Temporalizing digraphs via linear-size balanced bi-trees

Laurent Viennot

joint work with :
Alkida Balliu, Filippo Brunelli, Pierluigi Crescenzi, Dennis Olivetti
and :
Stéphane Bessy, Stéphan Thomassé

Île du Saussay

Directed graph temporalisation

Define the temporal reachability of a temporal graph as the number of node pairs that are (strictly) temporally connected.

Problem : given a strongly connected multi-digraph, assign one time label per edge so that temporal reachability is maximal.

or within a constant factor from maximal (approximation).

Directed graph temporalisation

Define the temporal reachability of a temporal graph as the number of node pairs that are (strictly) temporally connected.

Problem : given a strongly connected multi-digraph, assign one time label per edge so that temporal reachability is maximal.

or within a constant factor from maximal (approximation).

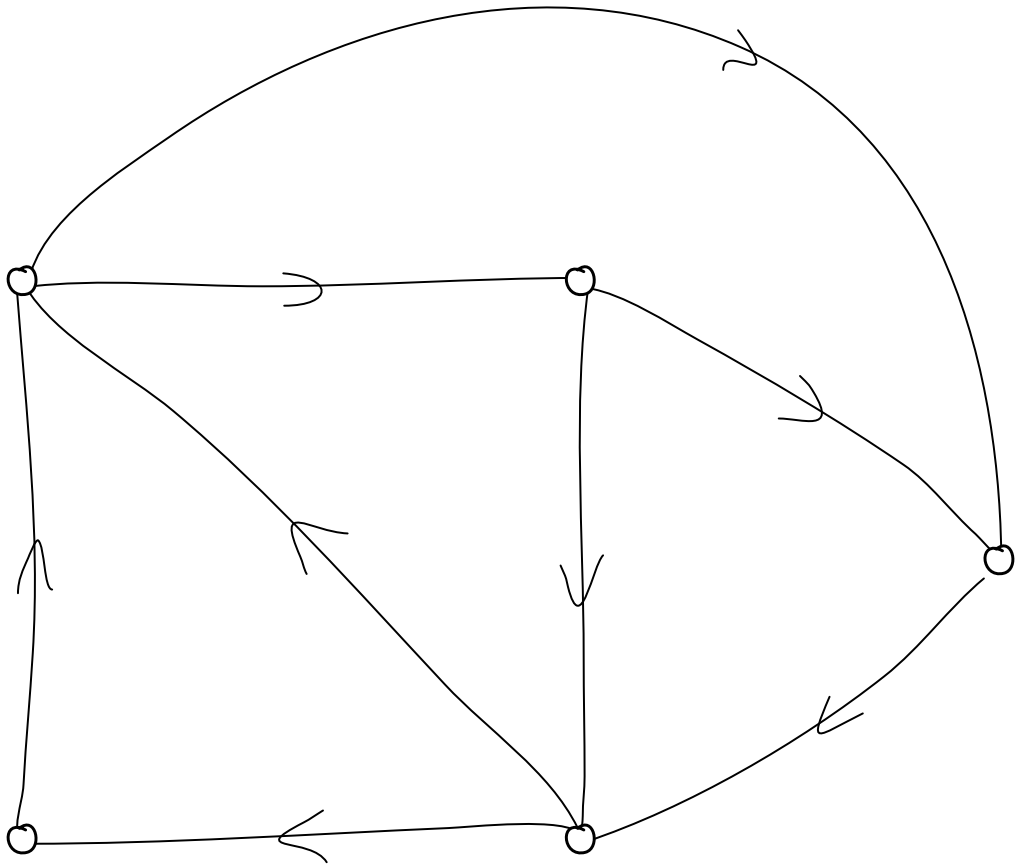
Directed graph temporalisation

Define the temporal reachability of a temporal graph as the number of node pairs that are (strictly) temporally connected.

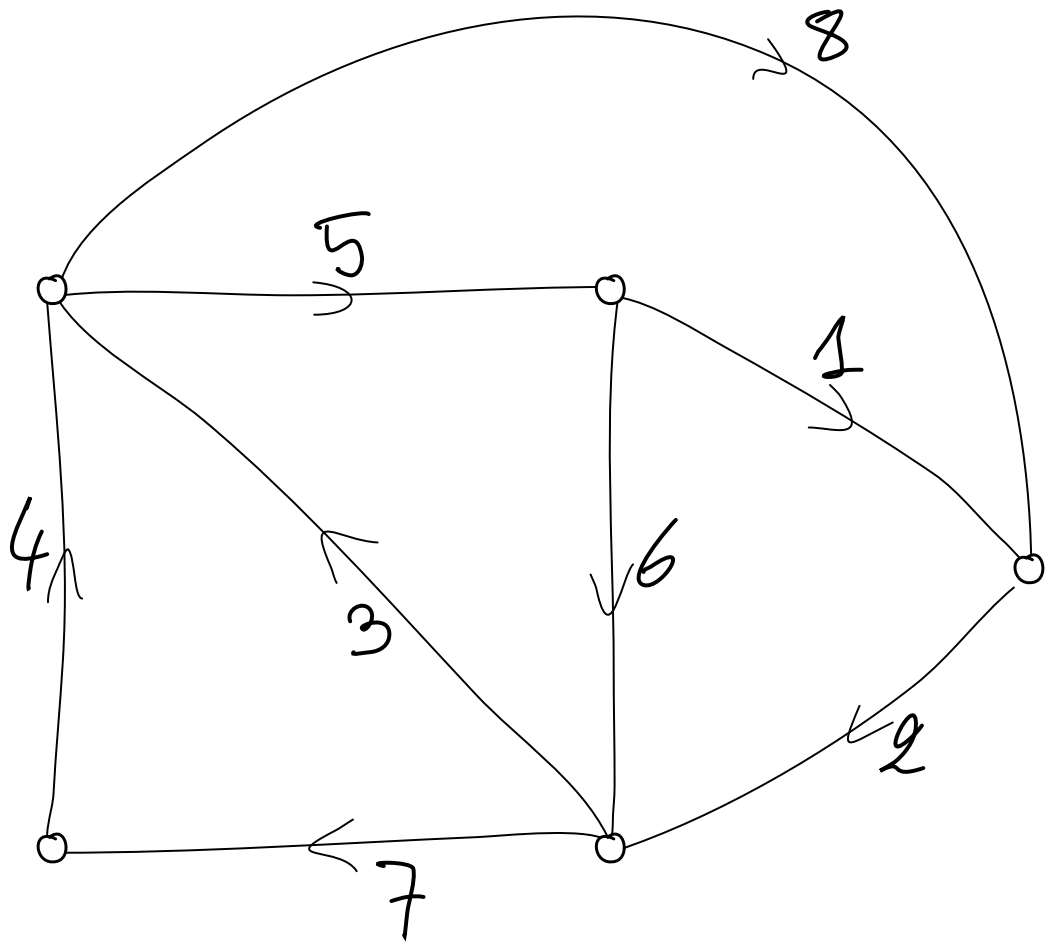
Problem : given a strongly connected multi-digraph, assign one time label per edge so that temporal reachability is maximal.

or within a constant factor from maximal (approximation).

Example



Example



Motivation

Original motivation : optimize the schedule of a public transit network.

In a social network : imagine a conference where attendees to talks are known, and where the order of talks is free.

The problem is mostly interesting when most pairs can be connected, we thus fucus on strong digraphs.

We will see that this problem is related to fundamental properties of strong digraphs.

Motivation

Original motivation : optimize the schedule of a public transit network.

In a social network : imagine a conference where attendees to talks are known, and where the order of talks is free.

The problem is mostly interesting when most pairs can be connected, we thus fucus on strong digraphs.

We will see that this problem is related to fundamental properties of strong digraphs.

Motivation

Original motivation : optimize the schedule of a public transit network.

In a social network : imagine a conference where attendees to talks are known, and where the order of talks is free.

The problem is mostly interesting when most pairs can be connected, we thus fucus on strong digraphs.

We will see that this problem is related to fundamental properties of strong digraphs.

Motivation

Original motivation : optimize the schedule of a public transit network.

In a social network : imagine a conference where attendees to talks are known, and where the order of talks is free.

The problem is mostly interesting when most pairs can be connected, we thus fucus on strong digraphs.

We will see that this problem is related to fundamental properties of strong digraphs.

Problem

Given a strong multi-digraph, compute a temporalization connecting a constant fraction of pairs through strict temporal paths.

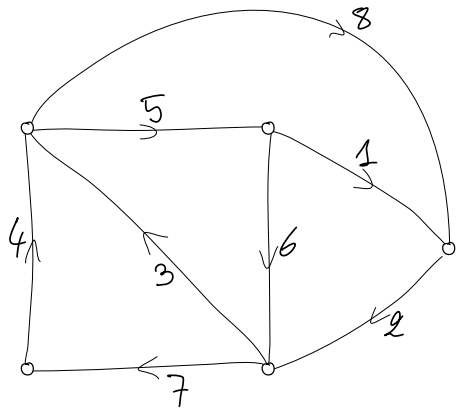
or equivalently an arc ordering σ for σ-respecting paths,

or a node ordering π for π-forward paths,

Problem

Given a strong multi-digraph, compute a temporalization connecting a constant fraction of pairs through strict temporal paths.

or equivalently an arc ordering σ for σ-respecting paths,

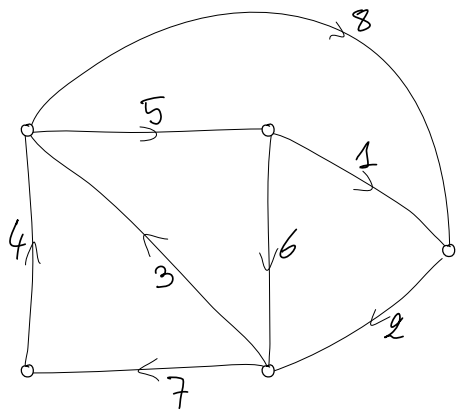


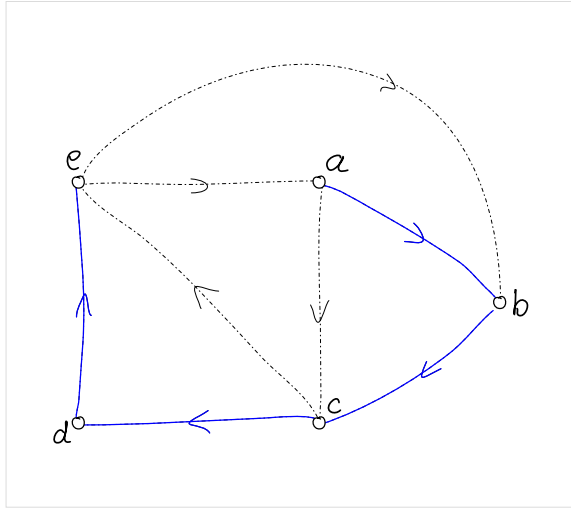
or a node ordering π for π-forward paths,

