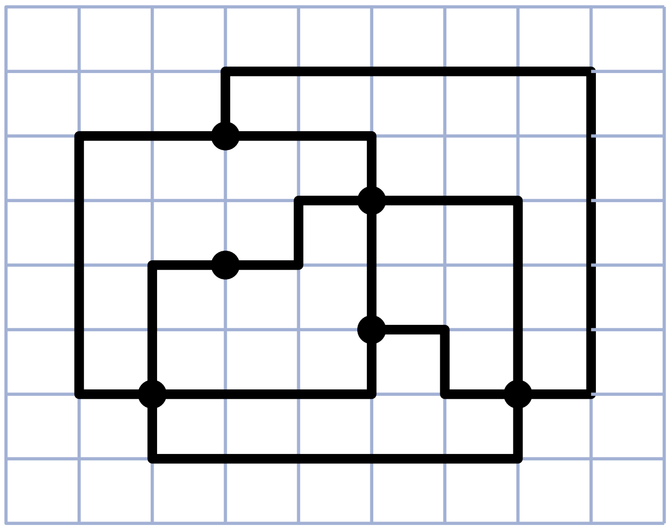
Karlsruhe Institute of Technology

# Algorithms for graph visualization

## Incremental algorithms. Orthogonal drawing.

WINTER SEMESTER 2018/2019

**Tamara Mchedlidze**



# Definition

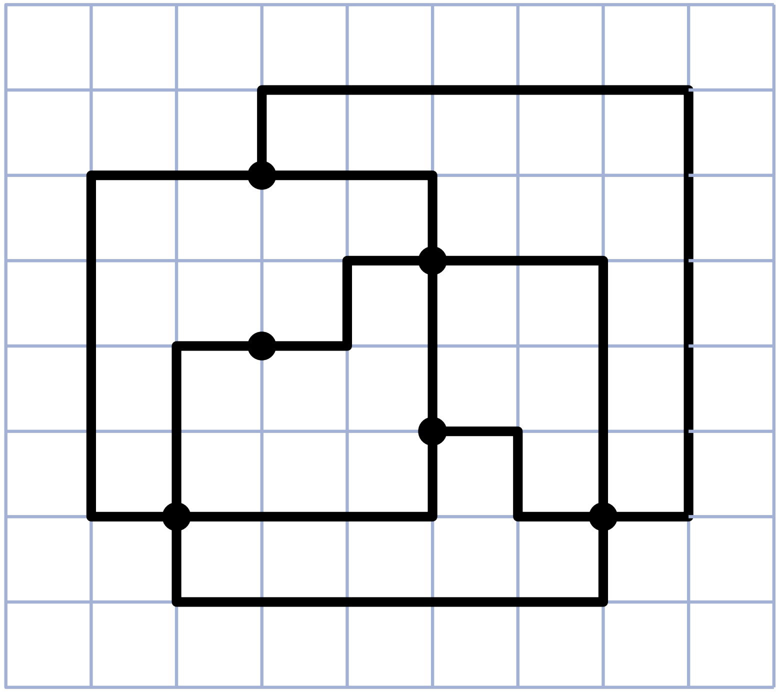
## Definition: Orthogonal Drawing

A drawing Γ of a graph $G = (V, E)$ is called orthogonal if its veritices are drawn as points and each edge is represented as a sequence of alternating horizontal and vertical segments.

# Definition

## Definition: Orthogonal Drawing

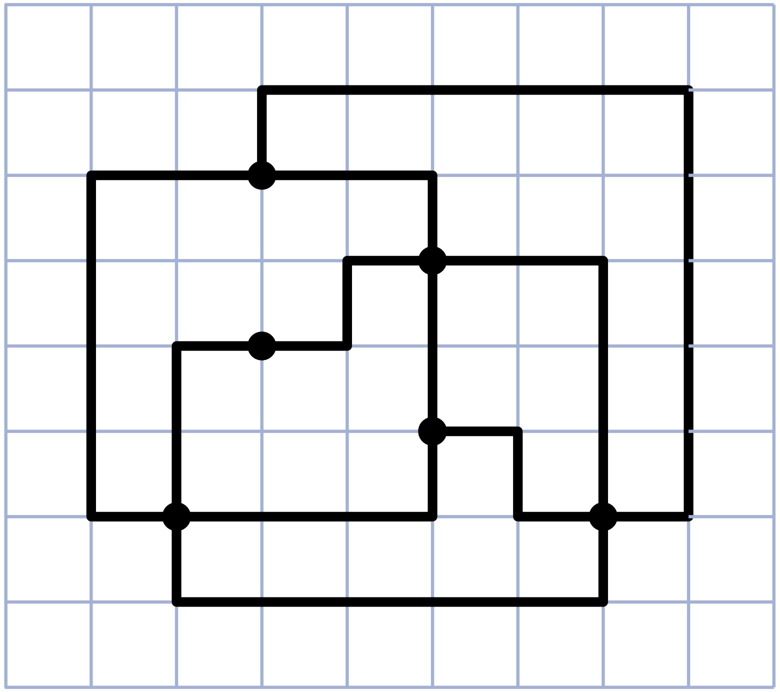
A drawing Γ of a graph $G = (V, E)$ is called orthogonal if its veritices are drawn as points and each edge is represented as a sequence of alternating horizontal and vertical segments.



# Definition

## Definition: Orthogonal Drawing

A drawing Γ of a graph $G = (V, E)$ is called orthogonal if its veritices are drawn as points and each edge is represented as a sequence of alternating horizontal and vertical segments.

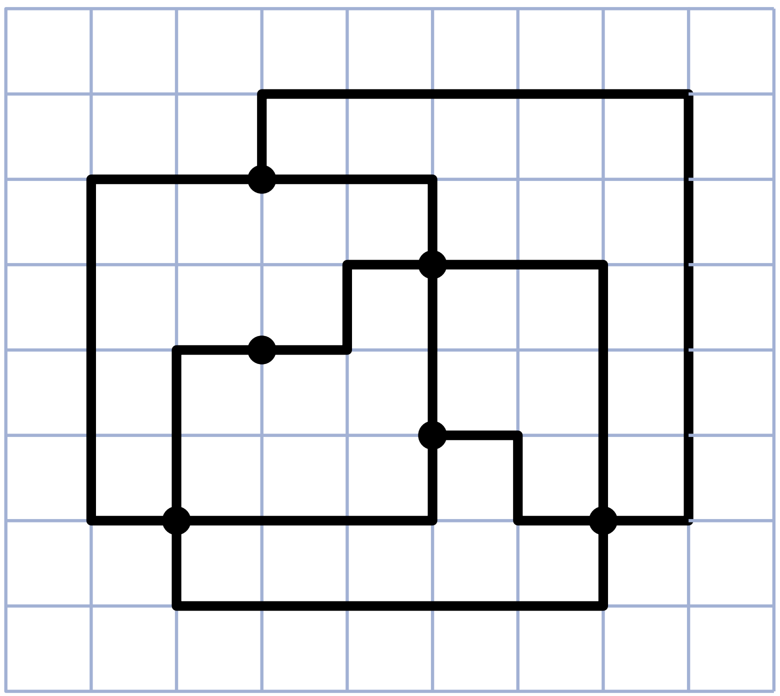


- Edges lie on the grid, i.e, bends lie on grid points

# Definition

## Definition: Orthogonal Drawing

A drawing Γ of a graph $G = (V, E)$ is called orthogonal if its veritices are drawn as points and each edge is represented as a sequence of alternating horizontal and vertical segments.

