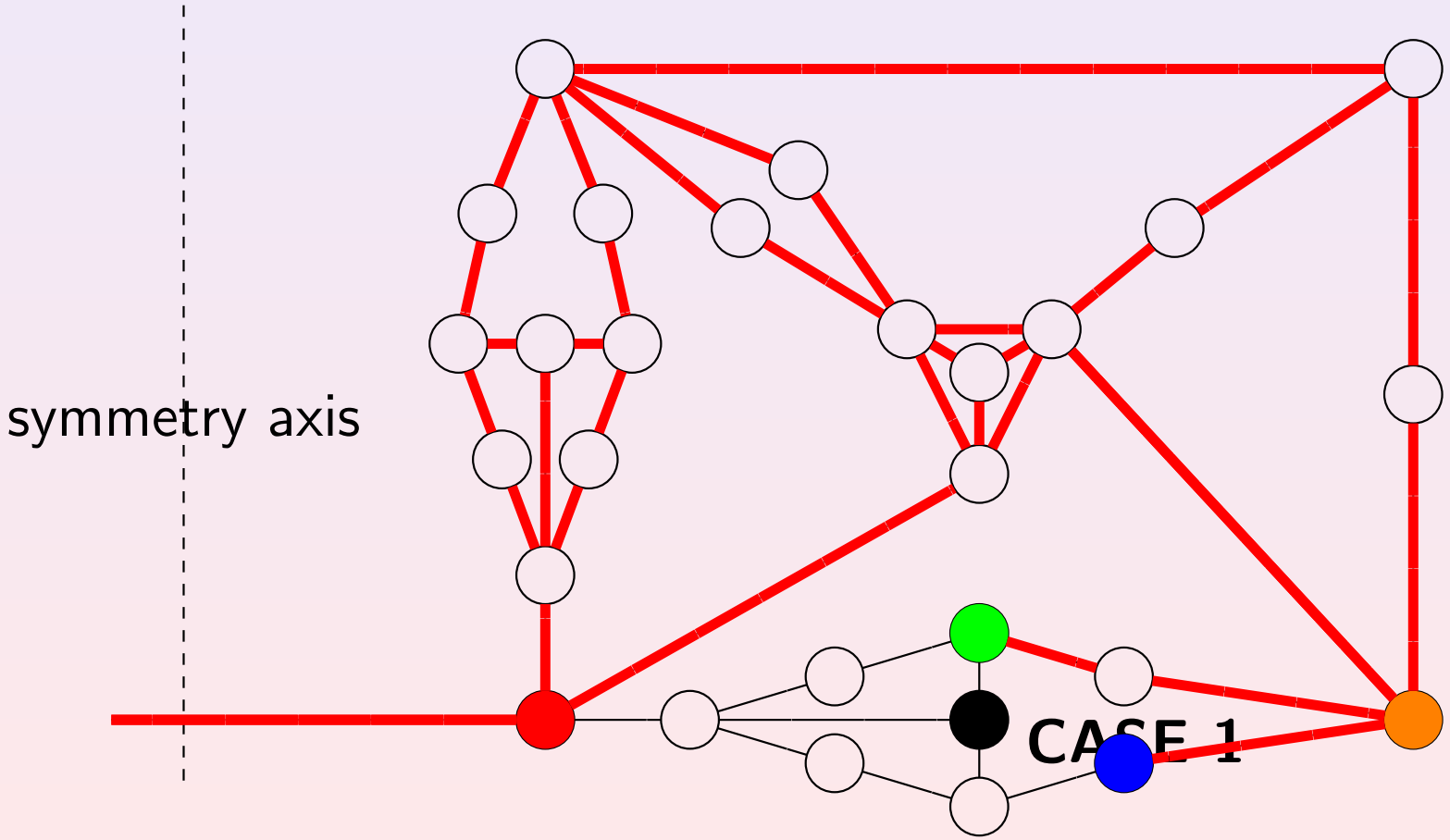
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