Problem

Given a strong multi-digraph, compute a temporalization connecting a constant fraction of pairs through strict temporal paths.

or equivalently an arc ordering σ for σ-respecting paths,



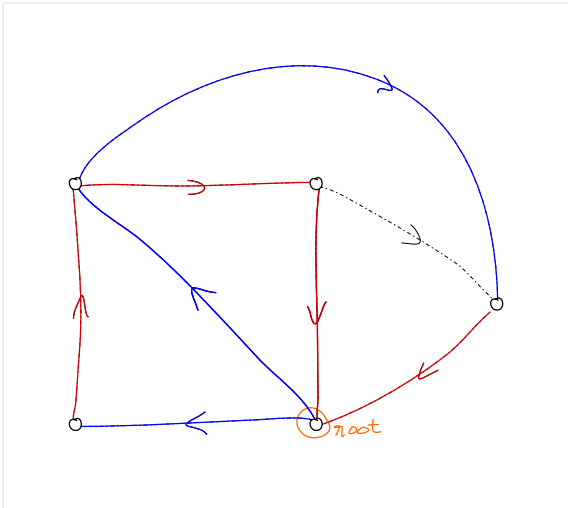


or a node ordering π for π-forward paths,

Problem

Given a strong multi-digraph, compute a temporalization connecting a constant fraction of pairs through strict temporal paths.

or a pair of edge-disjoint in-tree and out-tree both spanning a constant fraction of nodes,

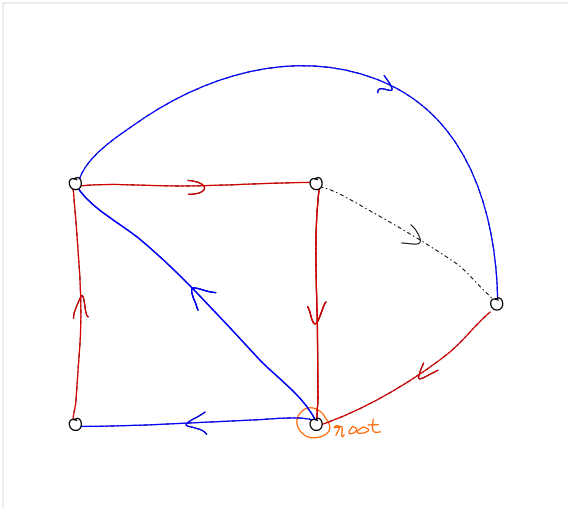


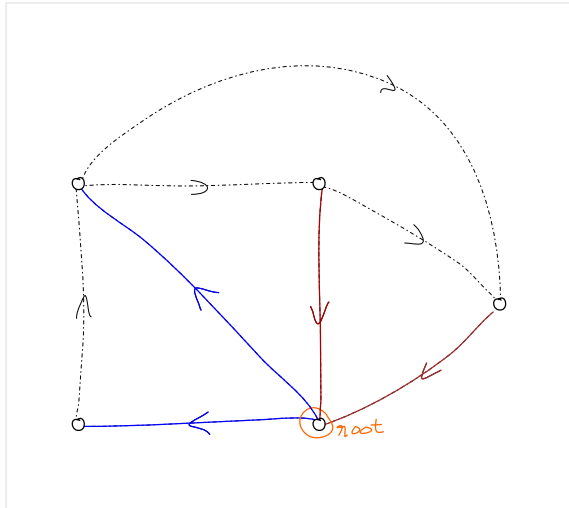
or a bitree, i.e. a pair of node-disjoint in-tree and out-tree both spanning a constant fraction of nodes.

Problem

Given a strong multi-digraph, compute a temporalization connecting a constant fraction of pairs through strict temporal paths.

or a pair of edge-disjoint in-tree and out-tree both spanning a constant fraction of nodes,





or a bitree, i.e. a pair of node-disjoint in-tree and out-tree both spanning a constant fraction of nodes.

Related work

Undirected graph : deciding "label connectivity" is NP-complete [Göbel, Cerdeira, Veldman 1991], approximation is easy.

Minimizing $|\lambda|$, i.e. the number of labels, for achieving temporal connectivity is NP-hard [Klobas, Mertzios, Molter, Spirakis 2022].

It is NP-hard to decide if a strong digraph has a pair of edge-disjoint spanning in-tree and out-tree [Bang-Jensen 1991].

Related work

Undirected graph : deciding "label connectivity" is NP-complete [Göbel, Cerdeira, Veldman 1991], approximation is easy.

Minimizing $|\lambda|$, i.e. the number of labels, for achieving temporal connectivity is NP-hard [Klobas, Mertzios, Molter, Spirakis 2022].

It is NP-hard to decide if a strong digraph has a pair of edge-disjoint spanning in-tree and out-tree [Bang-Jensen 1991].

Related work

Undirected graph : deciding "label connectivity" is NP-complete [Göbel, Cerdeira, Veldman 1991], approximation is easy.

Minimizing $|\lambda|$, i.e. the number of labels, for achieving temporal connectivity is NP-hard [Klobas, Mertzios, Molter, Spirakis 2022].

It is NP-hard to decide if a strong digraph has a pair of edge-disjoint spanning in-tree and out-tree [Bang-Jensen 1991].

Result 1 : hardness

Given a strong digraph D, deciding whether there exists an assignment of one time label per edge such that all pairs are temporally connected is NP-complete. [Balliu, Brunelli, Crescenzi, Olivetti, V. 2023]

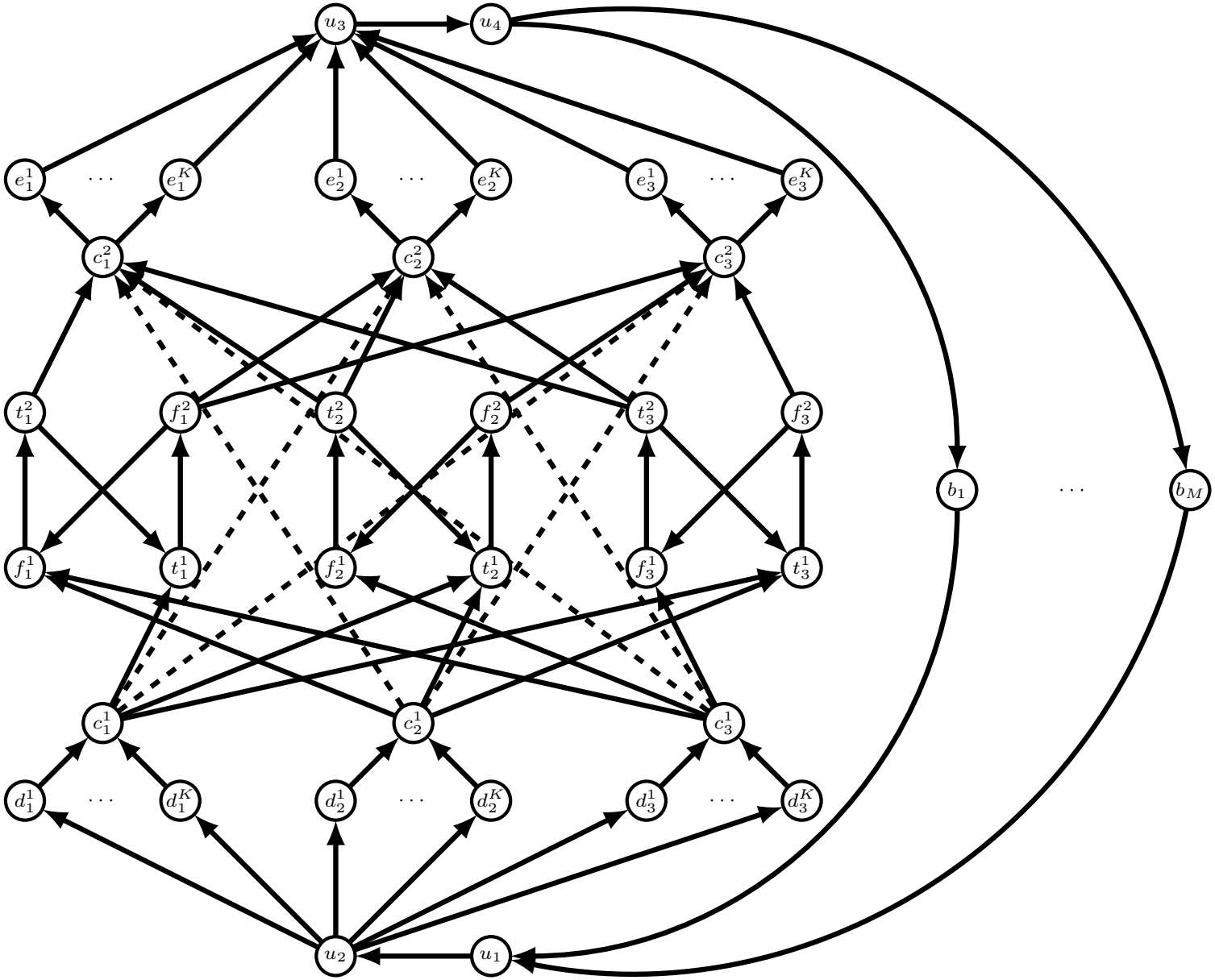
Conjecture : any strong digraph has a pair of edge-disjtoint in-tree and out-tree both spanning $n/3$ nodes.

Result 1 : hardness

Given a strong digraph D, deciding whether there exists an assignment of one time label per edge such that all pairs are temporally connected is NP-complete. [Balliu, Brunelli, Crescenzi, Olivetti, V. 2023]

Conjecture : any strong digraph has a pair of edge-disjtoint in-tree and out-tree both spanning $n/3$ nodes.

Proof of hardness : reduction from 3SAT



$$(\mathbf{x}_1 \vee \mathbf{x}_2 \vee \mathbf{x}_3) \wedge (\neg\mathbf{x}_1 \vee \mathbf{x}_2 \vee \mathbf{x}_3) \wedge (\neg\mathbf{x}_1 \vee \neg\mathbf{x}_2 \vee \neg\mathbf{x}_3)$$

Result 2 : approximation

Any strong digraph D has a pair of node-disjtoint in-tree and out-tree both spanning $n/6$ nodes that can be computed in $O(n^2)$ time [Bessy, Thomassé, V. 2023].

Lemma : any strong digraph (V, A) has a balanced cyclic separator C, that is V can be partionned in I, C, O such that :

- C is spanned by a directed cycle,
- there are no arcs from I to O (directed separator),
- both $I \cup C$ and $I \cup O$ has size at least $n/3$ (balanced).

Main tool : a "left-maximal" DFS tree.

Result 2 : approximation

Any strong digraph D has a pair of node-disjtoint in-tree and out-tree both spanning $n/6$ nodes that can be computed in $O(n^2)$ time [Bessy, Thomassé, V. 2023].

Lemma : any strong digraph (V, A) has a balanced cyclic separator C, that is V can be partionned in I, C, O such that :

- C is spanned by a directed cycle,
- there are no arcs from I to O (directed separator),
- both $I \cup C$ and $I \cup O$ has size at least $n/3$ (balanced).