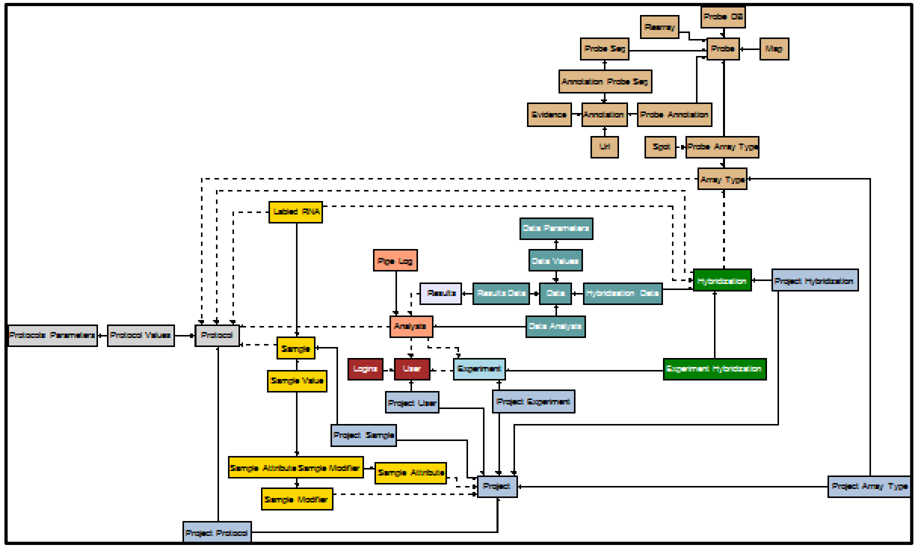


- Edges lie on the grid, i.e, bends lie on grid points

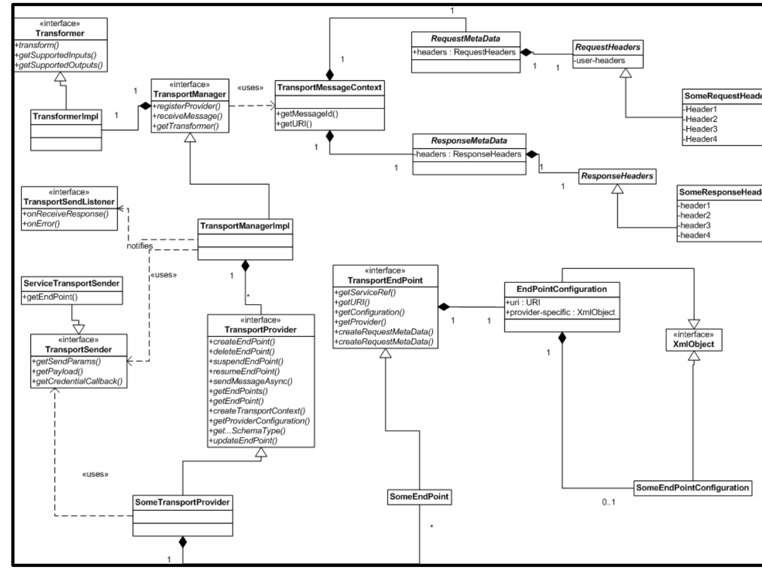
---

- degree of each vertex has to be at most 4

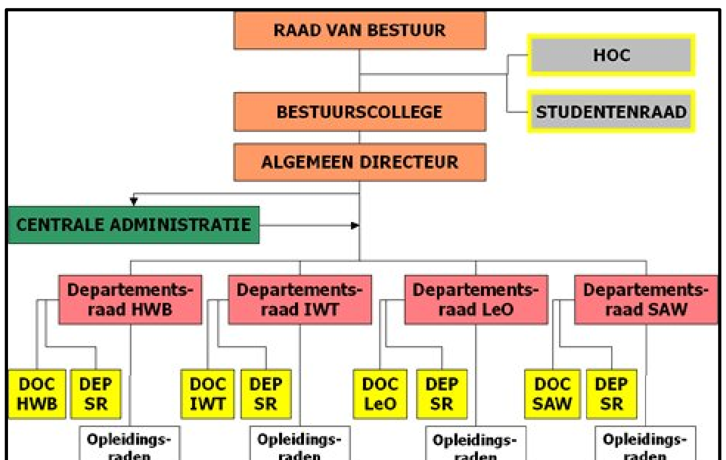
# Orthogonal Layout



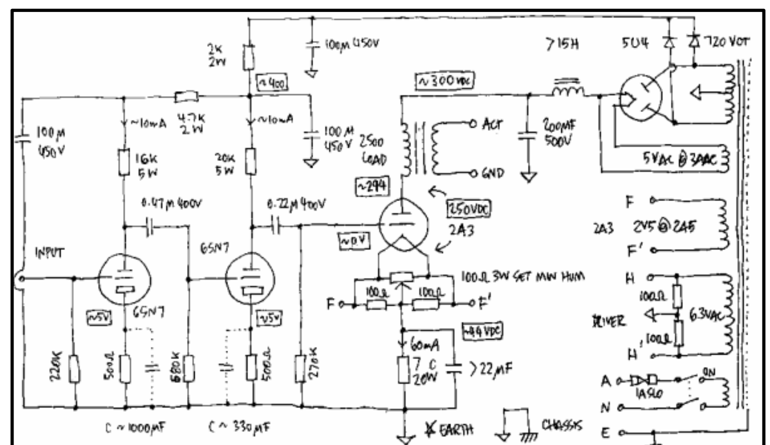
ER diagram in OGDF



UML diagram by Oracle




Organigram of HS Limburg



Circuit diagram by Jeff Atwood

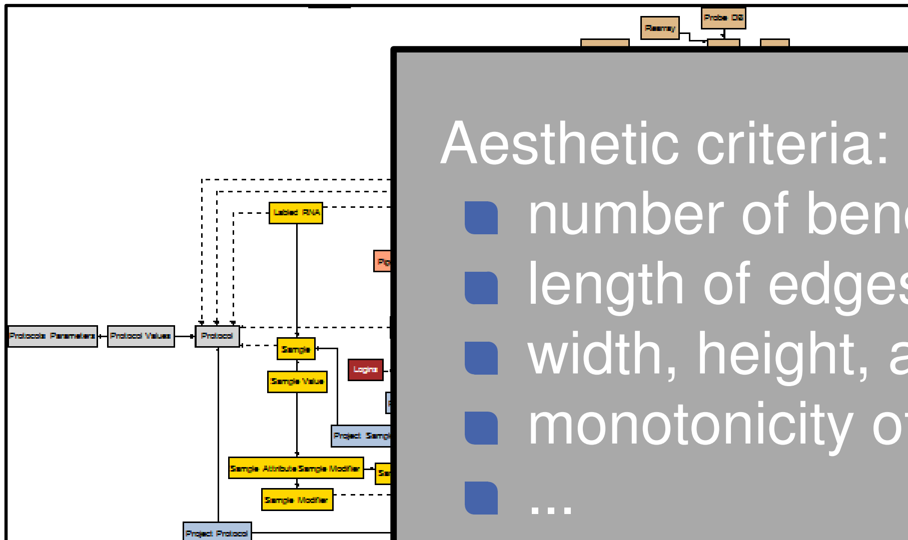


Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

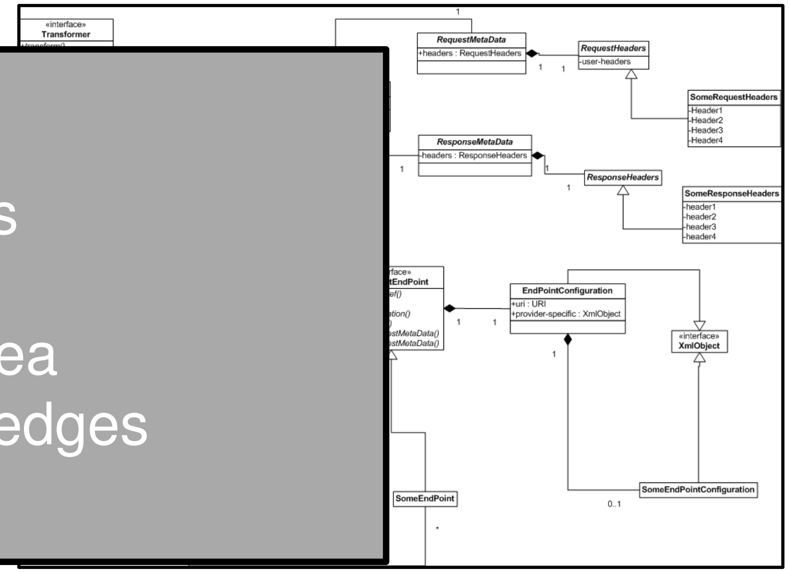
# Orthogonal Layout

Aesthetic criteria:
- number of bends
- length of edges
- width, height, area
- monotonicity of edges
- ...

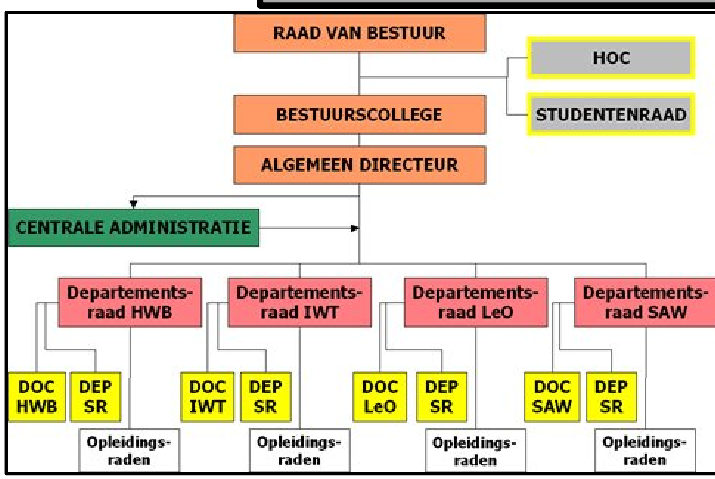
ER diagram in OGDF

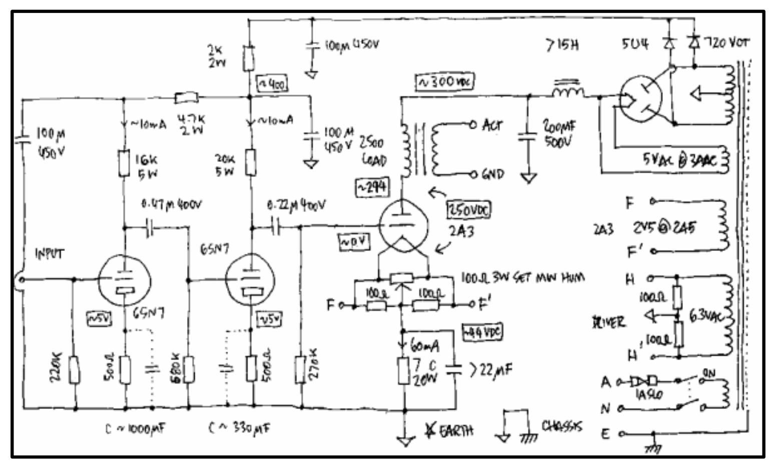


UML diagram by Oracle



Organigram of HS Limburg



Circuit diagram by Jeff Atwood

# Overview

- Our tool today: $st$-ordering
- Algorithm of Biedl&Kant
- Properties of the drawing, Planarity
- Construction of $st$-ordering through ear decomposition


Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

# $st$-ordering

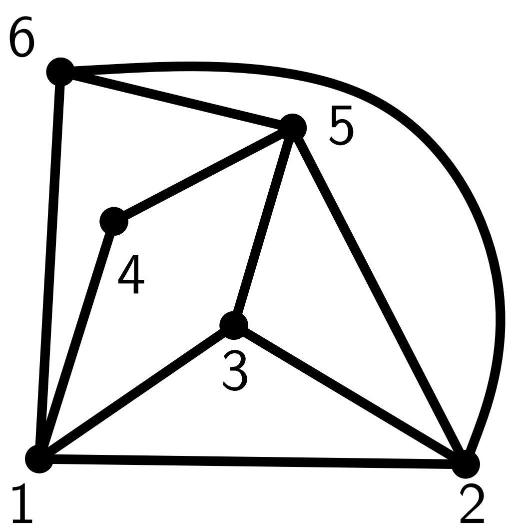
## Definition: $st$-ordering

An $st$-ordering of a graph $G = (V, E)$ is an ordering of the vertices $\{v_1, v_2, \ldots, v_n\}$, such that for each $j$, $2 \leq j \leq n - 1$, vertex $v_j$ has at least one neighbour $v_i$ with $i < j$, and at least one neighbour $v_k$ with $k > j$.

# $st$-ordering

**Definition: $st$-ordering**

An $st$-ordering of a graph $G = (V, E)$ is an ordering of the vertices $\{v_1, v_2, \ldots, v_n\}$, such that for each $j$, $2 \leq j \leq n - 1$, vertex $v_j$ has at least one neighbour $v_i$ with $i < j$, and at least one neighbour $v_k$ with $k > j$.

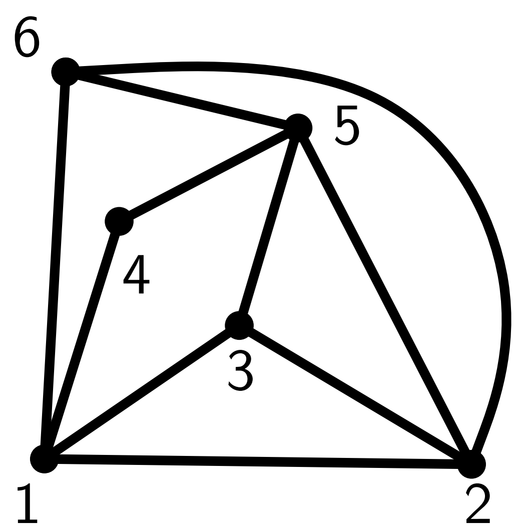



Example of an $st$-ordering

# $st$-ordering

## Definition: $st$-ordering

An $st$-ordering of a graph $G = (V, E)$ is an ordering of the vertices $\{v_1, v_2, \ldots, v_n\}$, such that for each $j$, $2 \leq j \leq n-1$, vertex $v_j$ has at least one neighbour $v_i$ with $i < j$, and at least one neighbour $v_k$ with $k > j$.