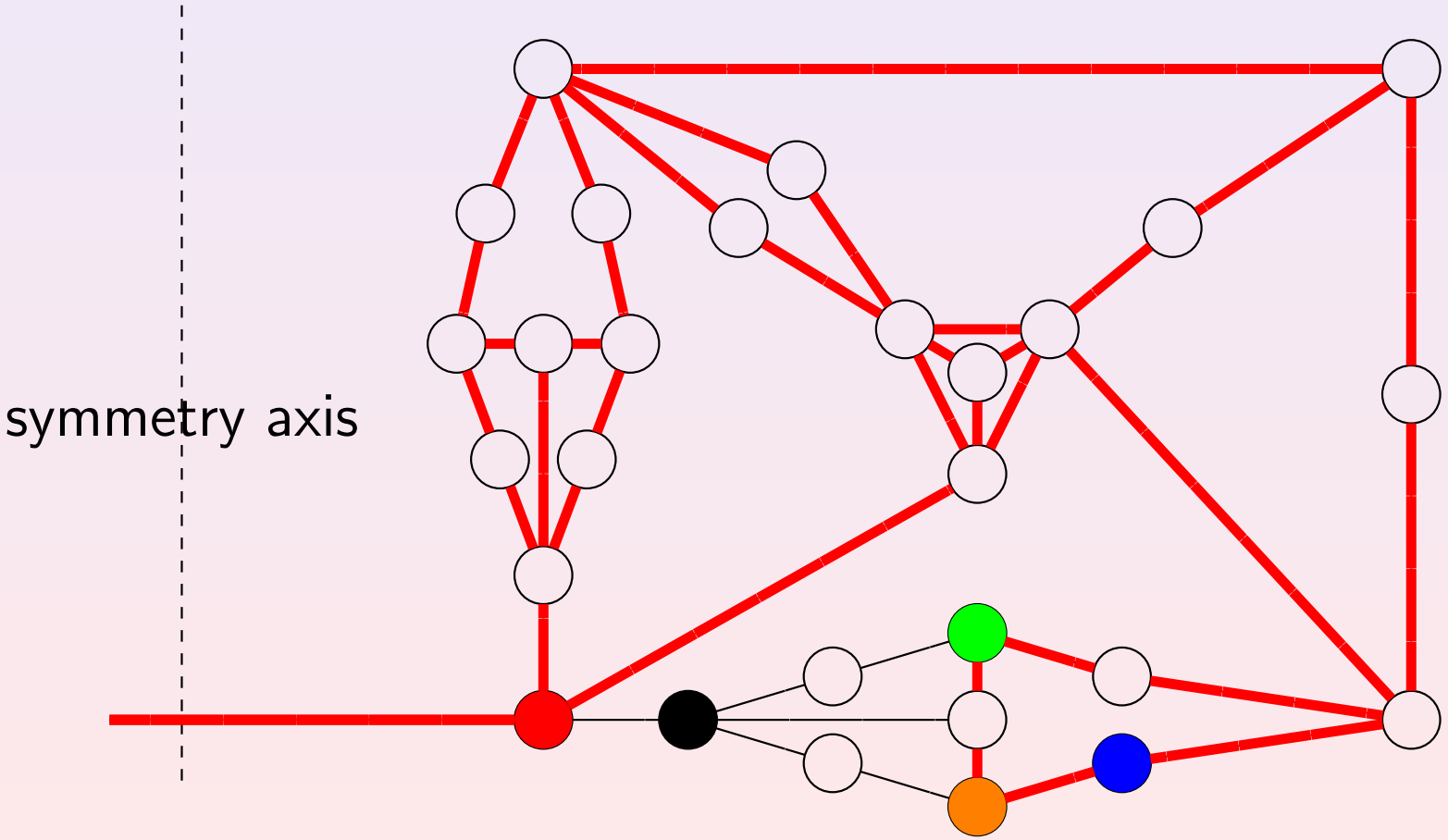
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

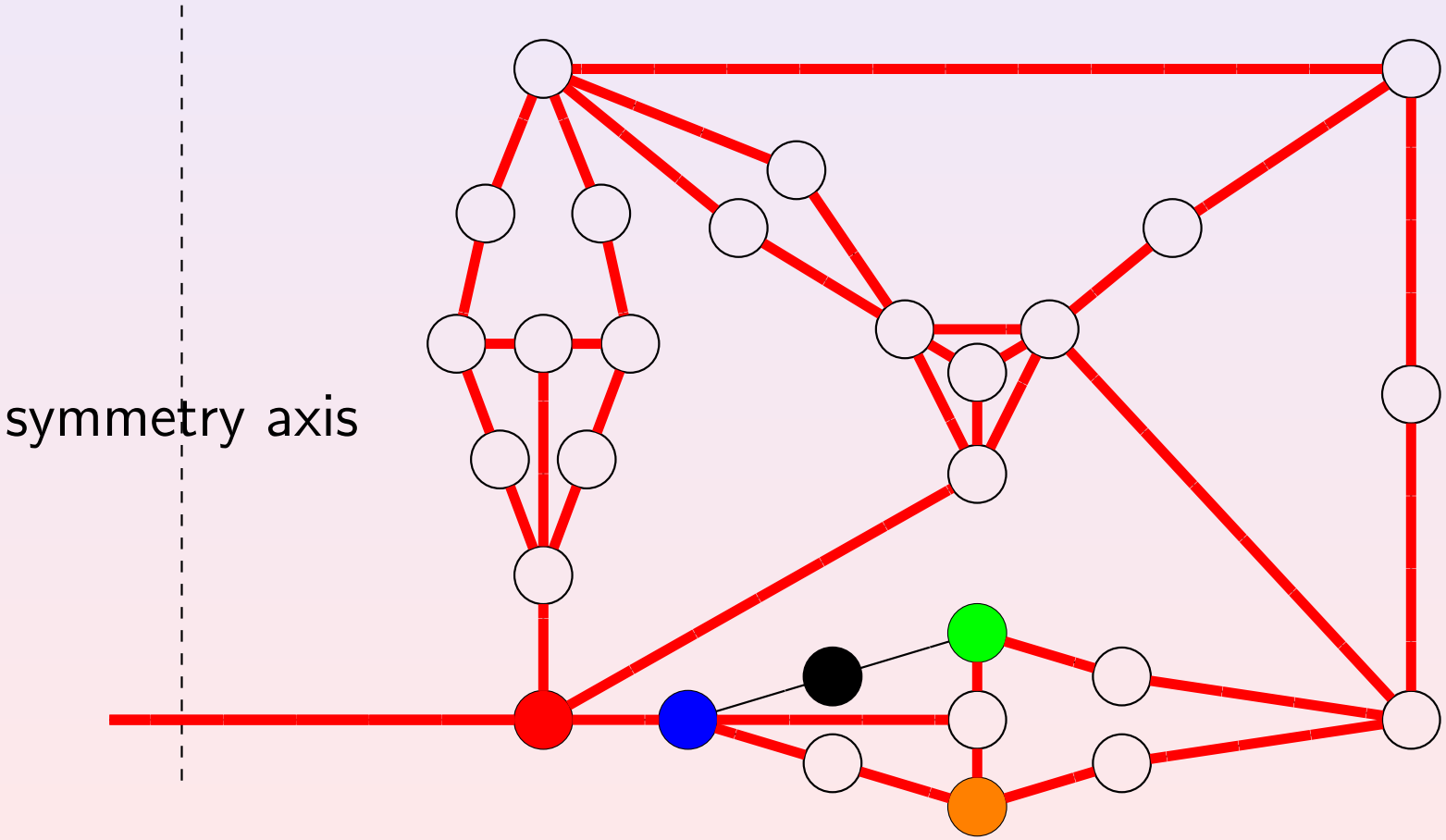
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