Main tool : a "left-maximal" DFS tree.

Result 2 : approximation

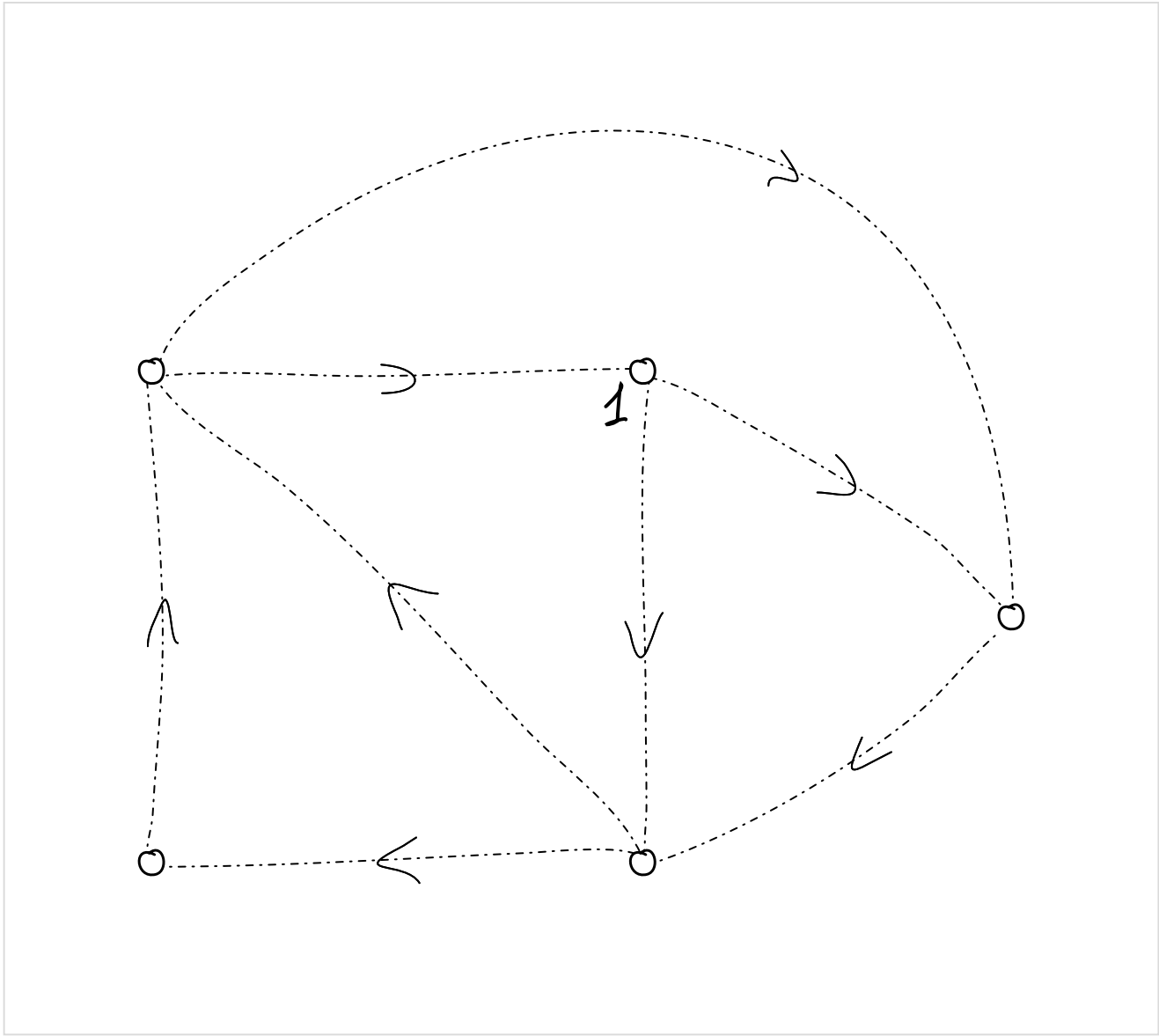
Any strong digraph D has a pair of node-disjtoint in-tree and out-tree both spanning $n/6$ nodes that can be computed in $O(n^2)$ time [Bessy, Thomassé, V. 2023].

Lemma : any strong digraph (V, A) has a balanced cyclic separator C, that is V can be partionned in I, C, O such that :

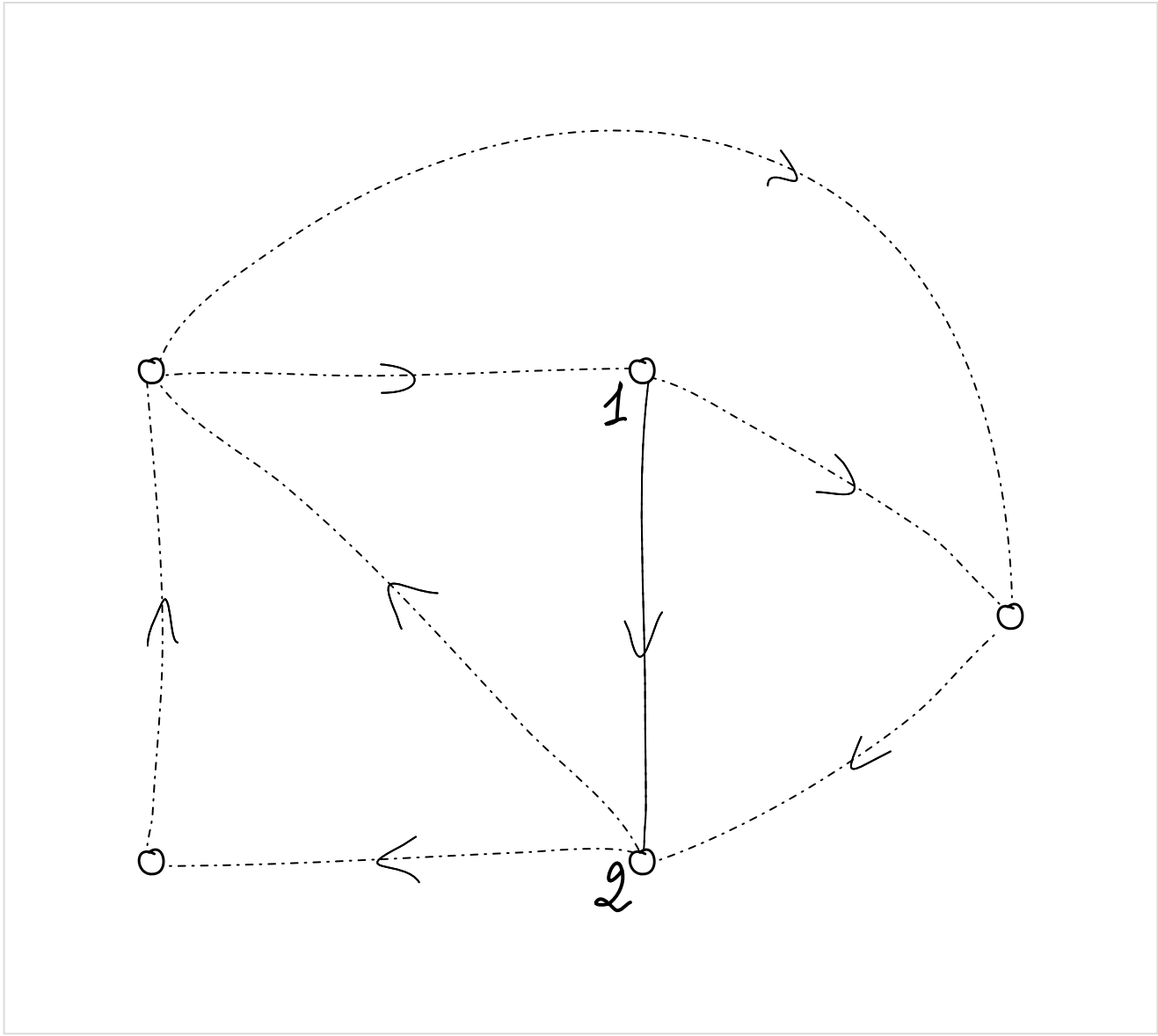
- C is spanned by a directed cycle,
- there are no arcs from I to O (directed separator),
- both $I \cup C$ and $I \cup O$ has size at least $n/3$ (balanced).

Main tool : a "left-maximal" DFS tree.