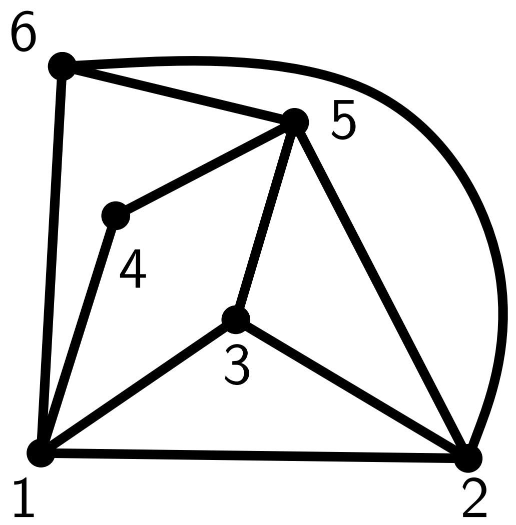
Example of an $st$-ordering

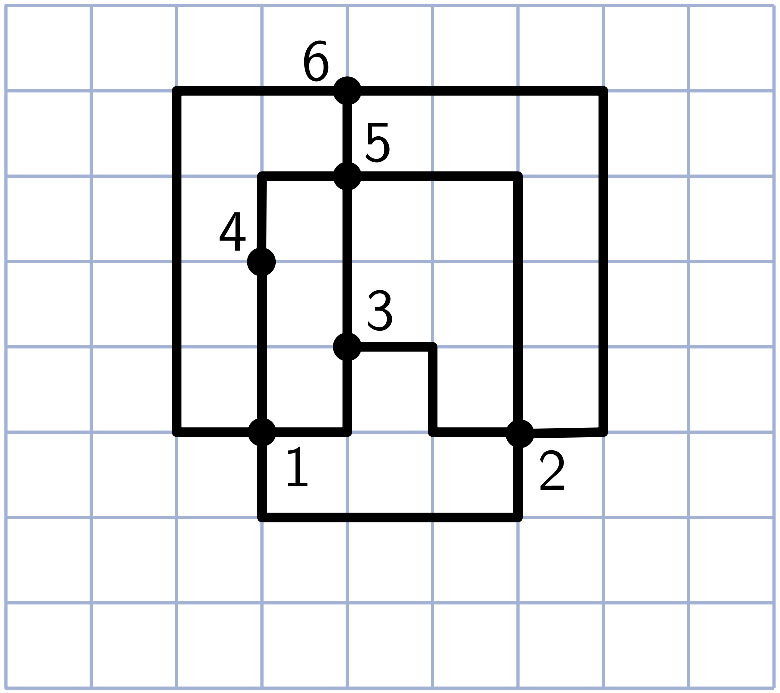
## Theorem [Lempel, Even, Cederbaum, 66]

Let $G$ be a biconnected graph $G$ and let $s$, $t$ be vetices of $G$. $G$ has an $st$-ordering such that $s$ appears as the first and $t$ as the last vertex in this ordering.

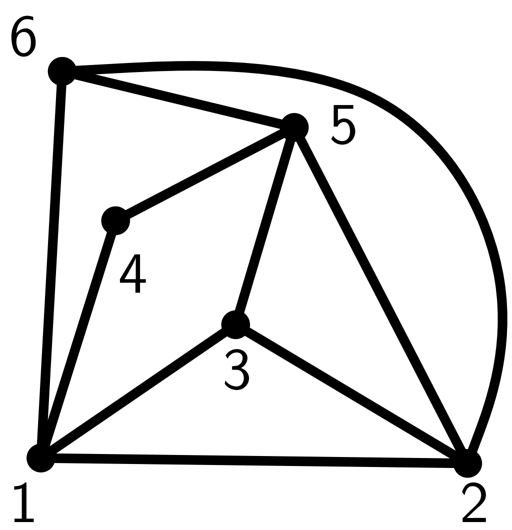

Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

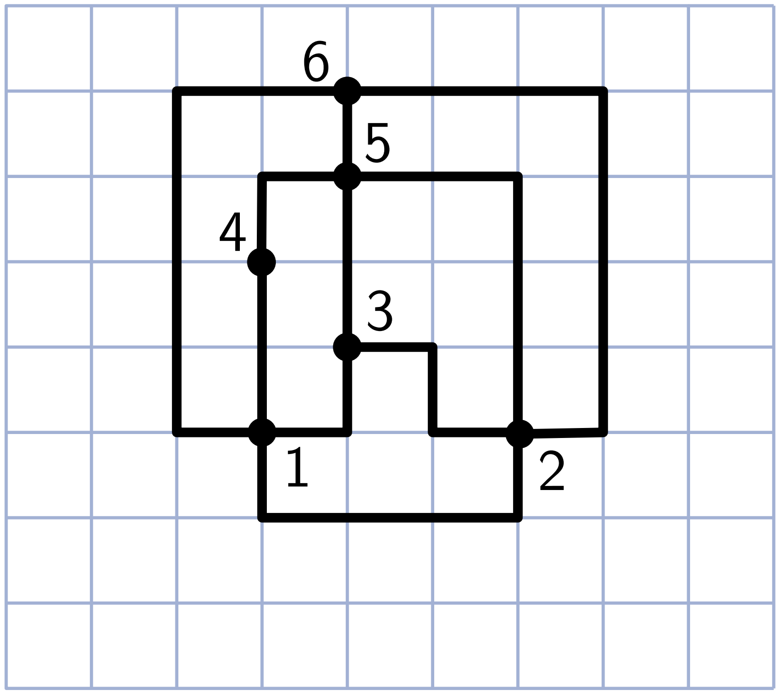
# Biedl & Kant Orthogonal Drawing Algorithm



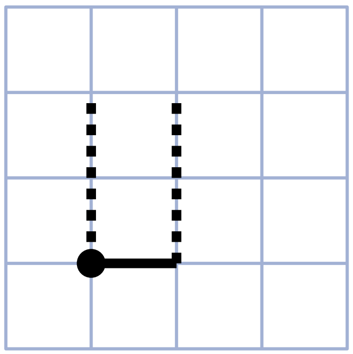



# Biedl & Kant Orthogonal Drawing Algorithm

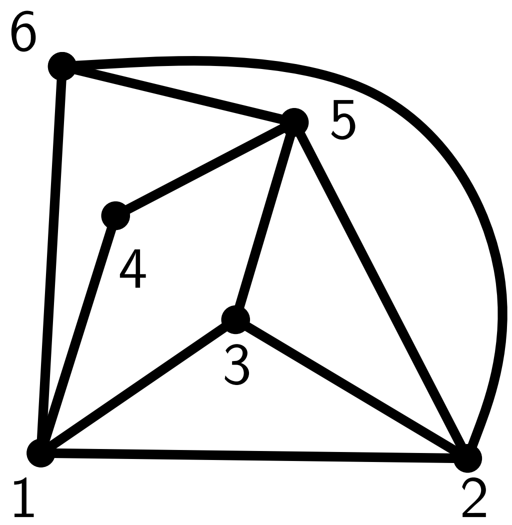


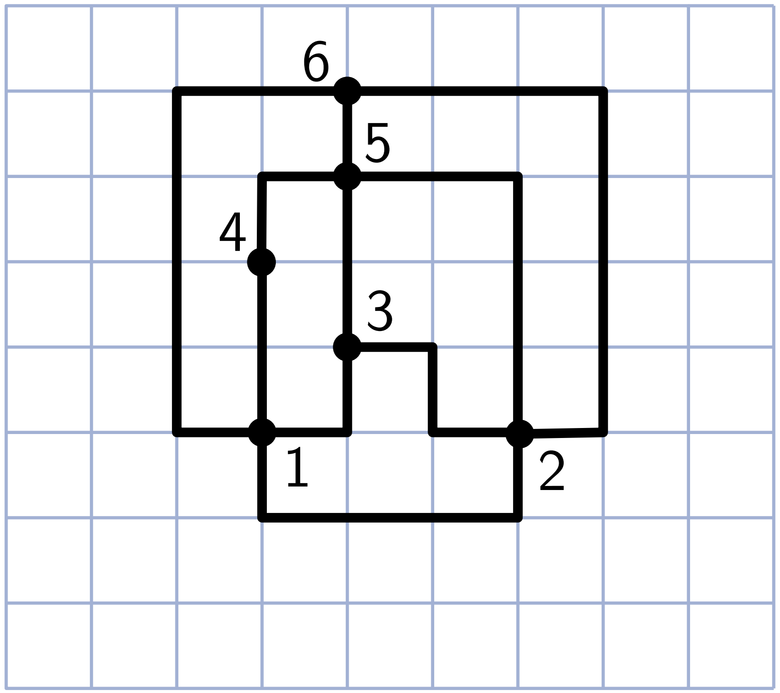




first vertex

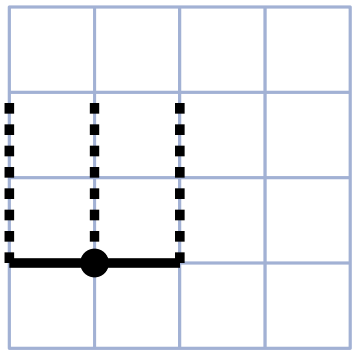


# Biedl & Kant Orthogonal Drawing Algorithm



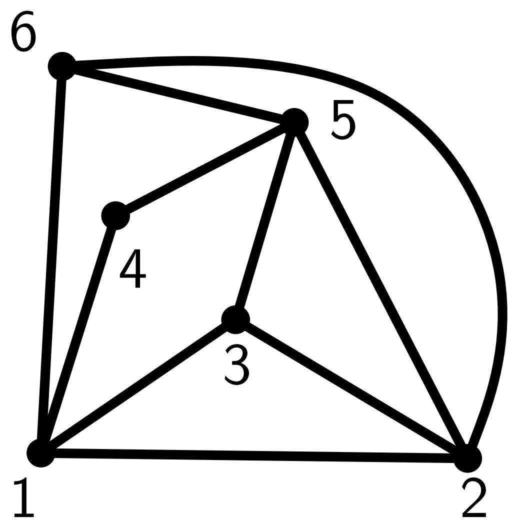

first vertex

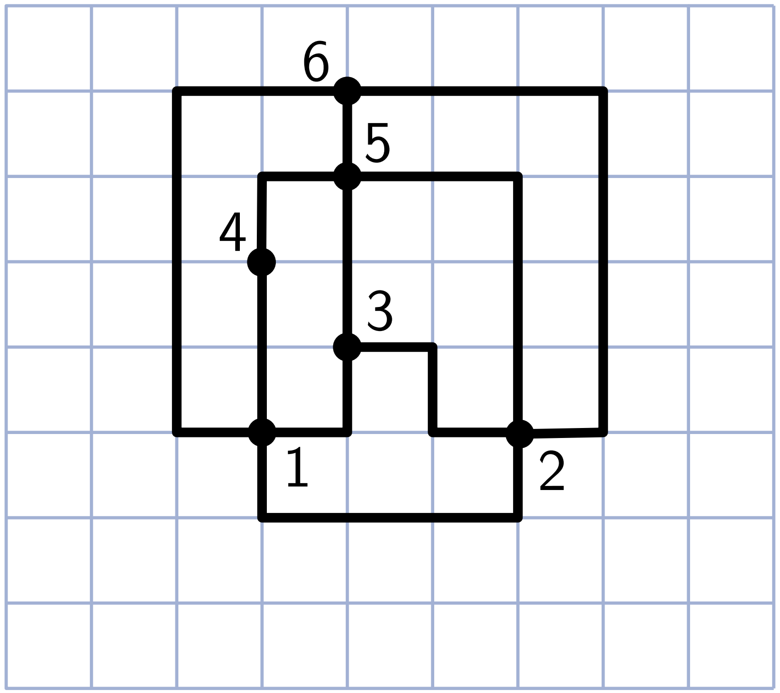




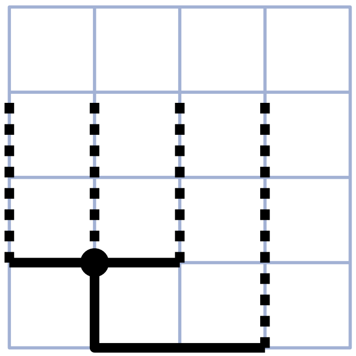
Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