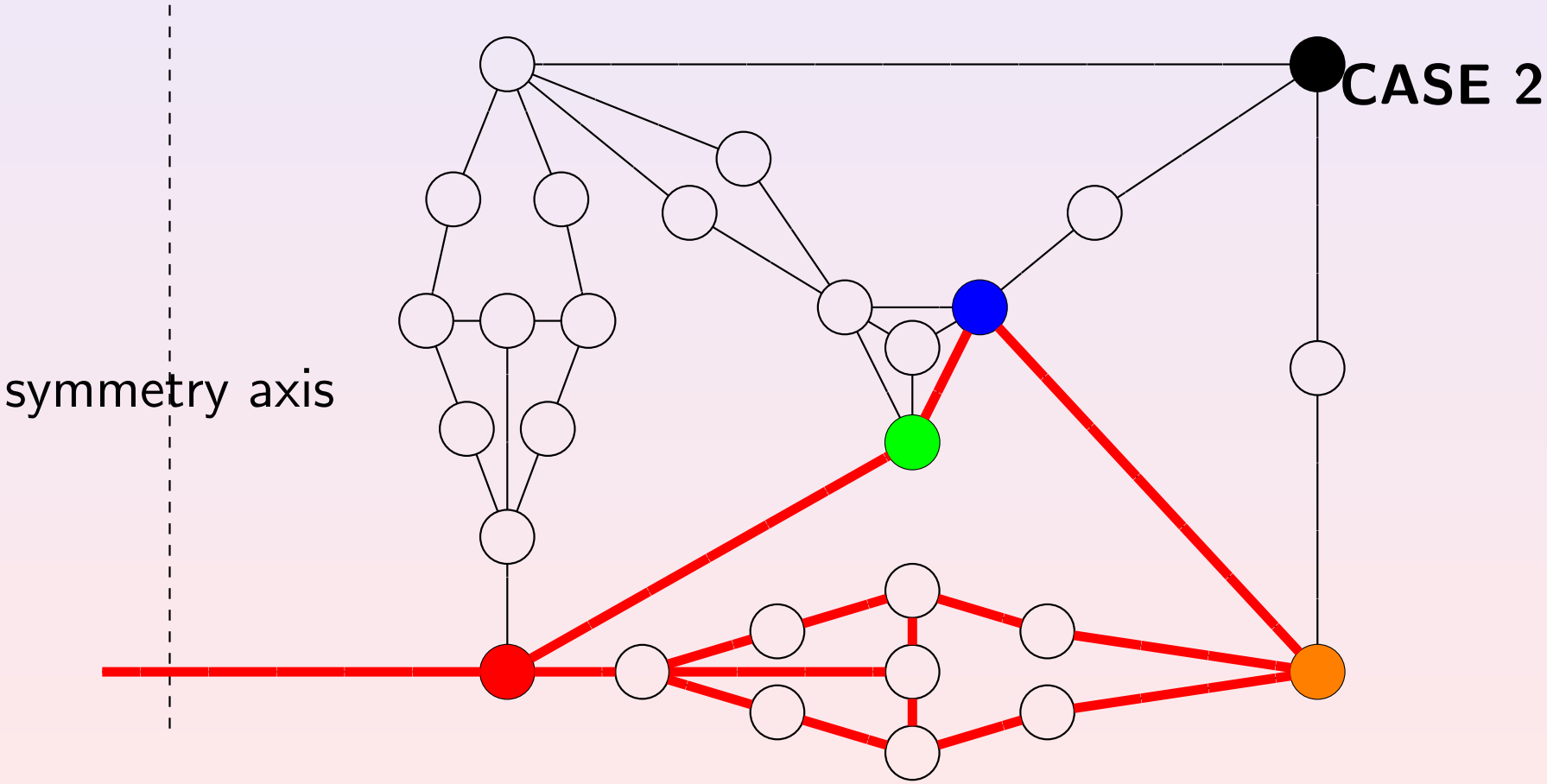
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