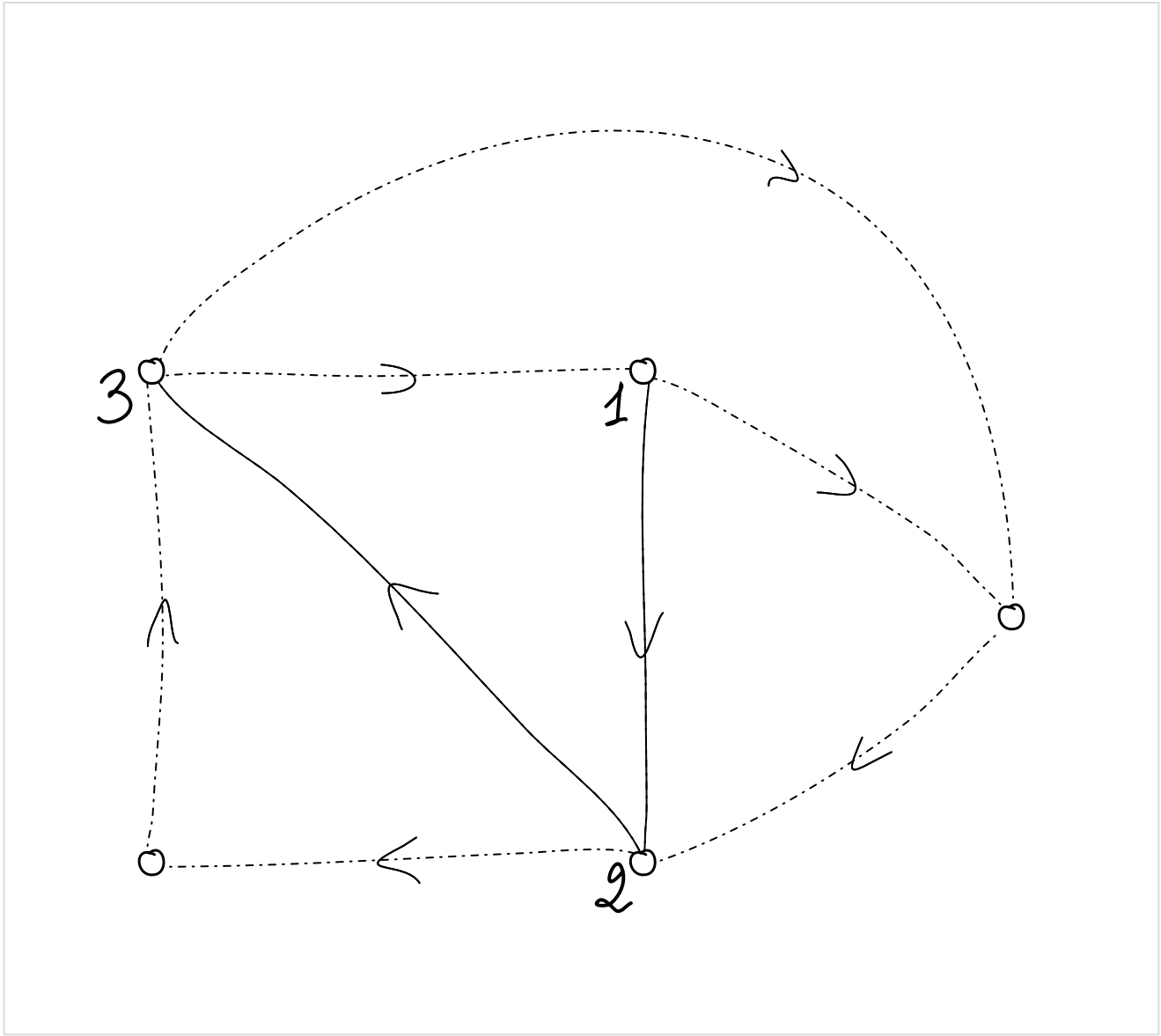
DFS to bitree



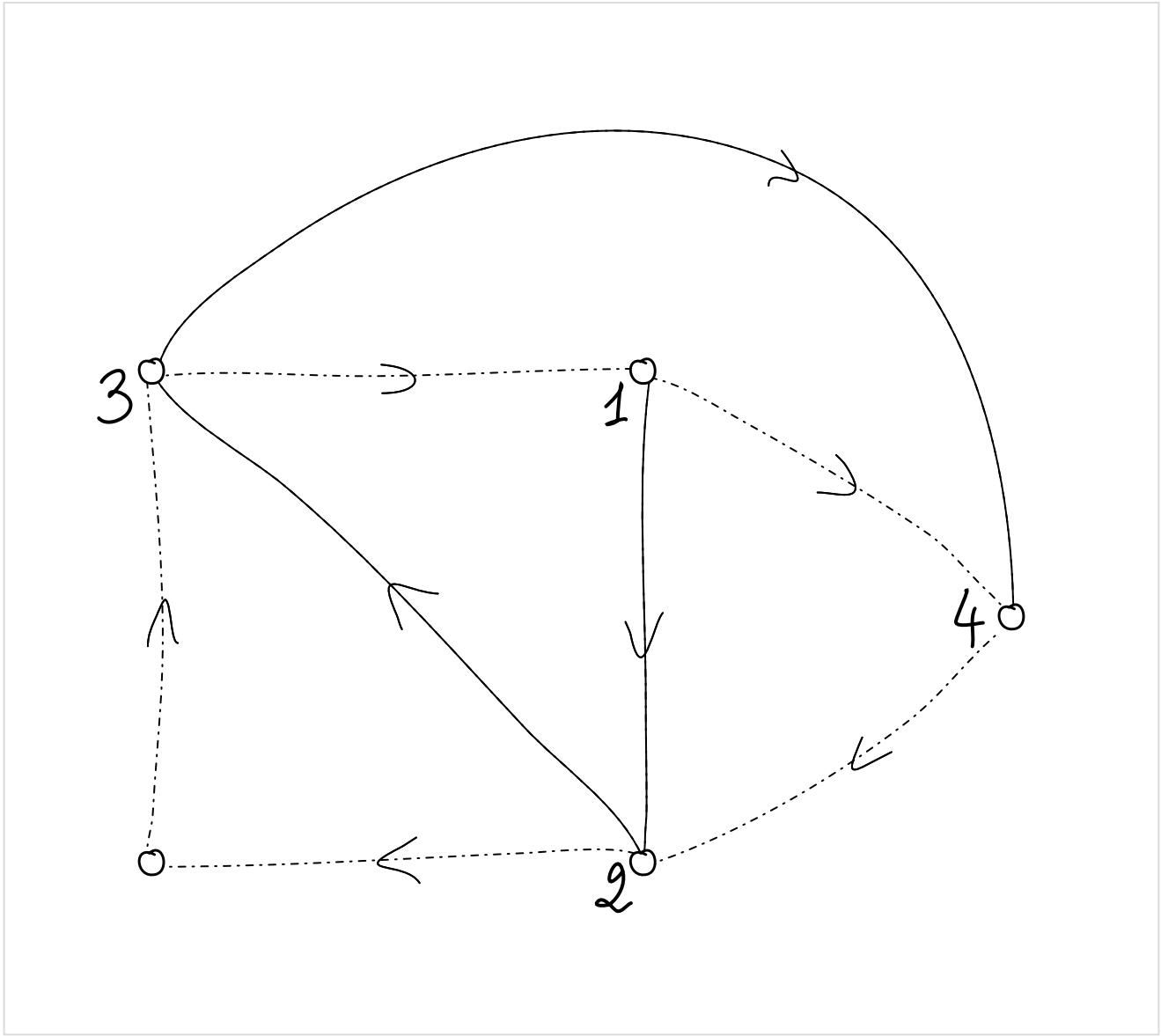
DFS to bitree



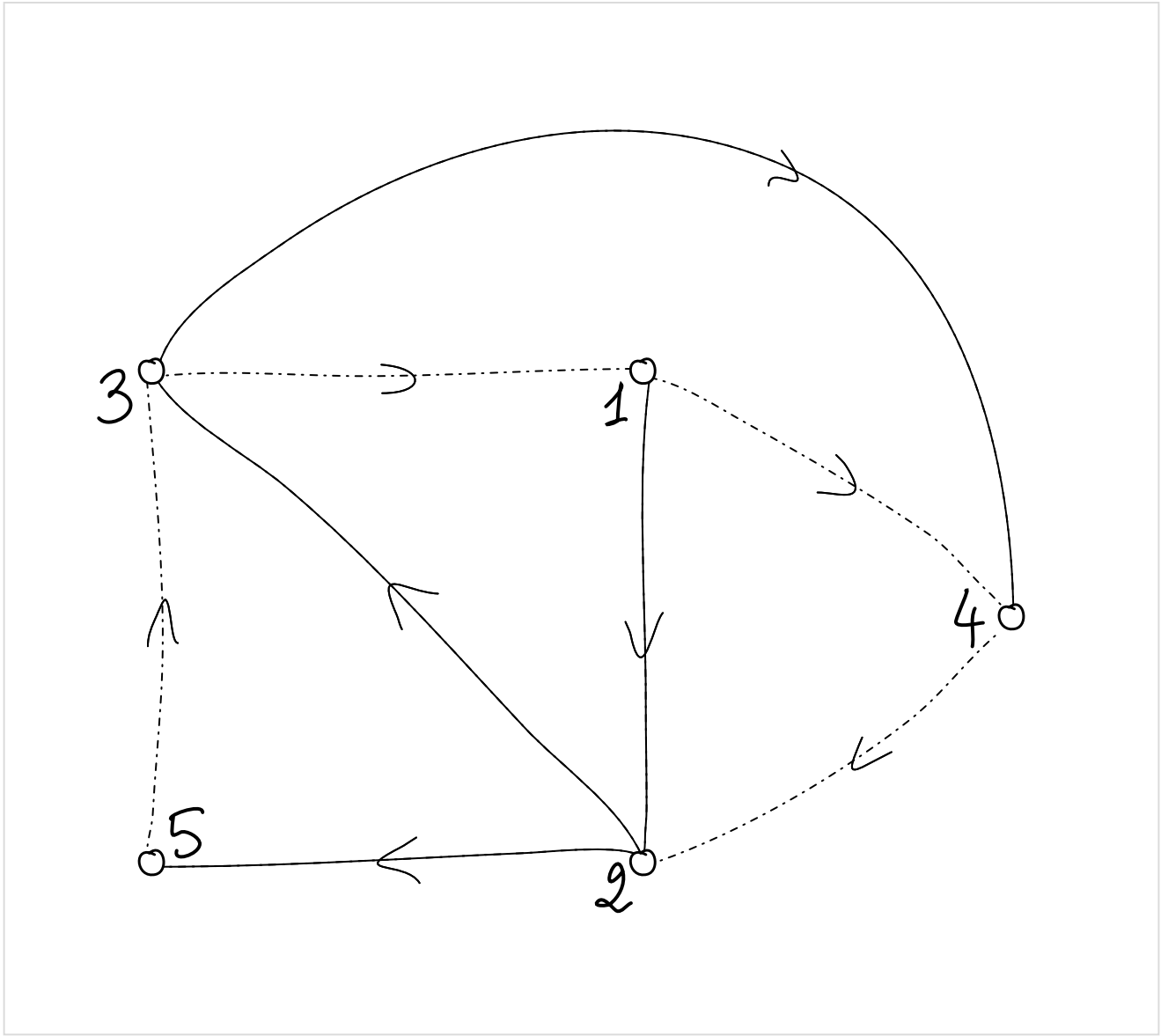
DFS to bitree



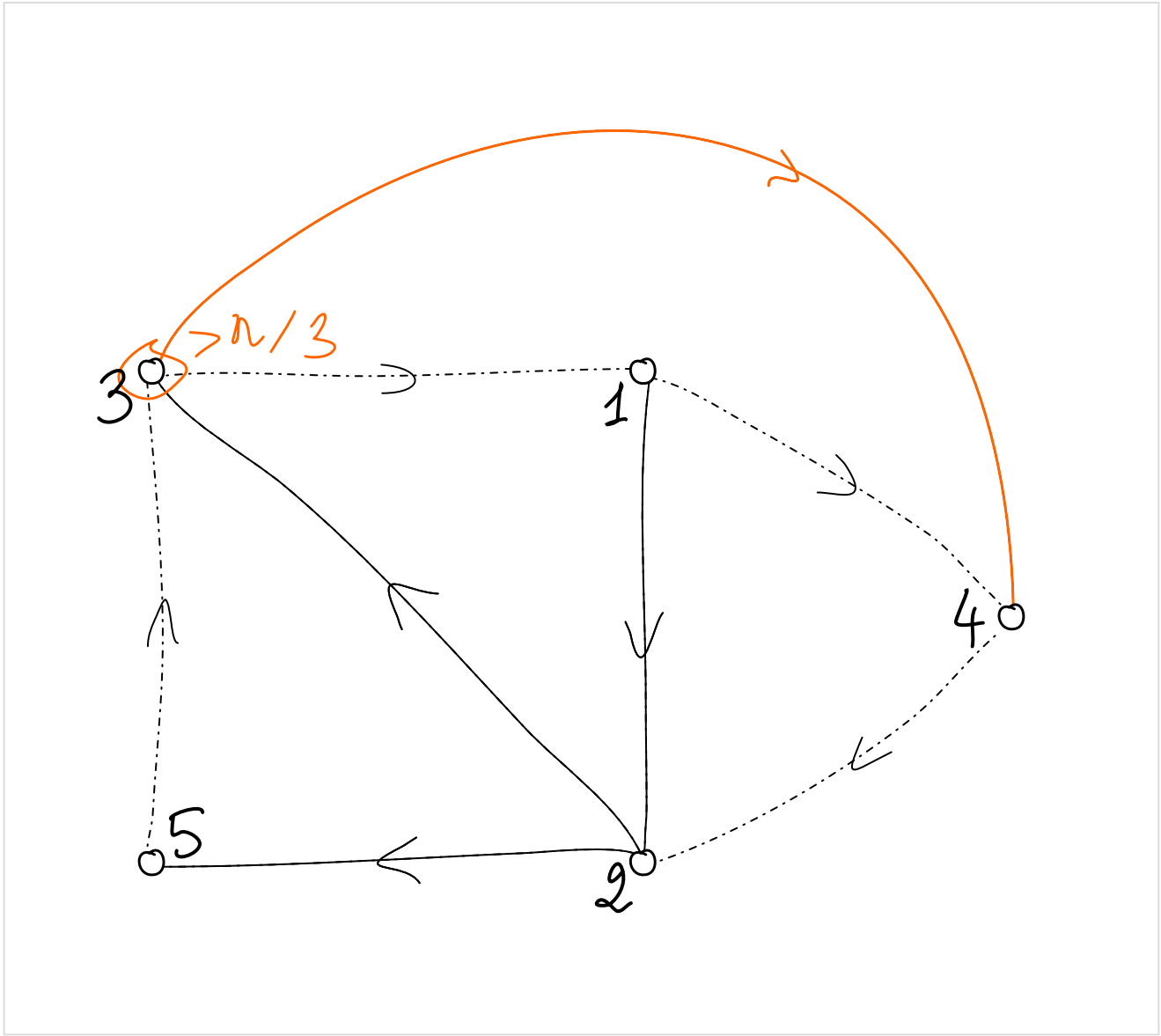