# Biedl & Kant Orthogonal Drawing Algorithm

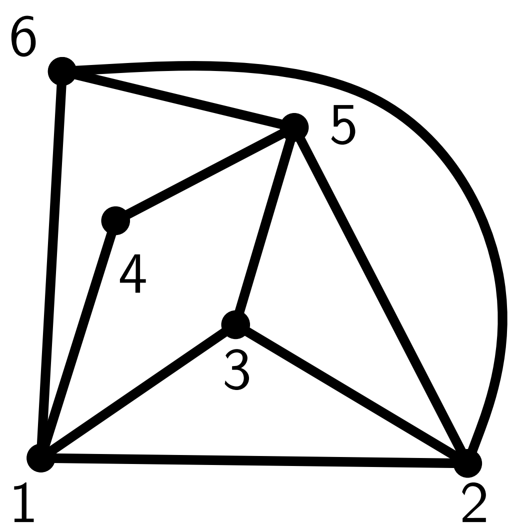


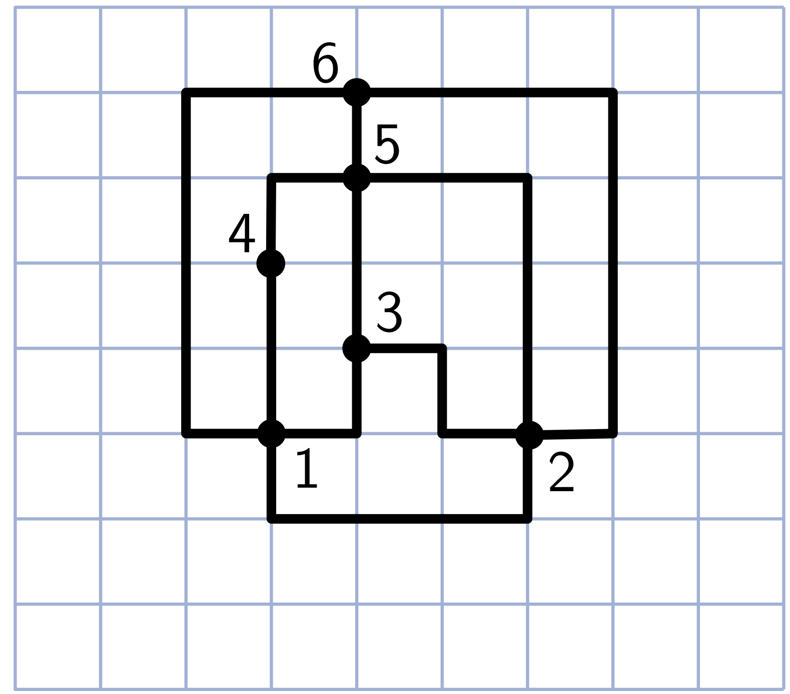




first vertex

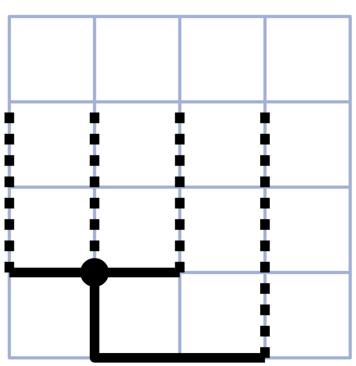


# Biedl & Kant Orthogonal Drawing Algorithm

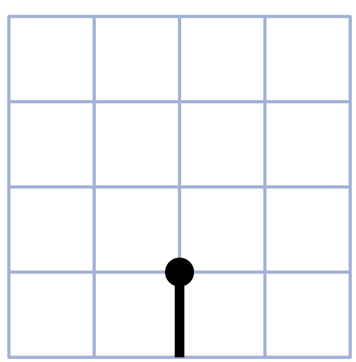


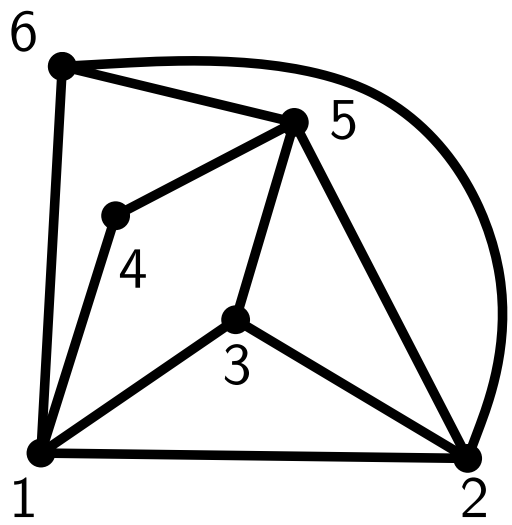

first vertex

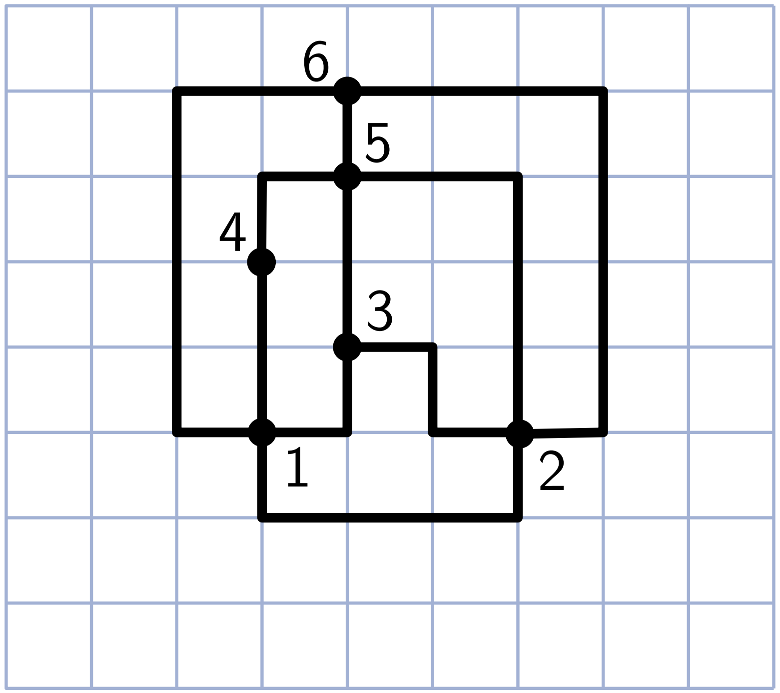


indegree = 1

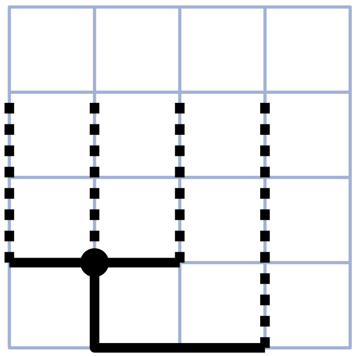


# Biedl & Kant Orthogonal Drawing Algorithm

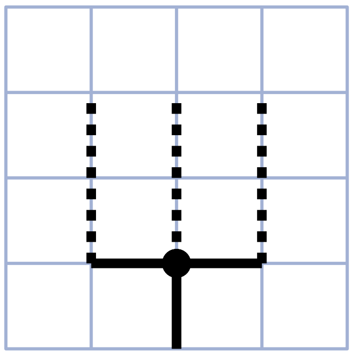



first vertex



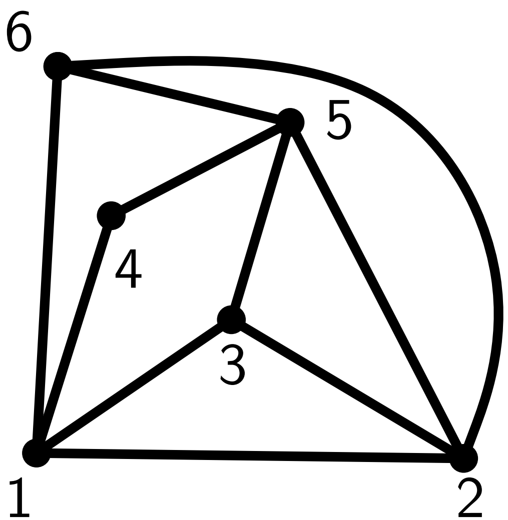
indegree = 1

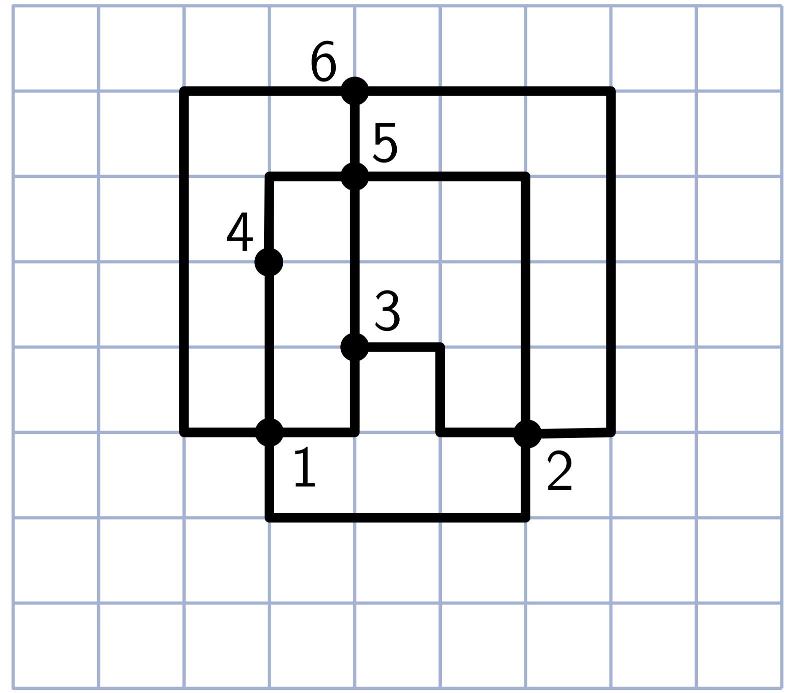




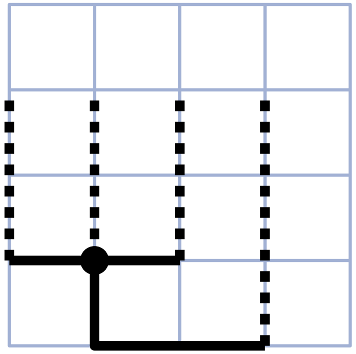
Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