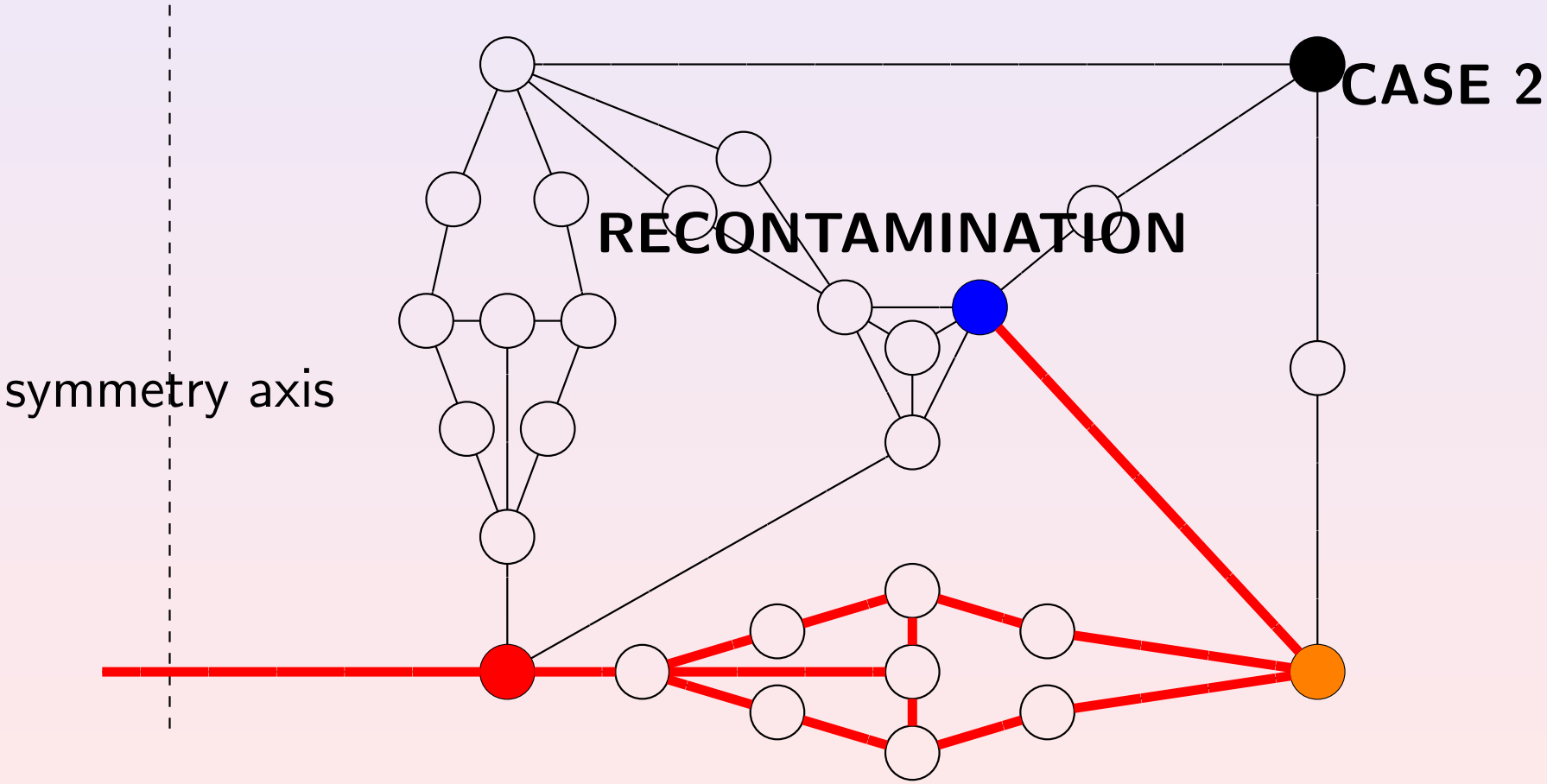
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

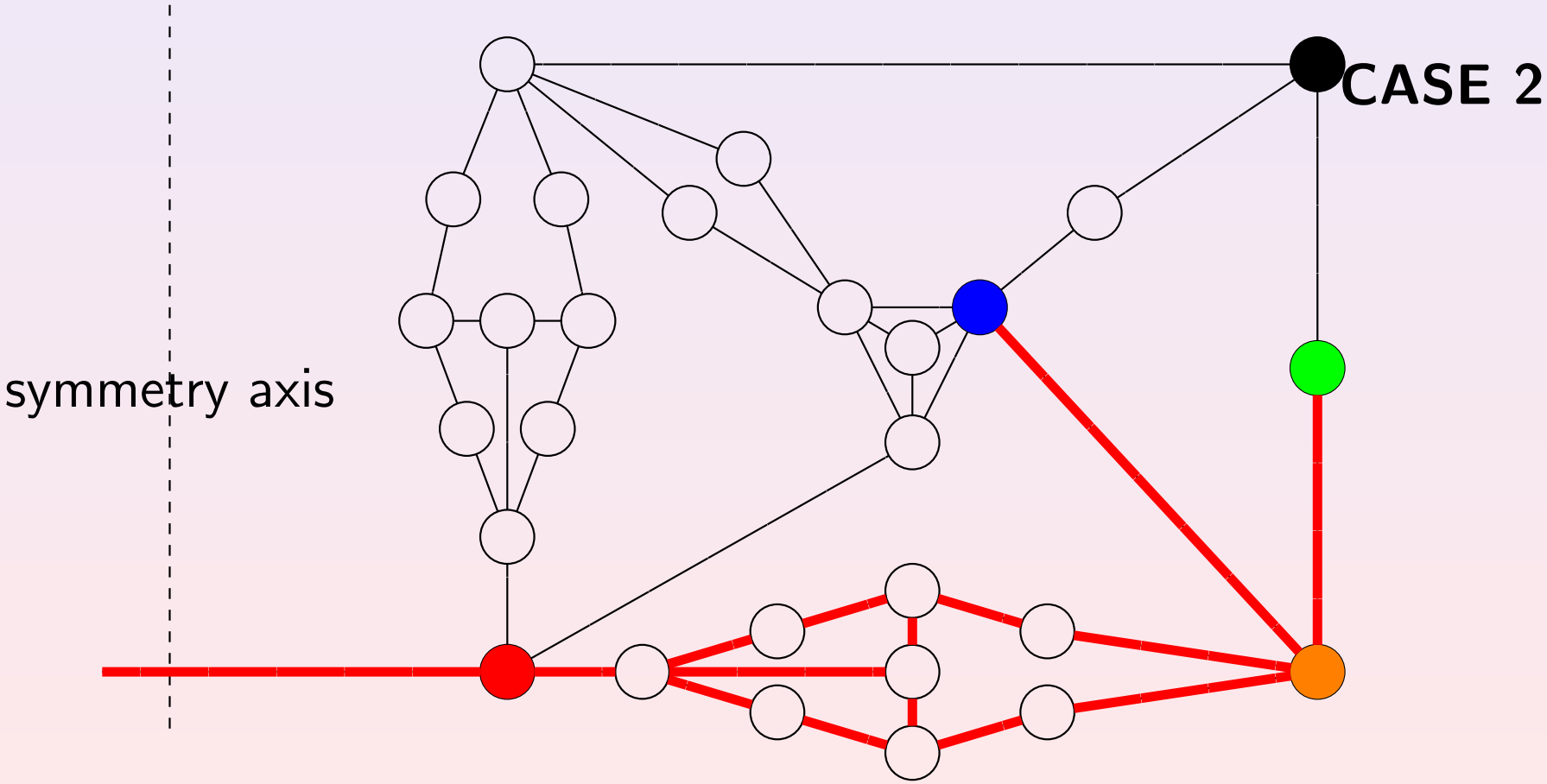
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