DFS to bitree



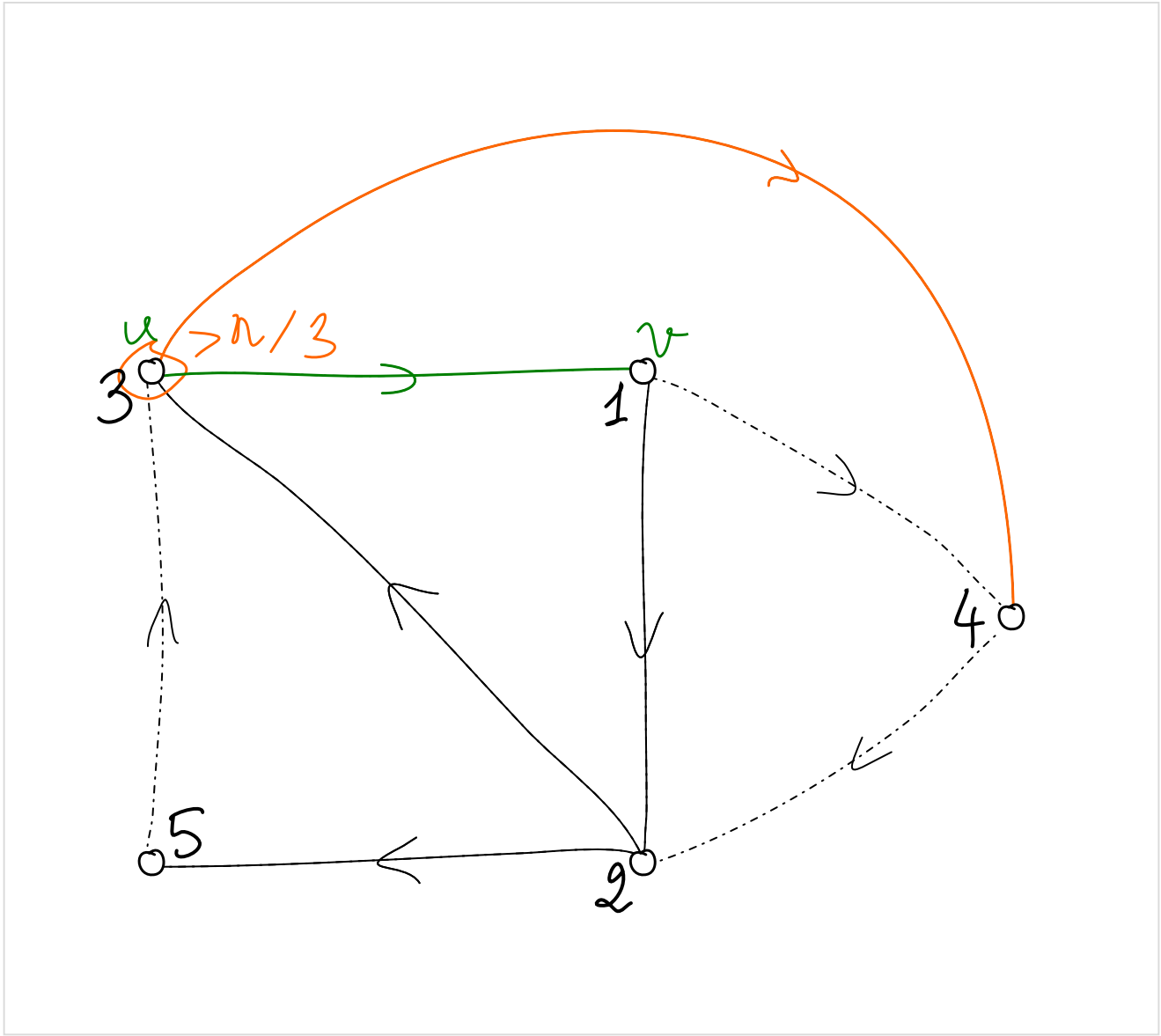
DFS to bitree



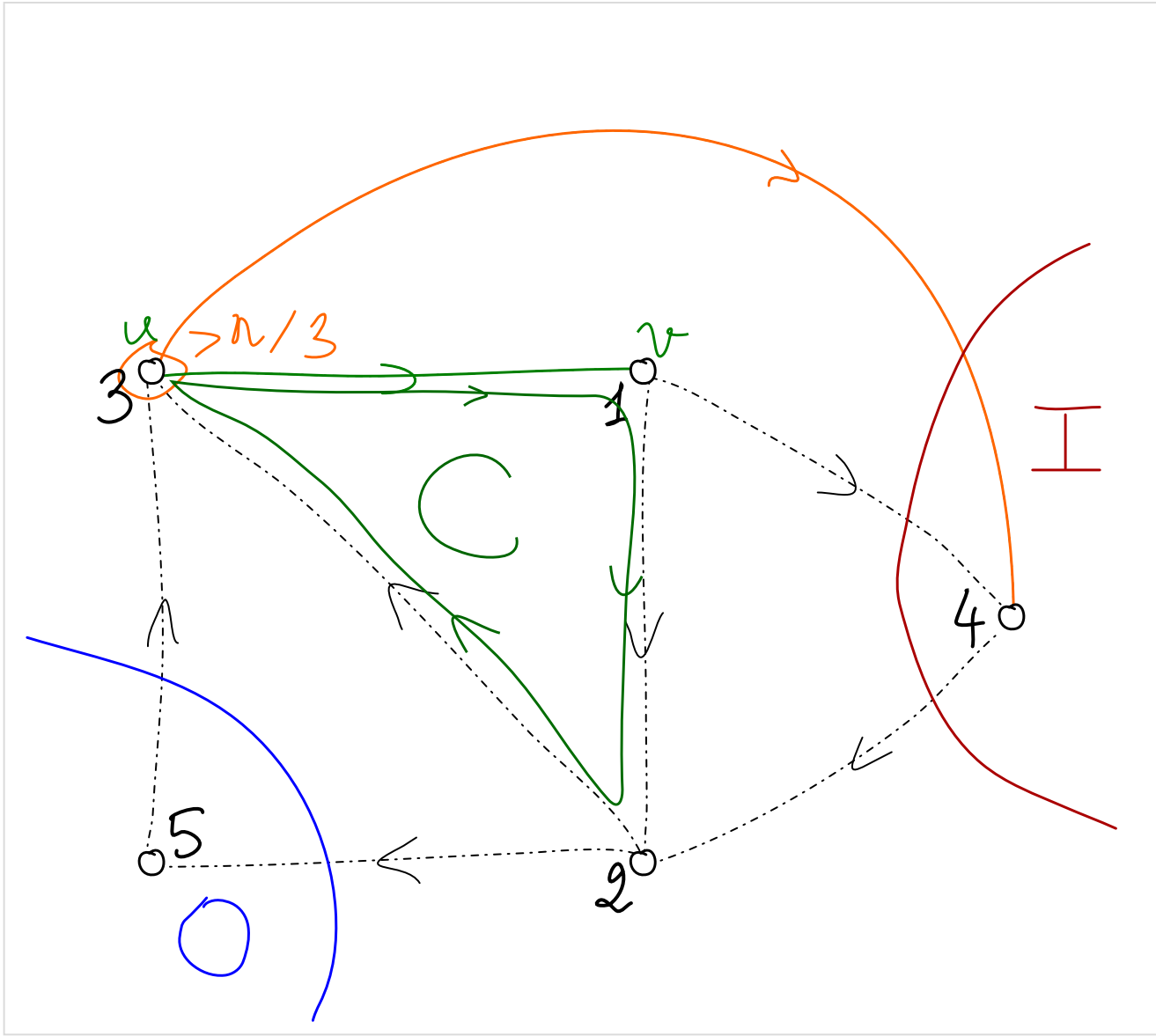
DFS to bitree



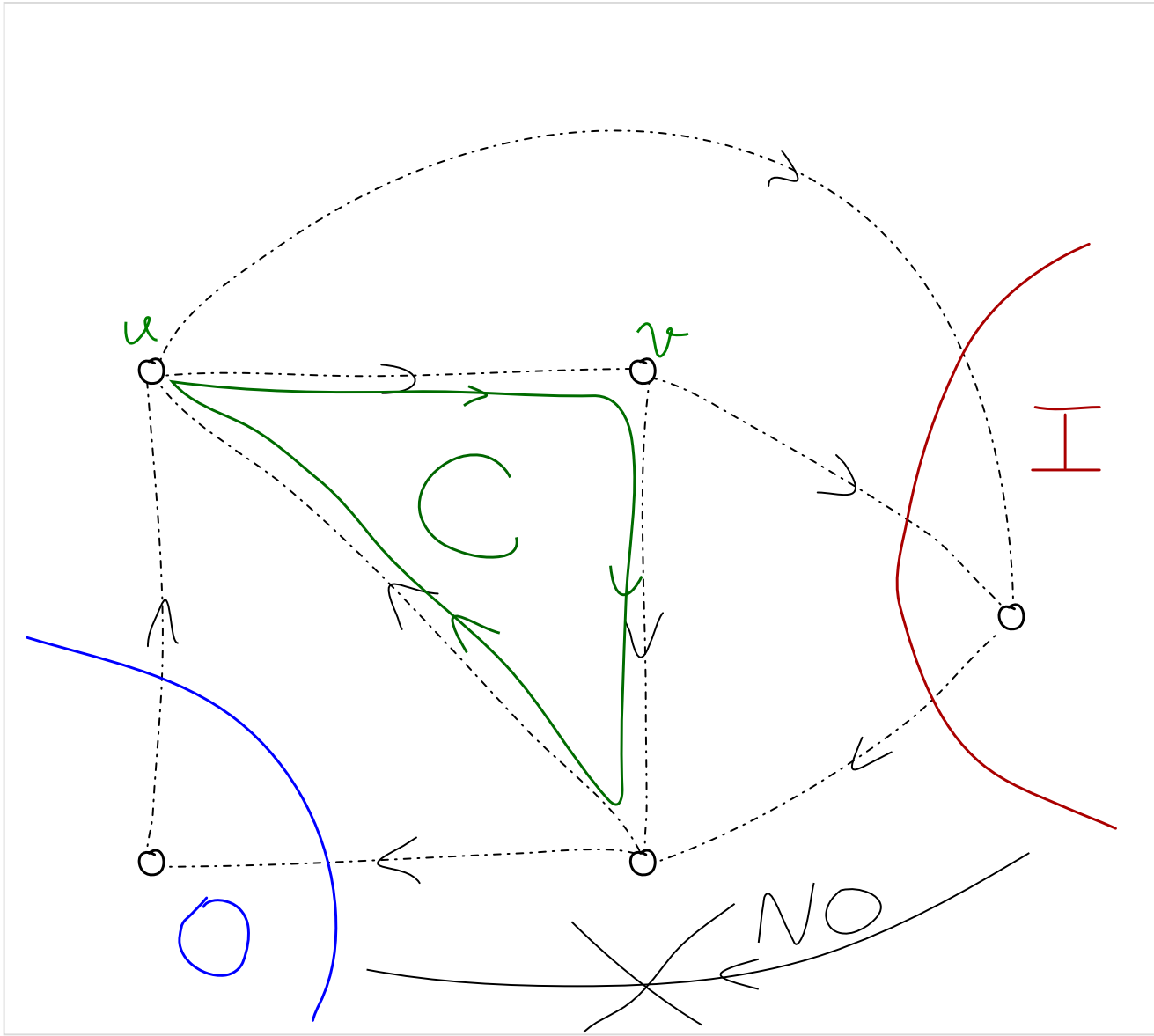
DFS to bitree