# Biedl & Kant Orthogonal Drawing Algorithm

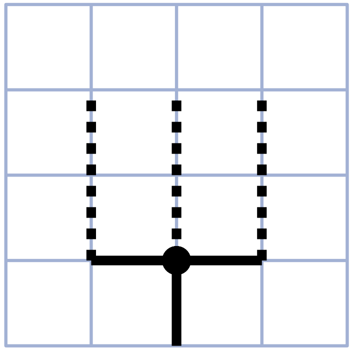


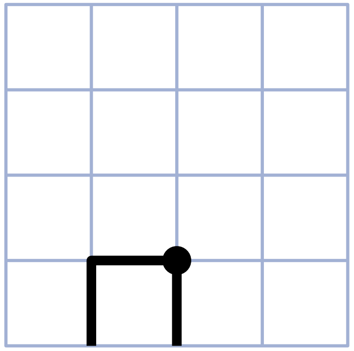

first vertex



indegree = 1



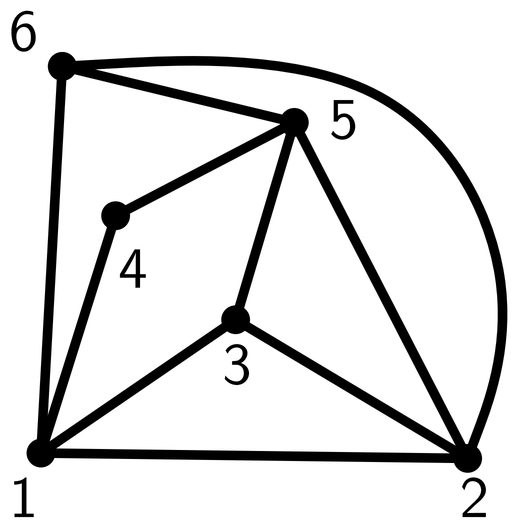
indegree = 2

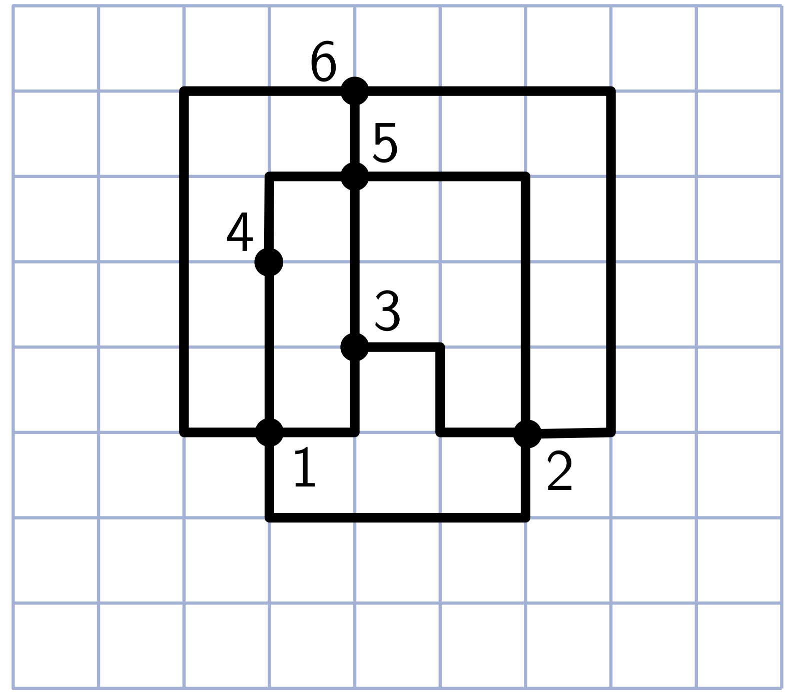




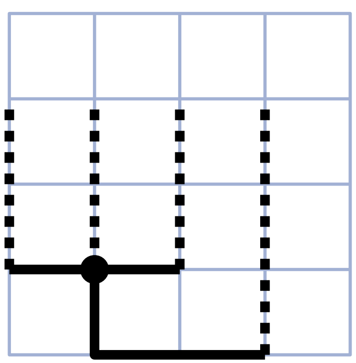
Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