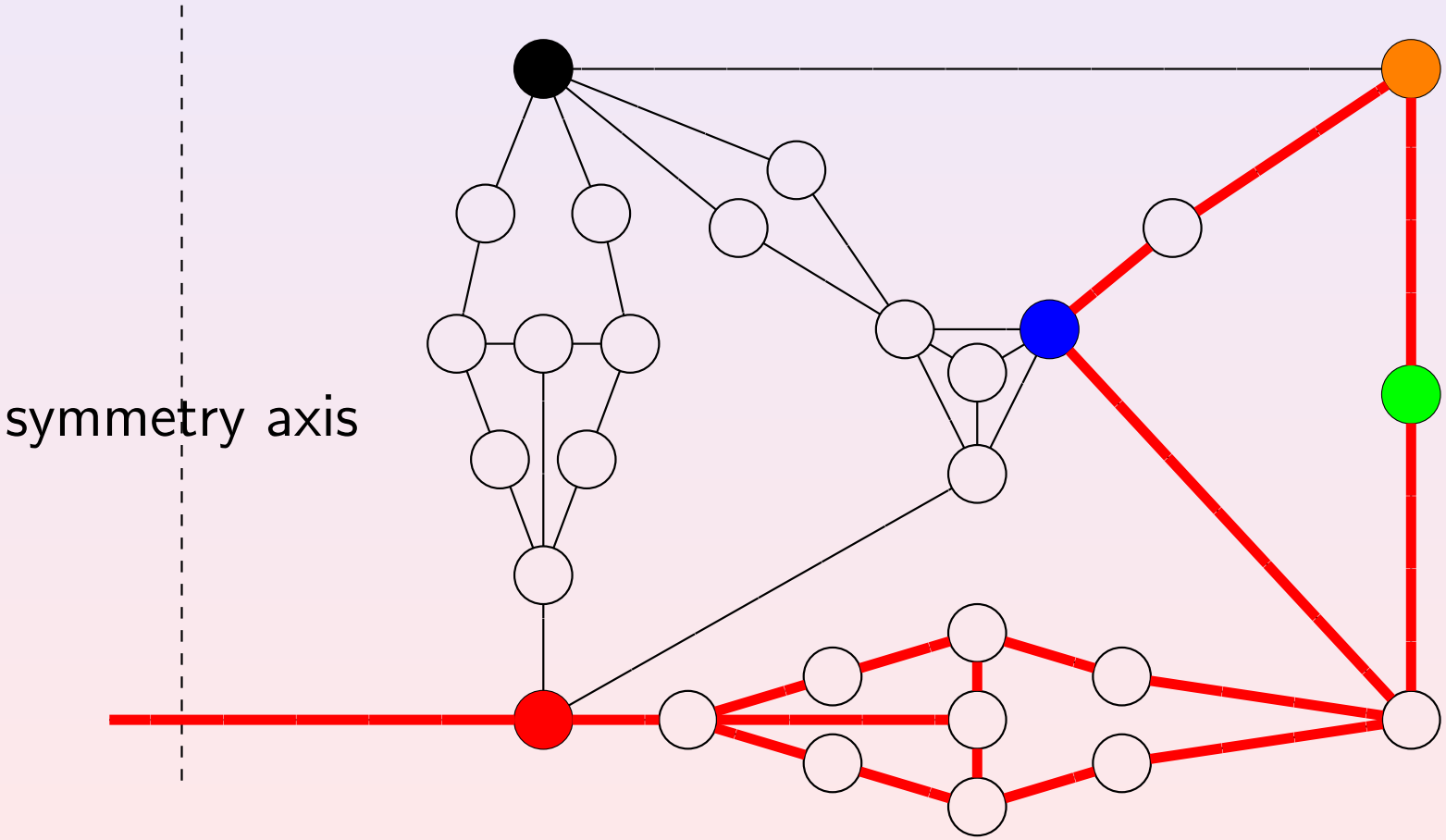
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