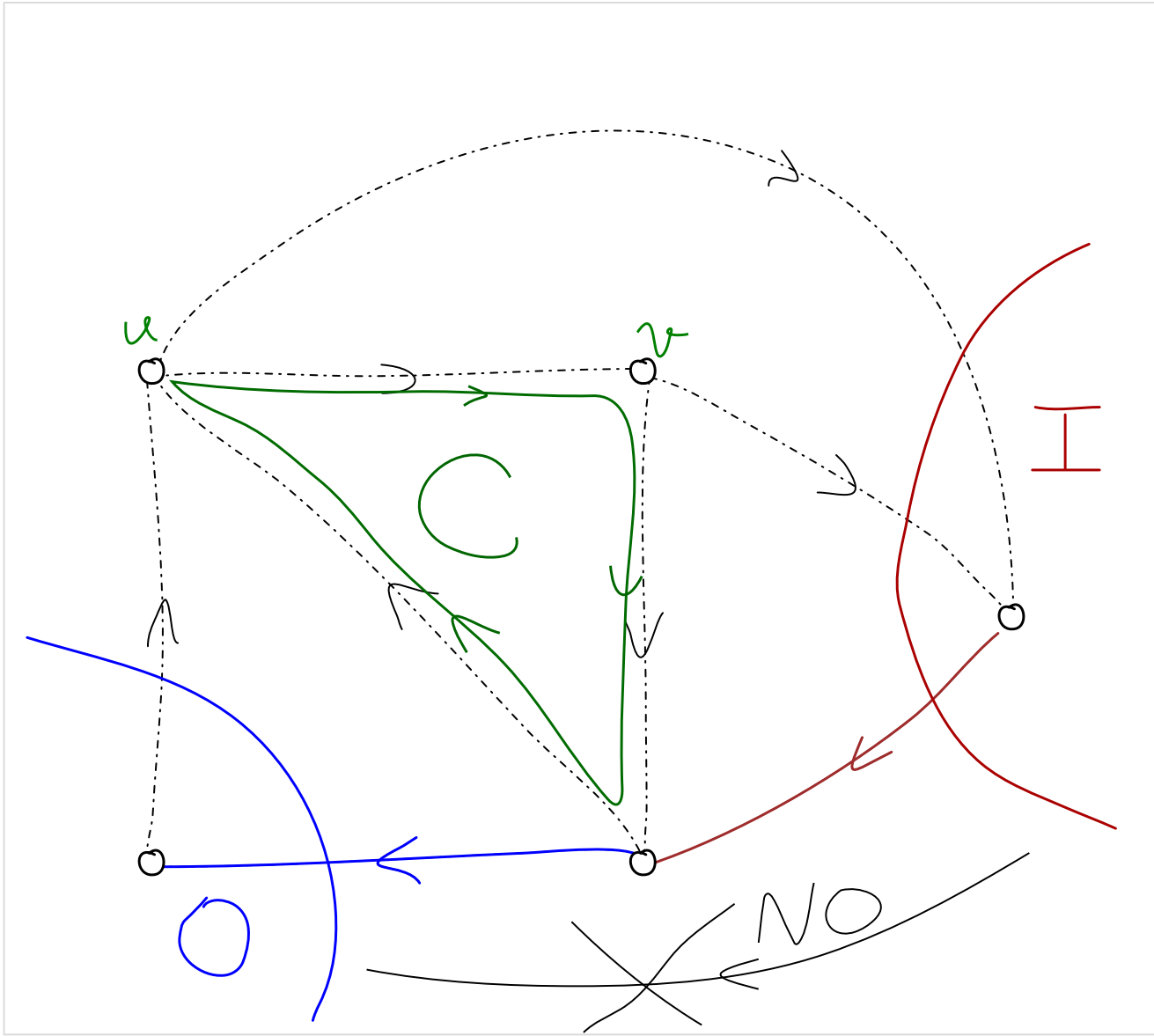
DFS to bitree



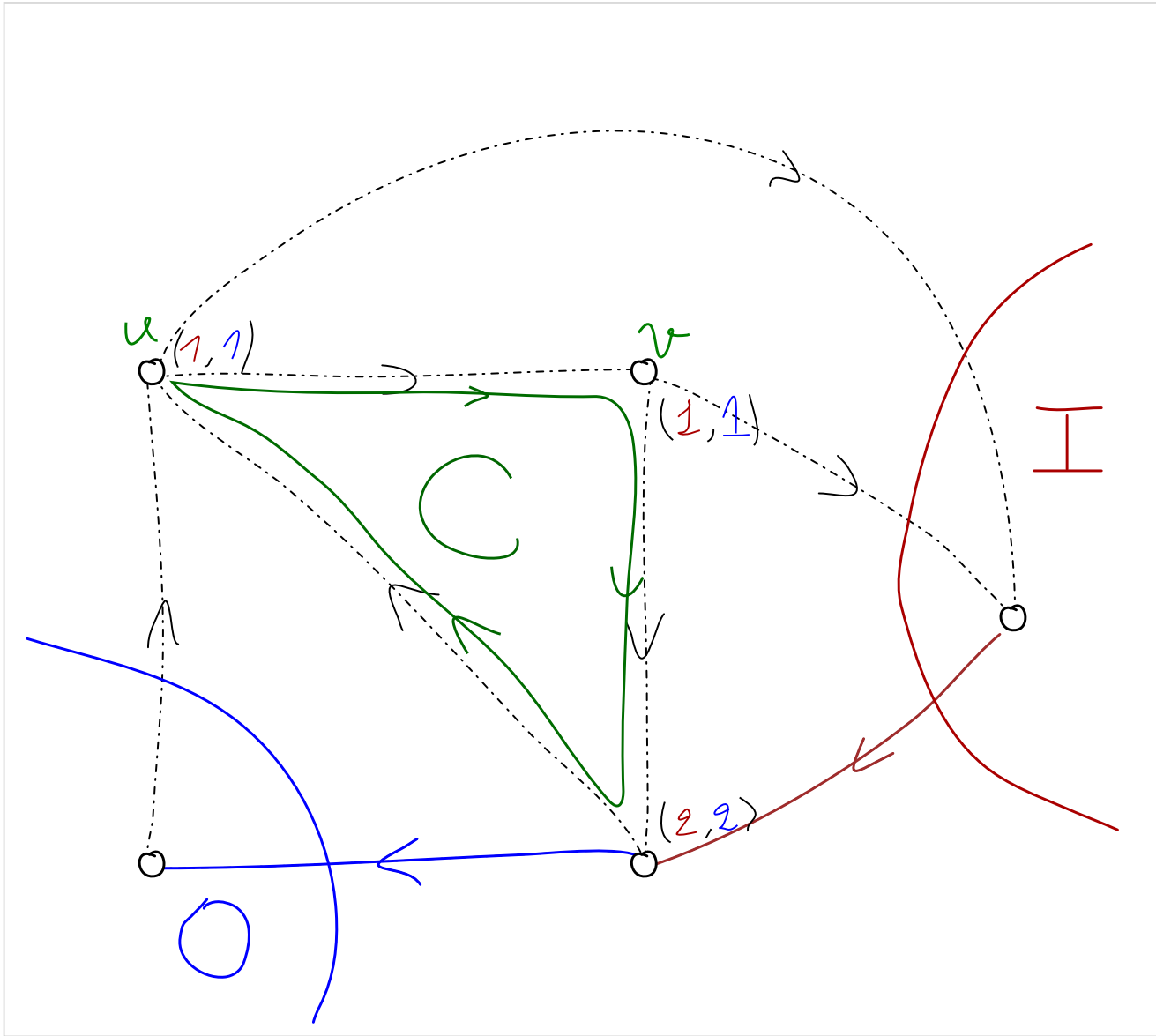
DFS to bitree



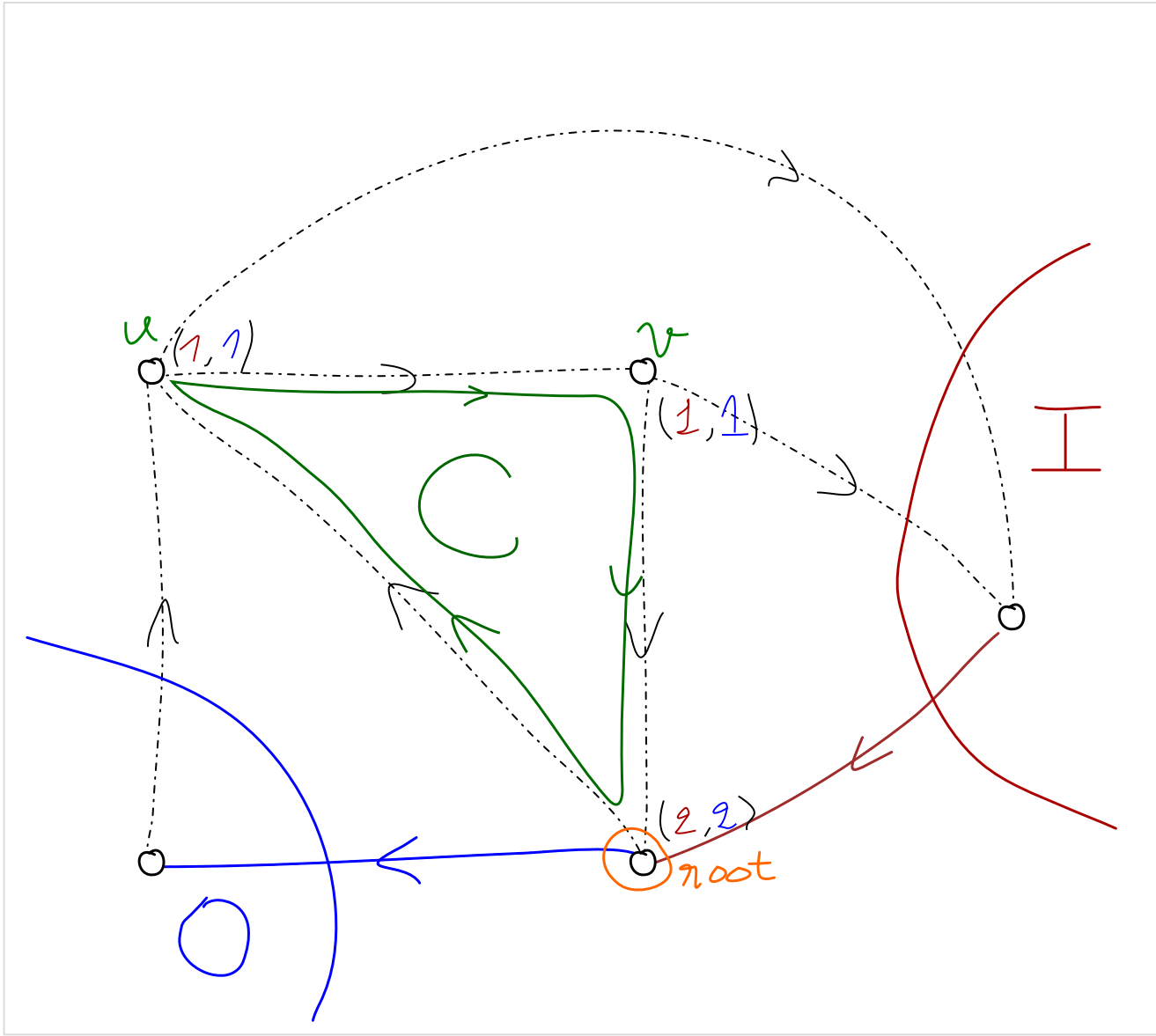
DFS to bitree