# Biedl & Kant Orthogonal Drawing Algorithm

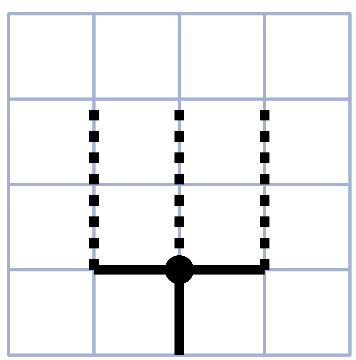


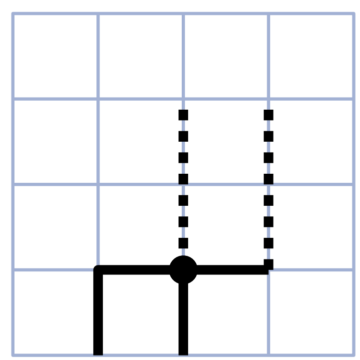

first vertex



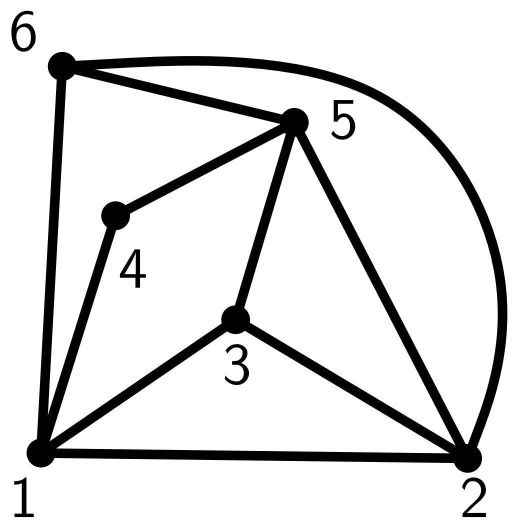
indegree = 1

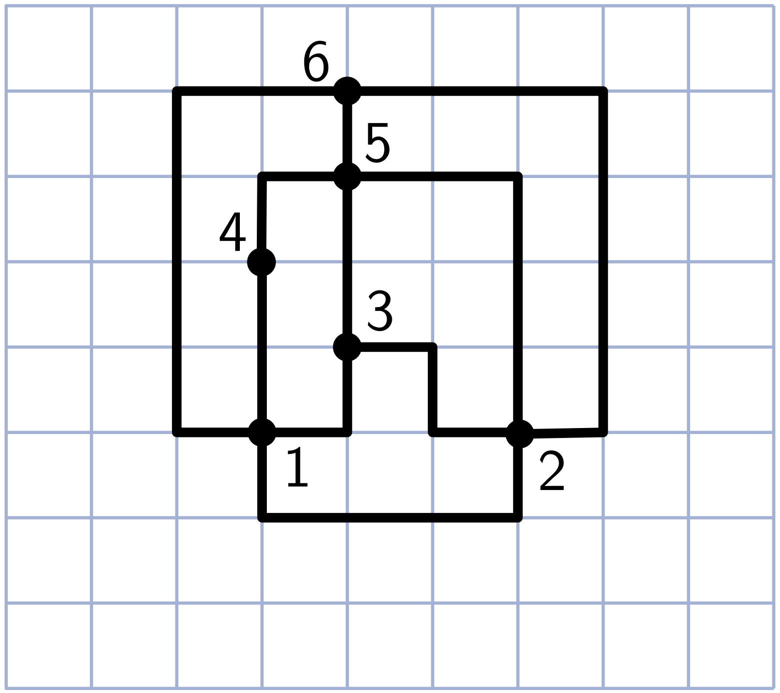


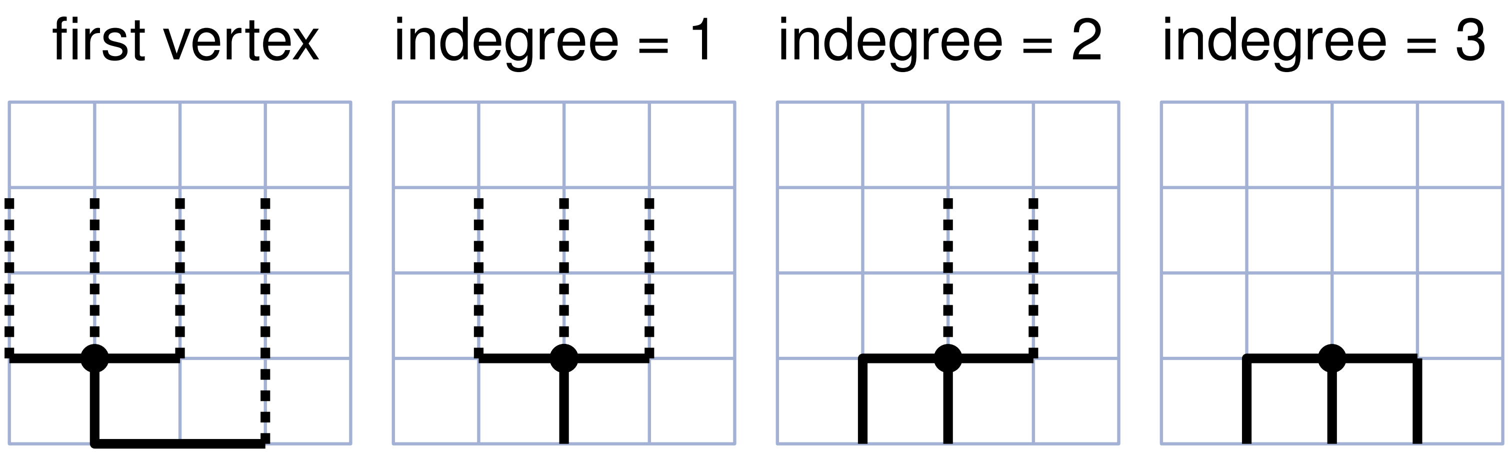
indegree = 2



# Biedl & Kant Orthogonal Drawing Algorithm



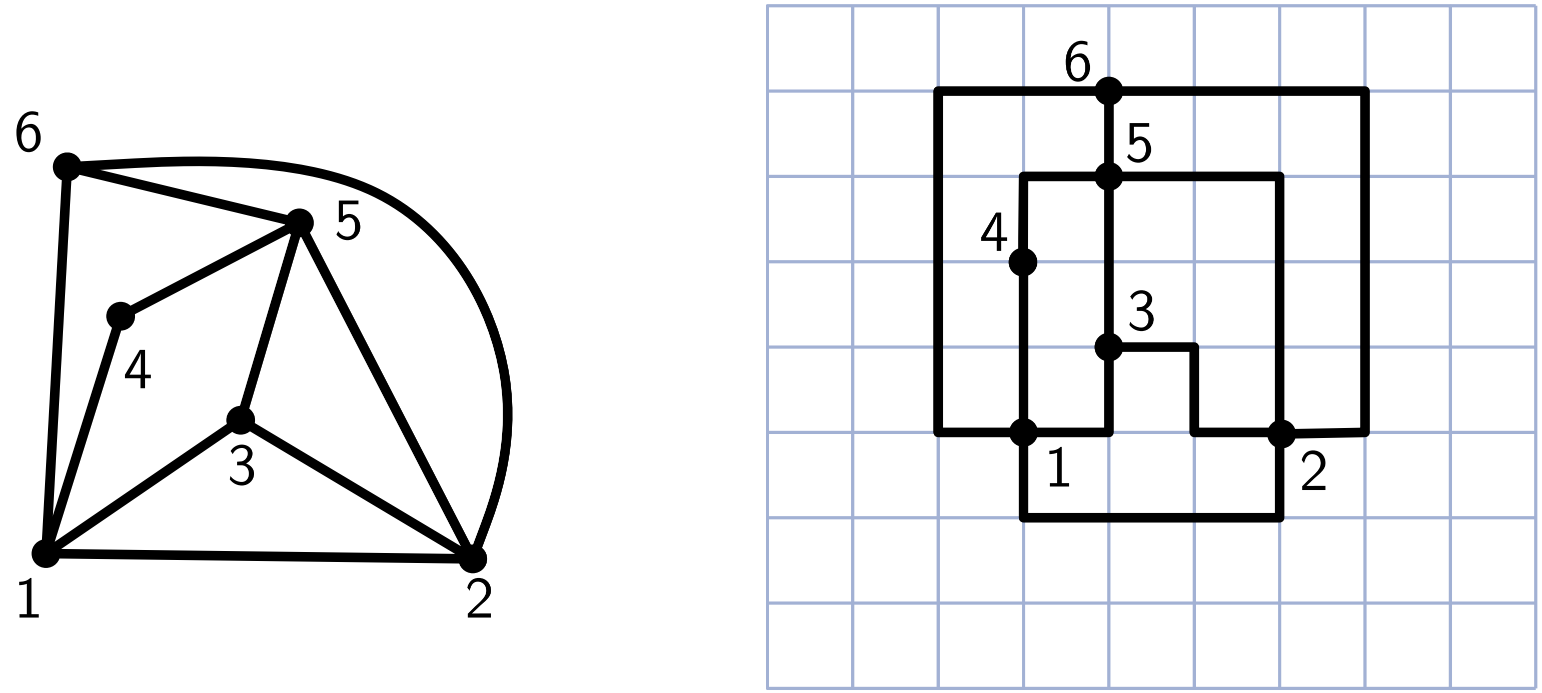



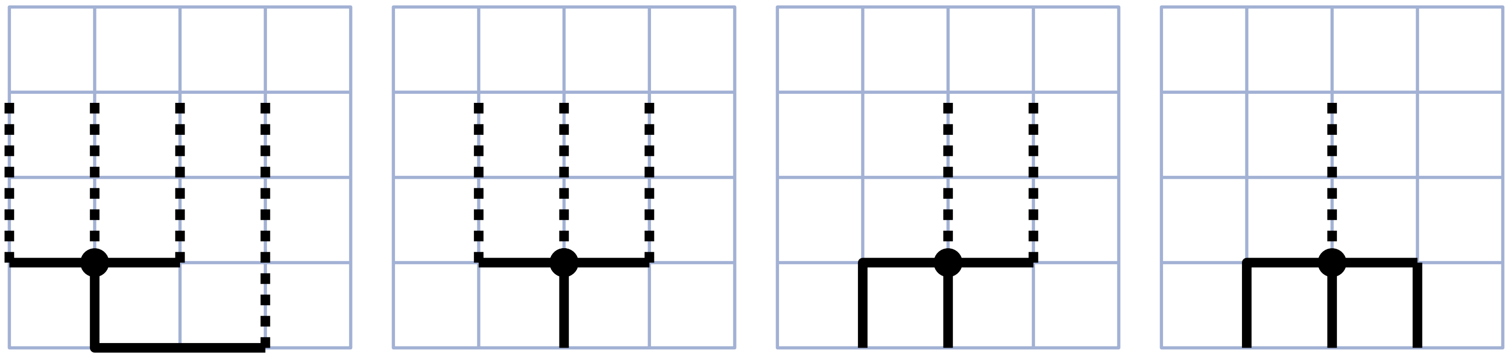



Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

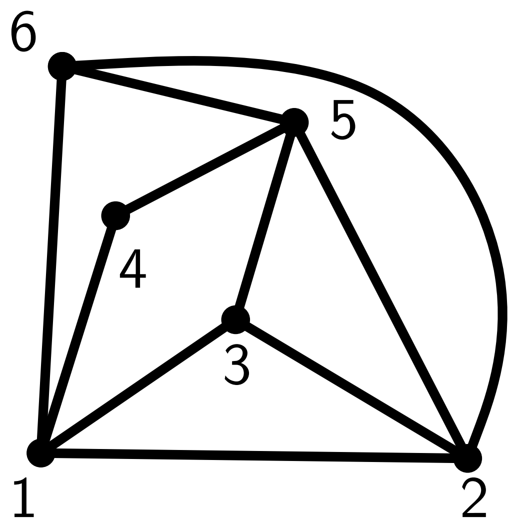
# Biedl & Kant Orthogonal Drawing Algorithm