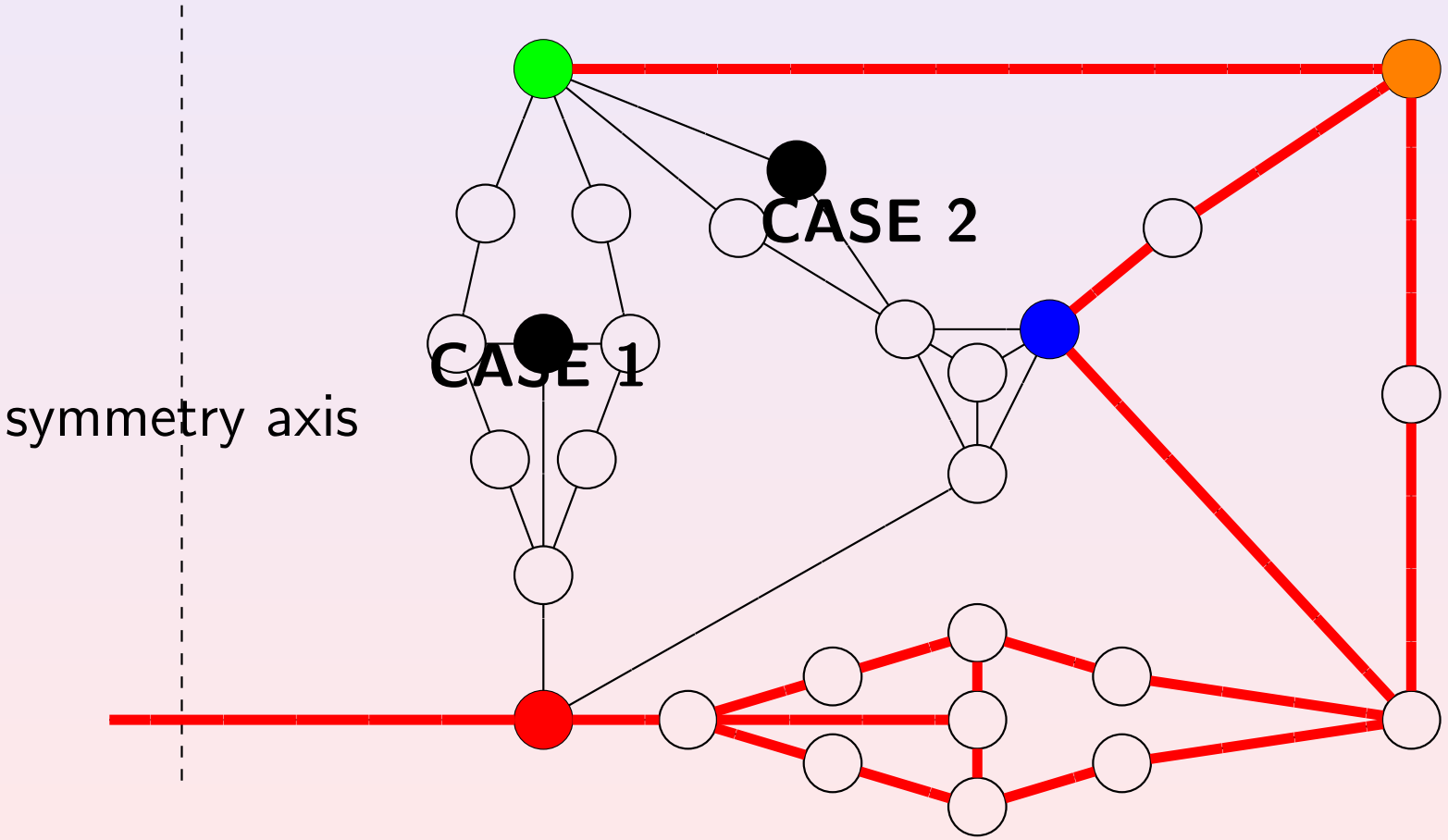
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