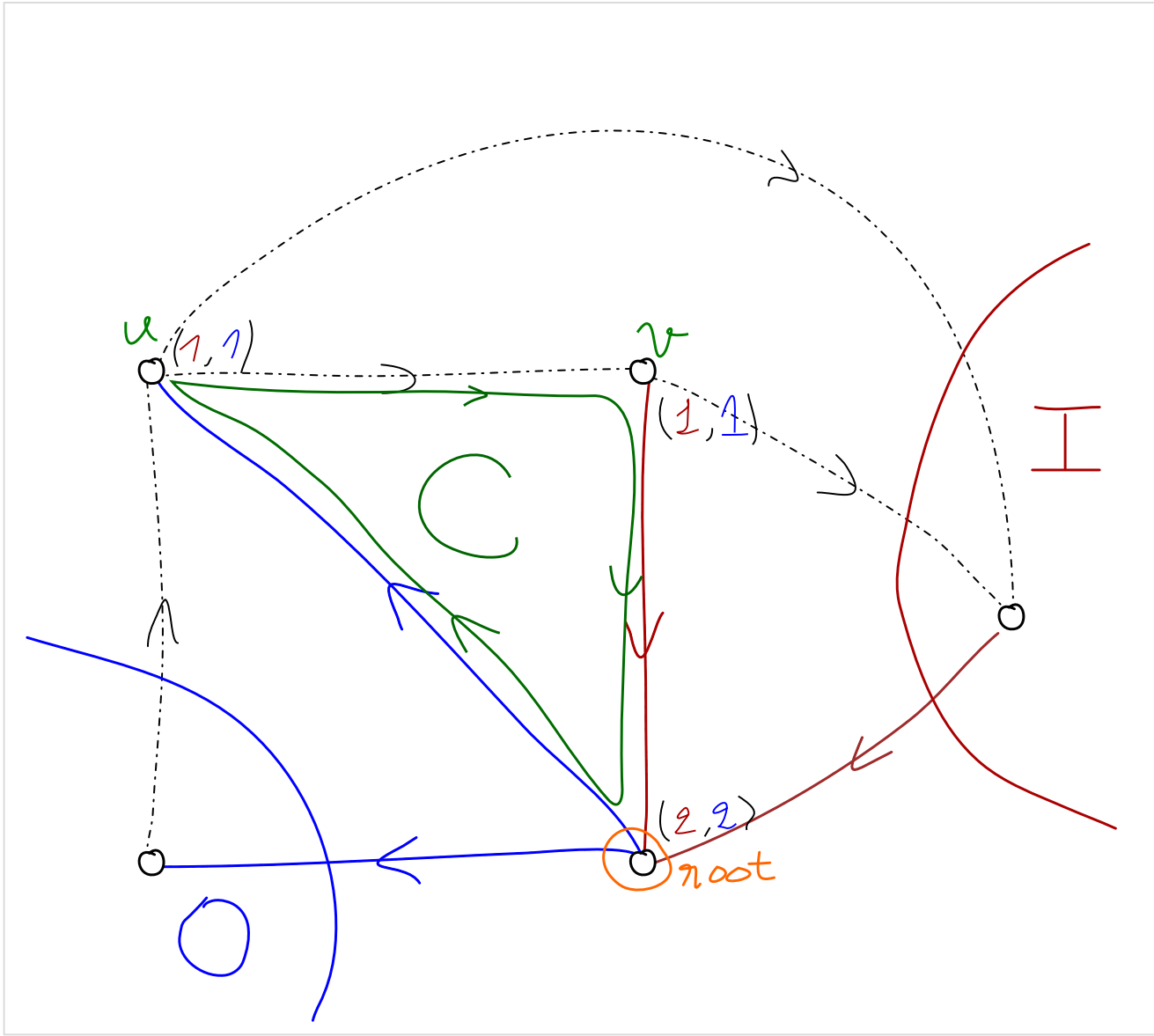
DFS to bitree



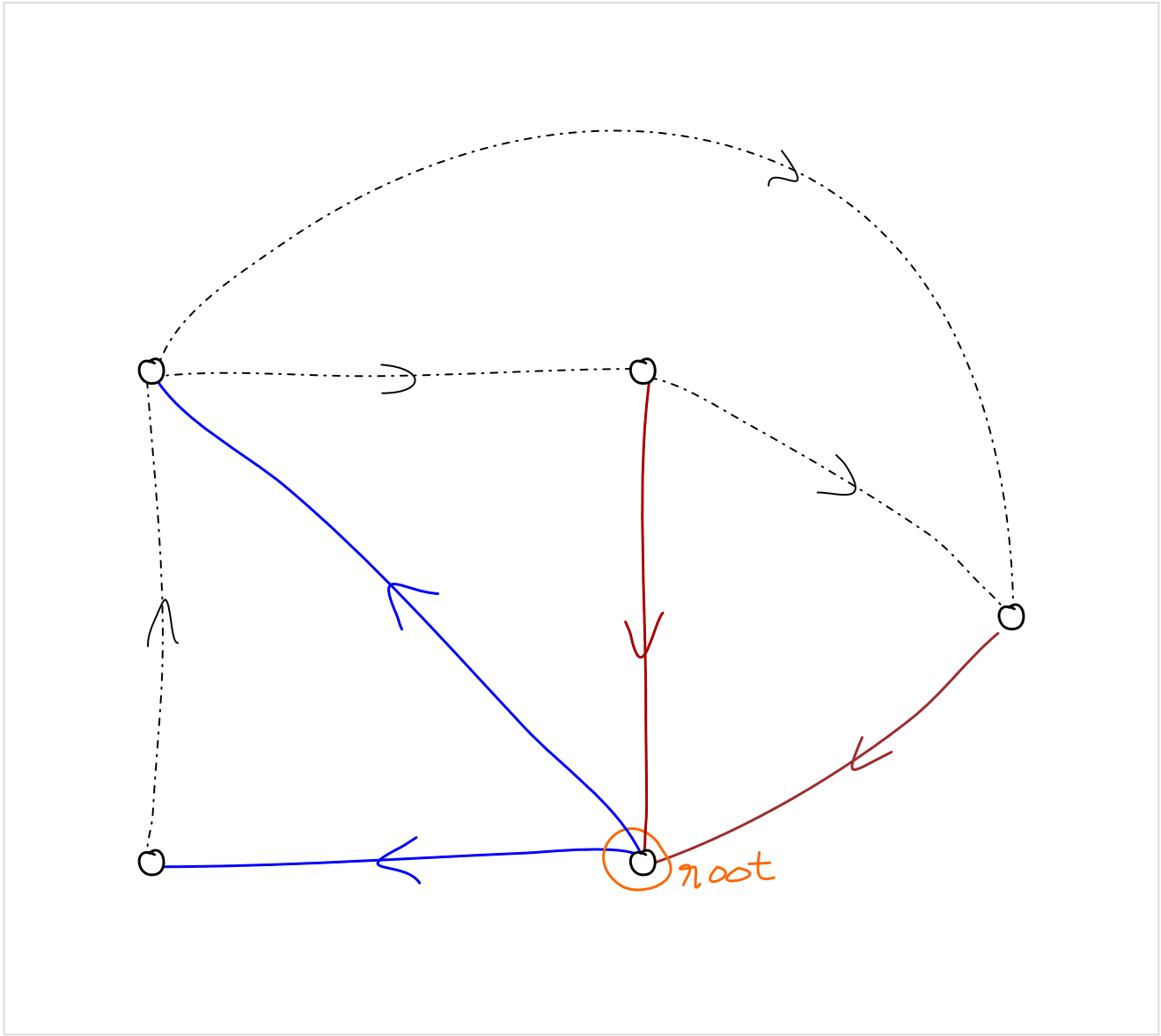
DFS to bitree



DFS to bitree



DFS to bitree



Extensions

The bitree construction generalizes to node weights.