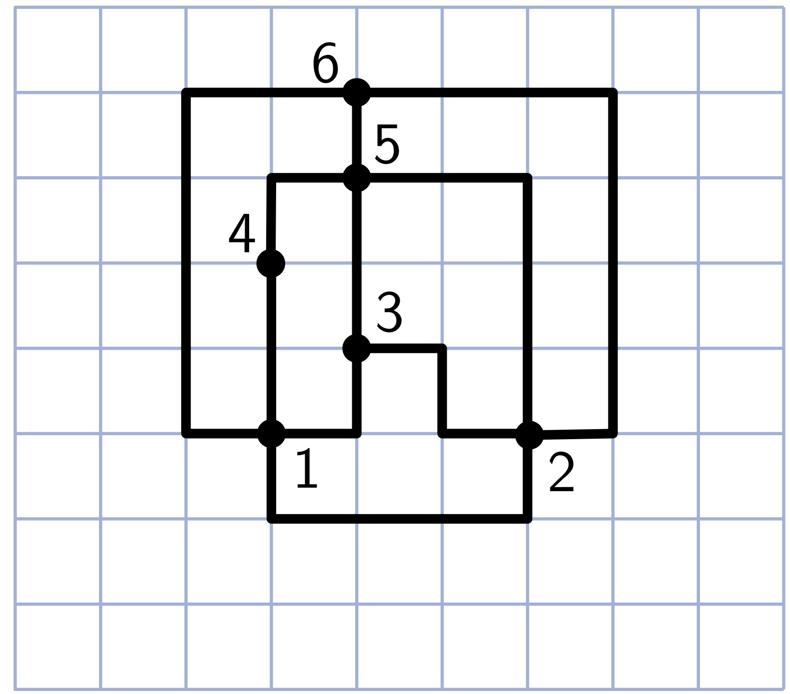


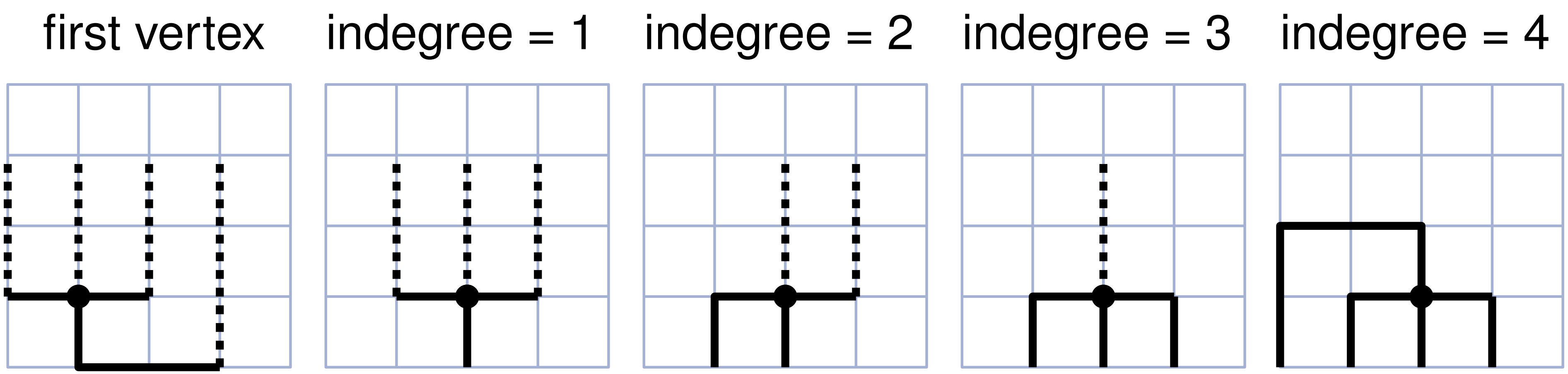

first vertex | indegree = 1 | indegree = 2 | indegree = 3



# Biedl & Kant Orthogonal Drawing Algorithm



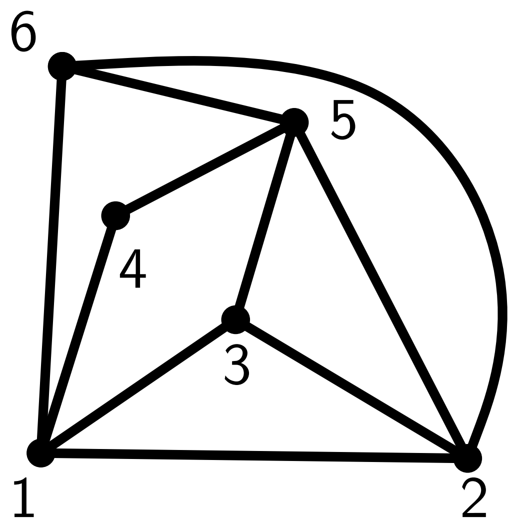



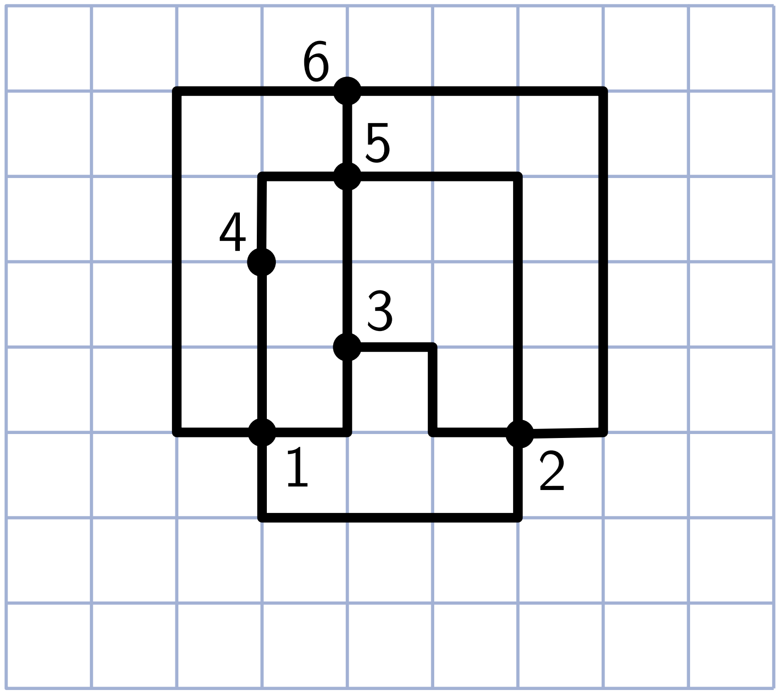





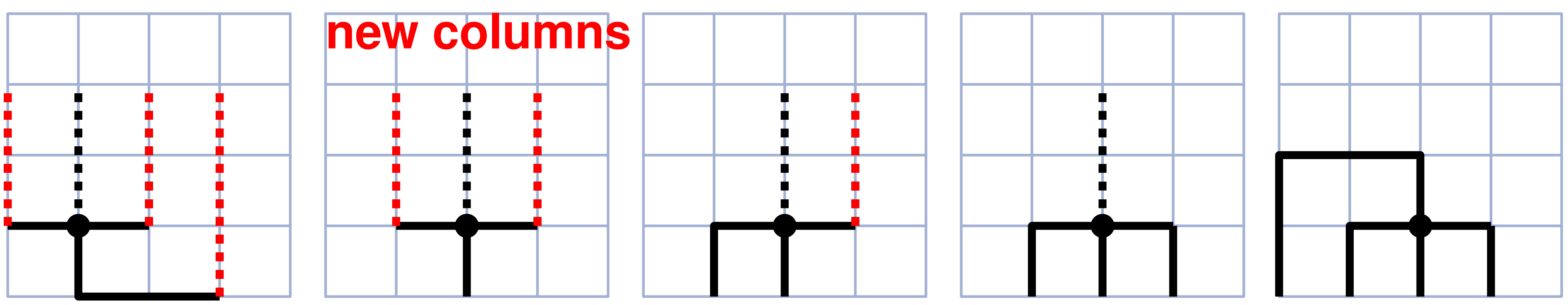

Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

# Biedl & Kant Orthogonal Drawing Algorithm




first vertex | indegree = 1 | indegree = 2 | indegree = 3 | indegree = 4



# Biedl & Kant Orthogonal Drawing Algorithm

**Lemma (Area of Biedl & Kant drawing)**

The width is $m - n + 1$ and the height at most $n + 1$.

# Biedl & Kant Orthogonal Drawing Algorithm

**Lemma (Area of Biedl & Kant drawing)**

The width is $m - n + 1$ and the height at most $n + 1$.

## Proof

Width: At each step we increase the number of columns by $outdeg(v_i) - 1$, if $i > 1$ and $outdeg(v_1)$ for $v_1$.

# Biedl & Kant Orthogonal Drawing Algorithm

**Lemma (Area of Biedl & Kant drawing)**

The width is $m - n + 1$ and the height at most $n + 1$.

## Proof

Width: At each step we increase the number of columns by $outdeg(v_i) - 1$, if $i > 1$ and $outdeg(v_1)$ for $v_1$.

Height: Vertices $v_1$ and $v_2$ use two rows, $v_i$, $i = 1, \ldots, n-1$ is placed in a new row. Vertex $v_n$ uses one more row if $indeg(v_n) = 4$.

# Biedl & Kant Orthogonal Drawing Algorithm

**Lemma (Area of Biedl & Kant drawing)**

The width is $m - n + 1$ and the height at most $n + 1$.

### Proof

Width: At each step we increase the number of columns by $outdeg(v_i) - 1$, if $i > 1$ and $outdeg(v_1)$ for $v_1$.

Height: Vertices $v_1$ and $v_2$ use two rows, $v_i$, $i = 1, \ldots, n - 1$ is placed in a new row. Vertex $v_n$ uses one more row if $indeg(v_n) = 4$.

**Lemma (Number of bends in Biedl & Kant drawing)**

There are at most $2m - 2n + 4$ bends.

# Biedl & Kant Orthogonal Drawing Algorithm

**Lemma (Area of Biedl & Kant drawing)**

The width is $m - n + 1$ and the height at most $n + 1$.

## Proof

Width: At each step we increase the number of columns by $outdeg(v_i) - 1$, if $i > 1$ and $outdeg(v_1)$ for $v_1$.

Height: Vertices $v_1$ and $v_2$ use two rows, $v_i$, $i = 1, \ldots, n-1$ is placed in a new row. Vertex $v_n$ uses one more row if $indeg(v_n) = 4$.

**Lemma (Number of bends in Biedl & Kant drawing)**

There are at most $2m - 2n + 4$ bends.

## Proof

Each vertex $v_i$, $i \neq 1, n$, introduces $indeg(v_i) - 1$ and $outdeg(v_i) - 1$ new bends.

# Biedl & Kant Orthogonal Drawing Algorithm

**Lemma (Number of bends per edge in Biedl & Kant drawing)**

All edges but one bent at most twice. The exceptional edge bents at most three times.

# Biedl & Kant Orthogonal Drawing Algorithm

**Lemma (Number of bends per edge in Biedl & Kant drawing)**

All edges but one bent at most twice. The exceptional edge bents at most three times.

## Proof

Let $(v_i, v_j)$, $i < j,\ i, j \neq 1, n$. Then $outdeg(v_i),\ indeg(v_j) \leq 3$. I.e $(v_i, v_j)$ gets at most one bend after placement of $v_i$ and at most one before placement of $v_j$. Edges outgoing from $v_1$ can me made 2-bend by using the column below $v_1$ for the edge $(v_1, v_2)$.

# Biedl & Kant Orthogonal Drawing Algorithm

**Lemma (Number of bends per edge in Biedl & Kant drawing)**

All edges but one bent at most twice. The exceptional edge bents at most three times.

**Proof**

Let $(v_i, v_j)$, $i < j,\ i, j \neq 1, n$. Then $outdeg(v_i),\ indeg(v_j) \leq 3$. I.e $(v_i, v_j)$ gets at most one bend after placement of $v_i$ and at most one before placement of $v_j$. Edges outgoing from $v_1$ can me made 2-bend by using the column below $v_1$ for the edge $(v_1, v_2)$.

**Lemma (planarity)**

For planar embedded graphs, with $v_1$ and $v_n$ on the outer face, the algorithm produces a planar drawing.

# Biedl & Kant Orthogonal Drawing Algorithm

**Lemma (Number of bends per edge in Biedl & Kant drawing)**

All edges but one bent at most twice. The exceptional edge bents at most three times.

**Proof**

Let $(v_i, v_j)$, $i < j$, $i, j \neq 1, n$. Then $outdeg(v_i)$, $indeg(v_j) \leq 3$. I.e $(v_i, v_j)$ gets at most one bend after placement of $v_i$ and at most one before placement of $v_j$. Edges outgoing from $v_1$ can me made 2-bend by using the column below $v_1$ for the edge $(v_1, v_2)$.

**Lemma (planarity)**

For planar embedded graphs, with $v_1$ and $v_n$ on the outer face, the algorithm produces a planar drawing.

**Proof**

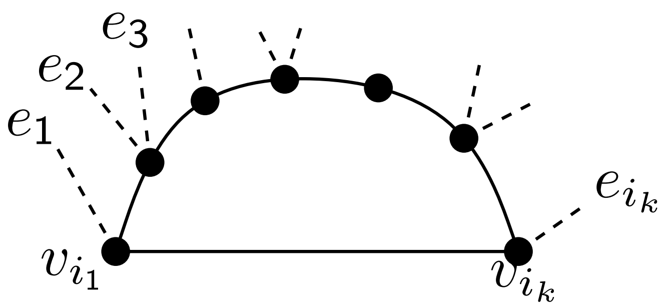
Consider a planar embedding of $G$. Let $v_1, \ldots, v_n$ be an $st$-ordering of $G$. Let $G_i$ be the graph induced by $v_1, \ldots, v_i$. It holds that

if $G$ is planar, vertex $v_{i+1}$ lies on the outer face of $G_i$

# Biedl & Kant Orthogonal Drawing Algorithm

**Proof** (Continuation)

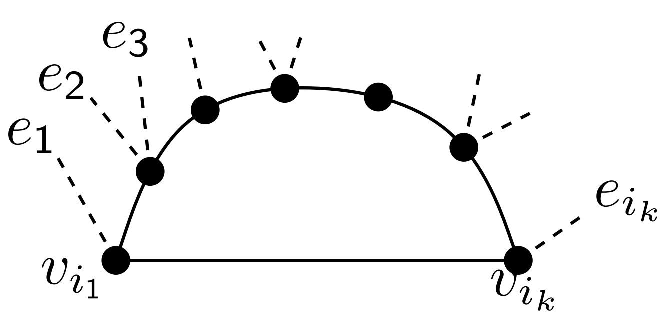
- The proof is by induction on $G_i$, $i = 1, \ldots, n$, with $G_n = G$.
- Let $E_i$ be the edges outgoing from the vertices of $G_i$ in the order they appear in the embedded $G$.
- We use as an **invariant** that edges $E_i$ appear in the same order in the orthogonal drawing of $G_i$.

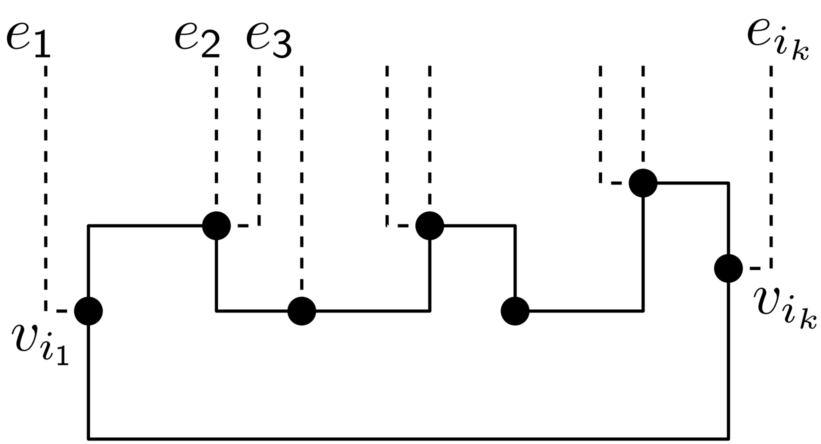


# Biedl & Kant Orthogonal Drawing Algorithm

**Proof** (Continuation)

- The proof is by induction on $G_i$, $i = 1, \dots, n$, with $G_n = G$.
- Let $E_i$ be the edges outgoing from the vertices of $G_i$ in the order they appear in the embedded $G$.
- We use as an **invariant** that edges $E_i$ appear in the same order in the orthogonal drawing of $G_i$.