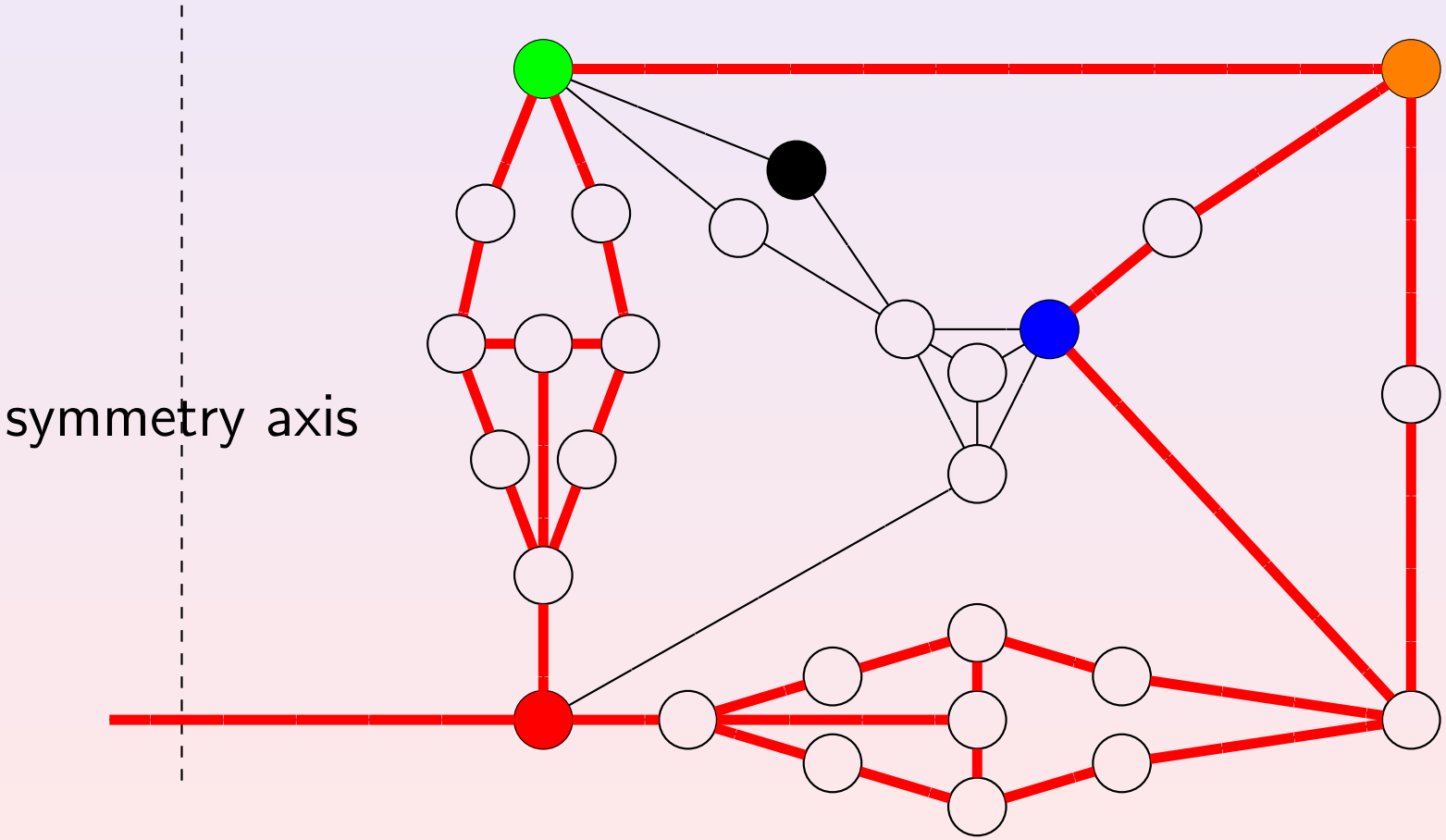
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

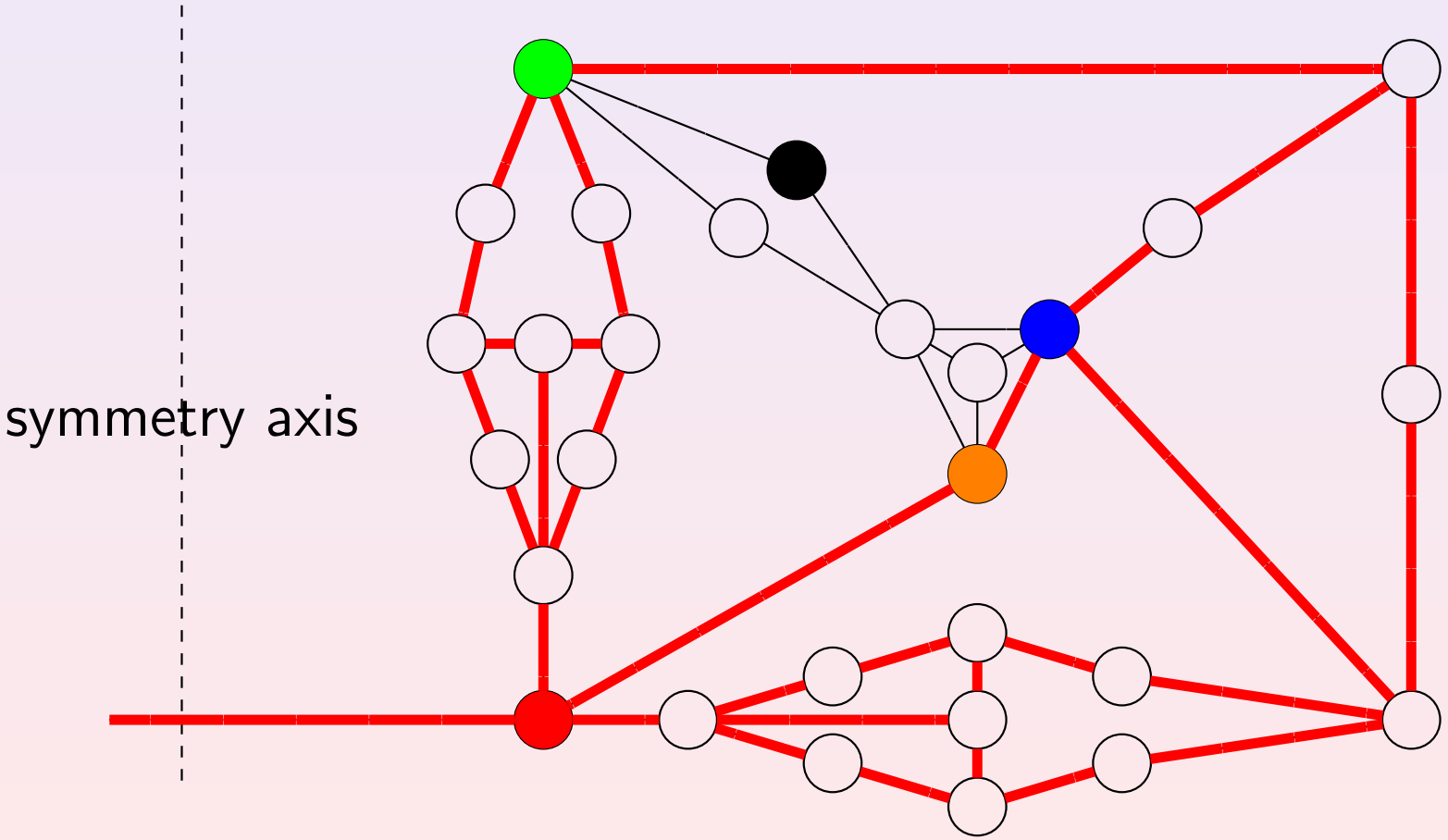
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