And thus to a subset of nodes U (only pairs in $\binom{\binom{U}{2}}{2}$ are counted).

But not to an arbitrary set $R \subset \binom{\binom{V}{2}}{2}$ of requested pairs.

Conjecture : every strong digraph has a $O(\log n)$ forward cover, i.e. $k = O(\log n)$ node orderings such that any pair $\{x, y\}$ is connected by a path respecting one of the k orderings.

Question : what is the complexity of left-maximal DFS ?

Extensions

The bitree construction generalizes to node weights.

And thus to a subset of nodes U (only pairs in $\left(\binom{U}{2}\right)$ are counted).

But not to an arbitrary set $R \subset \left(\binom{V}{2}\right)$ of requested pairs.

Conjecture : every strong digraph has a $O(\log n)$ forward cover, i.e. $k = O(\log n)$ node orderings such that any pair $\{x, y\}$ is connected by a path respecting one of the k orderings.

Question : what is the complexity of left-maximal DFS ?

Extensions

The bitree construction generalizes to node weights.

And thus to a subset of nodes U (only pairs in $\left(\binom{U}{2}\right)$ are counted).

But not to an arbitrary set $R \subset \left(\binom{V}{2}\right)$ of requested pairs.

Conjecture : every strong digraph has a $O(\log n)$ forward cover, i.e. $k = O(\log n)$ node orderings such that any pair $\{x, y\}$ is connected by a path respecting one of the k orderings.

Question : what is the complexity of left-maximal DFS ?

Extensions

The bitree construction generalizes to node weights.

And thus to a subset of nodes U (only pairs in $\binom{\binom{U}{2}}{}$ are counted).

But not to an arbitrary set $R \subset \binom{\binom{V}{2}}{}$ of requested pairs.

Conjecture : every strong digraph has a $O(\log n)$ forward cover, i.e. $k = O(\log n)$ node orderings such that any pair $\{x, y\}$ is connected by a path respecting one of the k orderings.

Question : what is the complexity of left-maximal DFS ?

Extensions

The bitree construction generalizes to node weights.

And thus to a subset of nodes U (only pairs in $\binom{\binom{U}{2}}{}$ are counted).

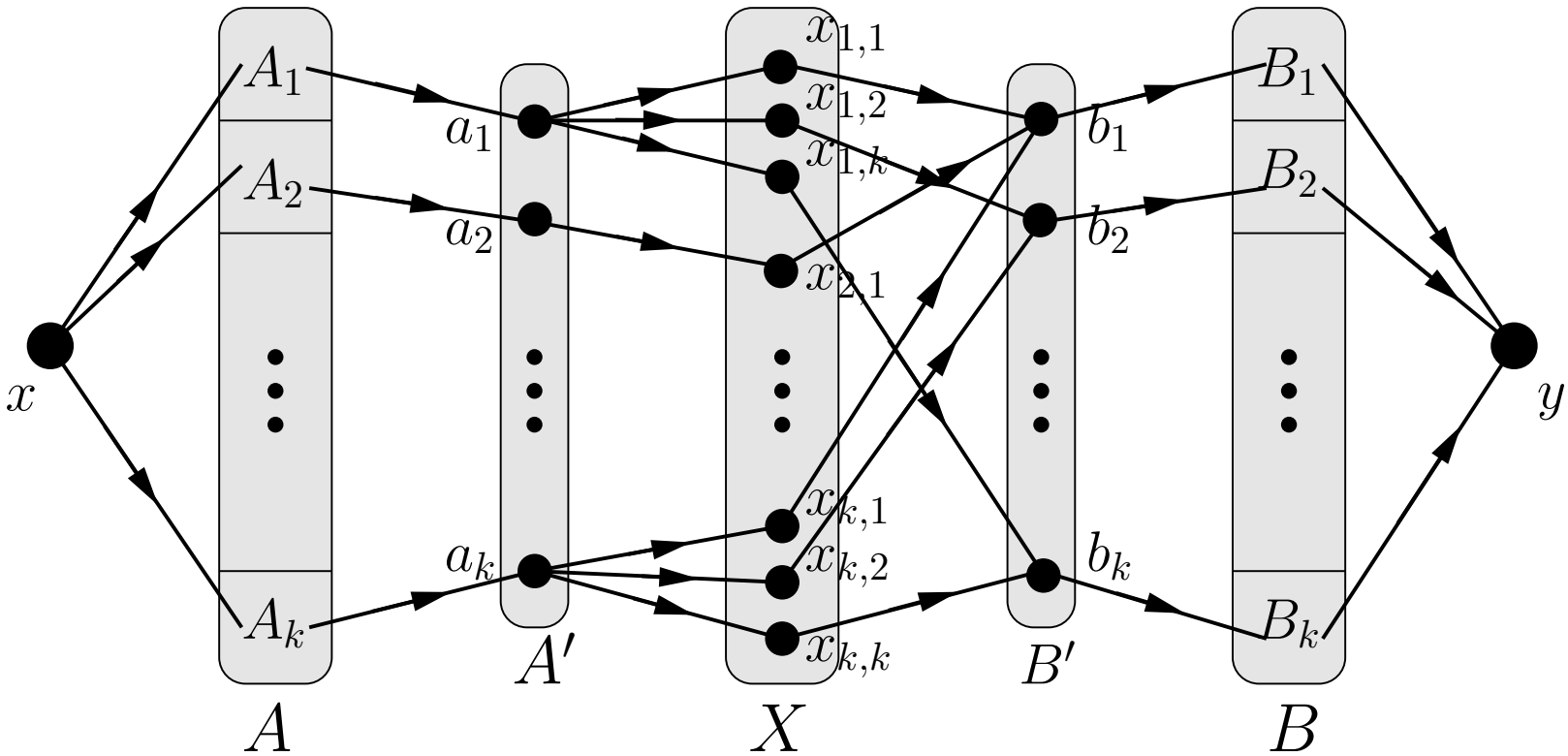
But not to an arbitrary set $R \subset \binom{\binom{V}{2}}{}$ of requested pairs.

Conjecture : every strong digraph has a $O(\log n)$ forward cover, i.e. $k = O(\log n)$ node orderings such that any pair $\{x, y\}$ is connected by a path respecting one of the k orderings.

Question : what is the complexity of left-maximal DFS ?

Thanks.

A good node ordering may give poor bitrees.



Open problem

MRET is in APX for strong digraphs.

Does it hold also for general digraphs?

Open problem

MRET is in APX for strong digraphs.

Does it hold also for general digraphs?

Edge-disjoint in-tree and out-tree : related work

Related problem : Find an in-tree and an out-tree with same root that are edge disjoint and have size $\Omega(n)$.

It is NP-hard to decide if a strong digraph has such a pair. [Bang-Jensen 1991]

Conjecture : There exist c such that any c-edge-connected digraph has such a pair. [Thomassen 1989]

Th : Every strong digraph has an in-tree and an out-tree with same root that are vertex disjoint and both have size $n/6$. [Bessy, Thomassé, Viennot 2023]

Question : How many pairs of in-tree, out-tree are needed to cover all pairs ?

Edge-disjoint in-tree and out-tree : related work

Related problem : Find an in-tree and an out-tree with same root that are edge disjoint and have size $\Omega(n)$.

It is NP-hard to decide if a strong digraph has such a pair. [Bang-Jensen 1991]

Conjecture : There exist c such that any c-edge-connected digraph has such a pair. [Thomassen 1989]

Th : Every strong digraph has an in-tree and an out-tree with same root that are vertex disjoint and both have size $n/6$. [Bessy, Thomassé, Viennot 2023]

Question : How many pairs of in-tree, out-tree are needed to cover all pairs ?

Edge-disjoint in-tree and out-tree : related work

Related problem : Find an in-tree and an out-tree with same root that are edge disjoint and have size $\Omega(n)$.

It is NP-hard to decide if a strong digraph has such a pair. [Bang-Jensen 1991]

Conjecture : There exist c such that any c-edge-connected digraph has such a pair. [Thomassen 1989]

Th : Every strong digraph has an in-tree and an out-tree with same root that are vertex disjoint and both have size $n/6$. [Bessy, Thomassé, Viennot 2023]

Question : How many pairs of in-tree, out-tree are needed to cover all pairs ?

Edge-disjoint in-tree and out-tree : related work

Related problem : Find an in-tree and an out-tree with same root that are edge disjoint and have size $\Omega(n)$.

It is NP-hard to decide if a strong digraph has such a pair. [Bang-Jensen 1991]

Conjecture : There exist c such that any c-edge-connected digraph has such a pair. [Thomassen 1989]

Th : Every strong digraph has an in-tree and an out-tree with same root that are vertex disjoint and both have size $n/6$. [Bessy, Thomassé, Viennot 2023]

Question : How many pairs of in-tree, out-tree are needed to cover all pairs ?

Edge-disjoint in-tree and out-tree : related work

Related problem : Find an in-tree and an out-tree with same root that are edge disjoint and have size $\Omega(n)$.

It is NP-hard to decide if a strong digraph has such a pair. [Bang-Jensen 1991]

Conjecture : There exist c such that any c-edge-connected digraph has such a pair. [Thomassen 1989]

Th : Every strong digraph has an in-tree and an out-tree with same root that are vertex disjoint and both have size $n/6$. [Bessy, Thomassé, Viennot 2023]

Question : How many pairs of in-tree, out-tree are needed to cover all pairs?

Undirected graph temporalisation is quite known

Given an undirected graph G, deciding whether there exists an assignment of one time label per edge such that all pairs are temporally connected is NP-complete. [Göbel, Cerdeira, Veldman 1991]

Approximation is obvious : take a spanning tree and assign time labels that connect $(n/2)^2$ pairs.

Related (Gossip/telephone problem [Bumby 1981]) : The minimum number of time labels allowing to temporally connect all pairs at least $2n-4$, and equals $2n-4$ if G has a C_4 (one or two time labels per edge).

Undirected graph temporalisation is quite known

Given an undirected graph G, deciding whether there exists an assignment of one time label per edge such that all pairs are temporally connected is NP-complete. [Göbel, Cerdeira, Veldman 1991]

Approximation is obvious : take a spanning tree and assign time labels that connect $(n/2)^2$ pairs.

Related (Gossip/telephone problem [Bumby 1981]) : The minimum number of time labels allowing to temporally connect all pairs at least $2n-4$, and equals $2n-4$ if G has a C_4 (one or two time labels per edge).

Undirected graph temporalisation is quite known

Given an undirected graph G, deciding whether there exists an assignment of one time label per edge such that all pairs are temporally connected is NP-complete. [Göbel, Cerdeira, Veldman 1991]

Approximation is obvious : take a spanning tree and assign time labels that connect $(n/2)^2$ pairs.

Related (Gossip/telephone problem [Bumby 1981]) : The minimum number of time labels allowing to temporally connect all pairs at least $2n - 4$, and equals $2n - 4$ if G has a C_4 (one or two time labels per edge).