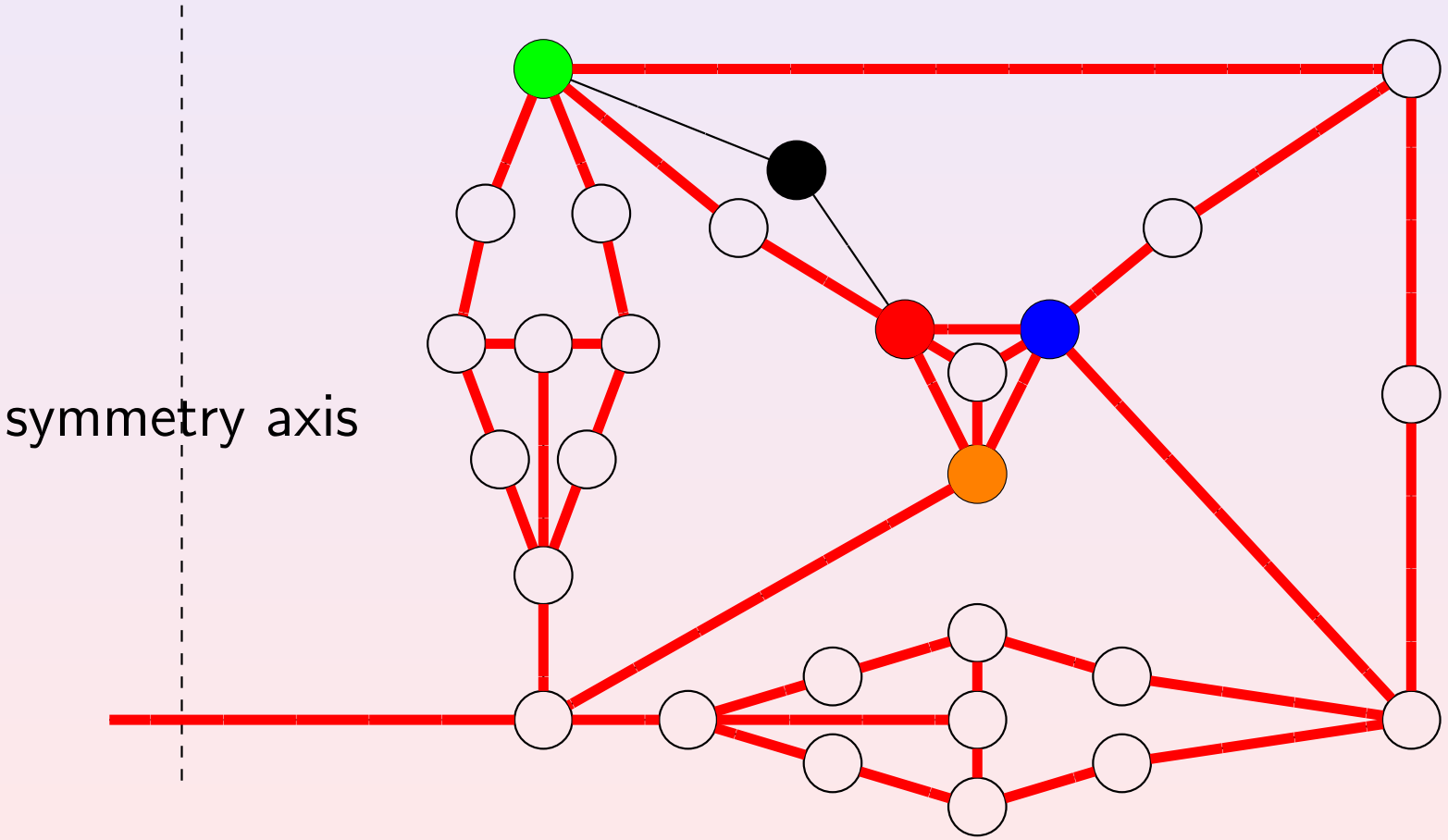
Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Monotony in Visible Connected Graph Searching

Recontamination helps

Let G be the graph below: $mcvs(G) > cvs(G) = 4$.



Conclusion and Further Work

Cost of connectivity

In visible Graph Searching

- $\mathbf{cvs}(G)/\mathbf{vs}(G) = \Theta(\log n)$
- recontamination helps

Open Problems

- $\mathbf{cs}(G)/\mathbf{s}(G) = O(\log n)$, what is the tight bound?
- What is the minimum k such that there is a graph G with $cs(G) = k$ and recontamination helps?
  The smallest known values is $k = 281$...
- Are there graphs G, such that $mcvs(G) > cvs(G) = 3$?
  I.e., does recontamination helps to catch a visible fugitive in a connected way using at most 3 searchers?