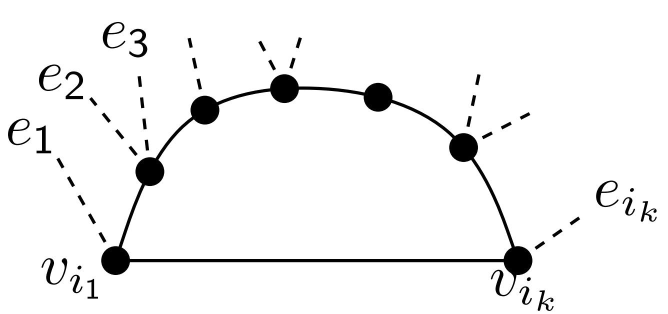


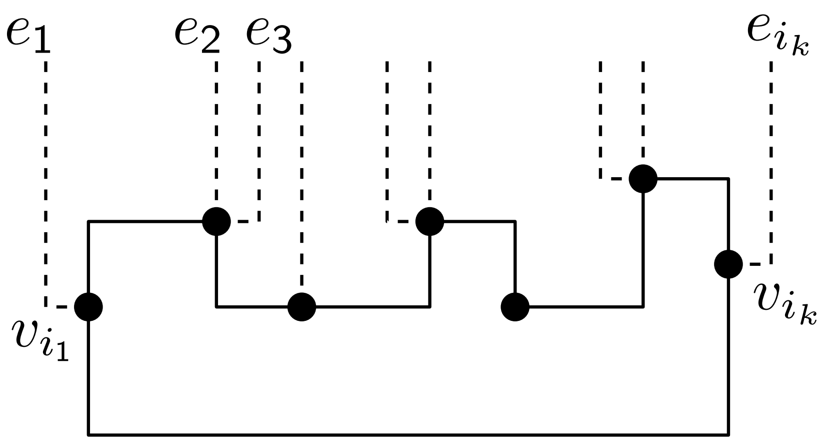



# Biedl & Kant Orthogonal Drawing Algorithm

**Proof** (Continuation)

- The proof is by induction on $G_i$, $i = 1, \ldots, n$, with $G_n = G$.
- Let $E_i$ be the edges outgoing from the vertices of $G_i$ in the order they appear in the embedded $G$.
- We use as an **invariant** that edges $E_i$ appear in the same order in the orthogonal drawing of $G_i$.
- Since $v_{i+1}$ is on the outer face of $G_i$, it can be placed without creating any crossing.

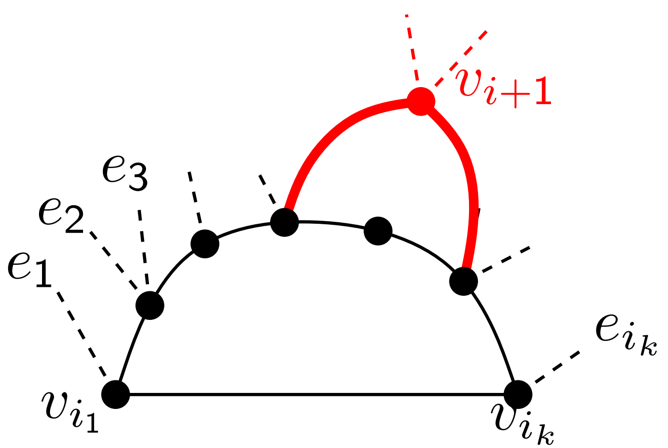


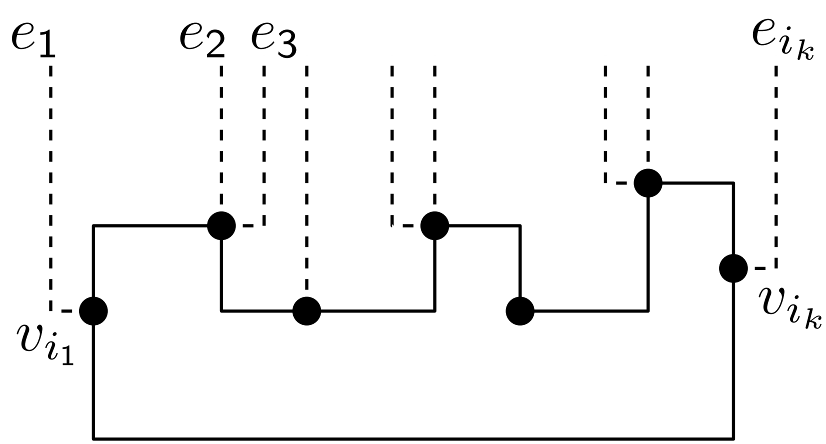



# Biedl & Kant Orthogonal Drawing Algorithm

**Proof** (Continuation)

- The proof is by induction on $G_i$, $i = 1, \ldots, n$, with $G_n = G$.
- Let $E_i$ be the edges outgoing from the vertices of $G_i$ in the order they appear in the embedded $G$.
- We use as an **invariant** that edges $E_i$ appear in the same order in the orthogonal drawing of $G_i$.
- Since $v_{i+1}$ is on the outer face of $G_i$, it can be placed without creating any crossing.

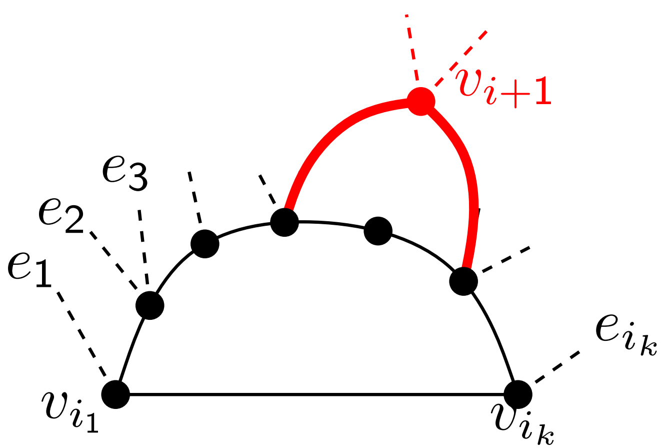


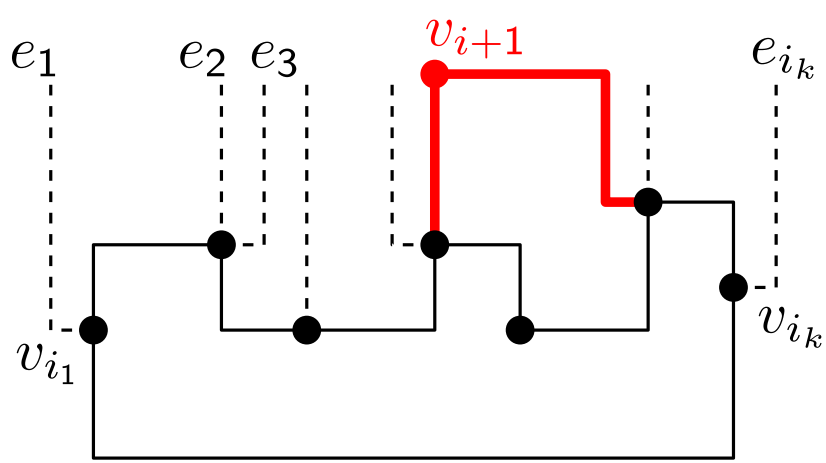



# Biedl & Kant Orthogonal Drawing Algorithm

**Proof** (Continuation)

- The proof is by induction on $G_i$, $i = 1, \ldots, n$, with $G_n = G$.
- Let $E_i$ be the edges outgoing from the vertices of $G_i$ in the order they appear in the embedded $G$.
- We use as an **invariant** that edges $E_i$ appear in the same order in the orthogonal drawing of $G_i$.
- Since $v_{i+1}$ is on the outer face of $G_i$, it can be placed without creating any crossing.

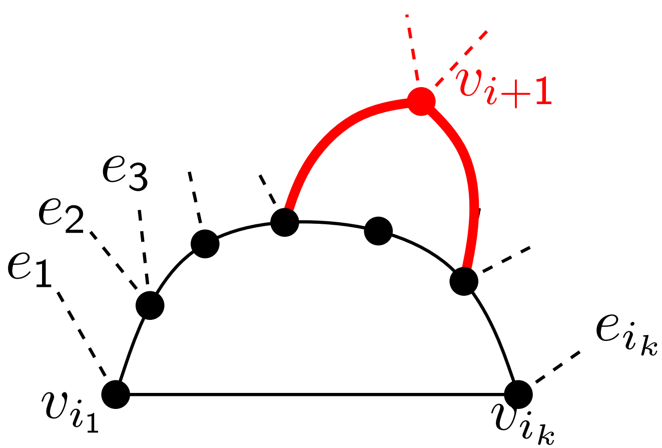


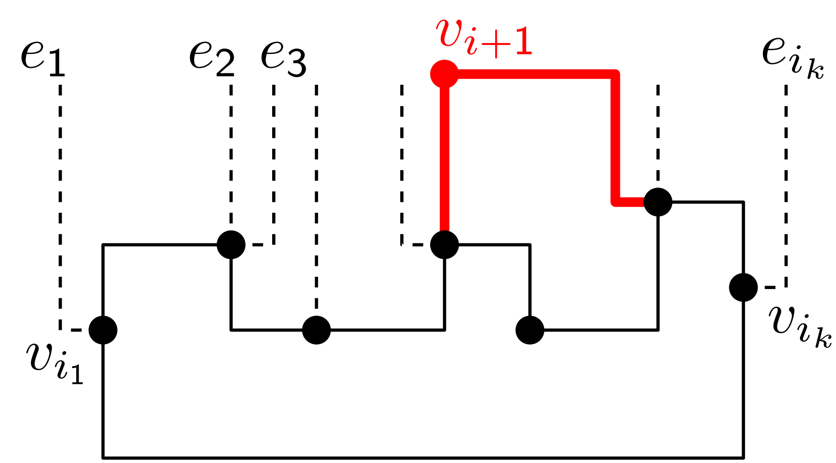



# Biedl & Kant Orthogonal Drawing Algorithm

**Proof** (Continuation)

- The proof is by induction on $G_i$, $i = 1, \ldots, n$, with $G_n = G$.
- Let $E_i$ be the edges outgoing from the vertices of $G_i$ in the order they appear in the embedded $G$.
- We use as an **invariant** that edges $E_i$ appear in the same order in the orthogonal drawing of $G_i$.
- Since $v_{i+1}$ is on the outer face of $G_i$, it can be placed without creating any crossing.
- The invariant holds after the induction step.

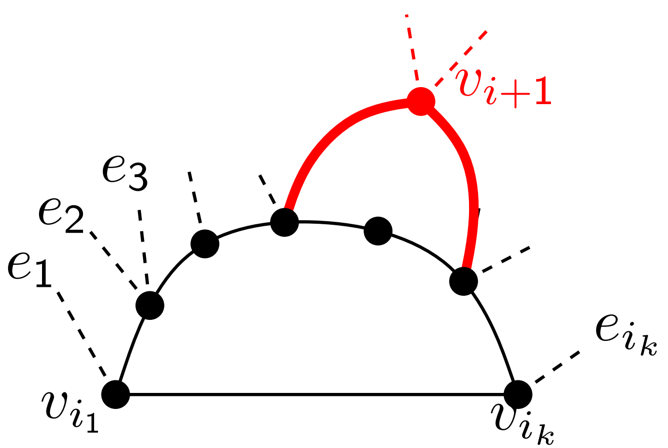


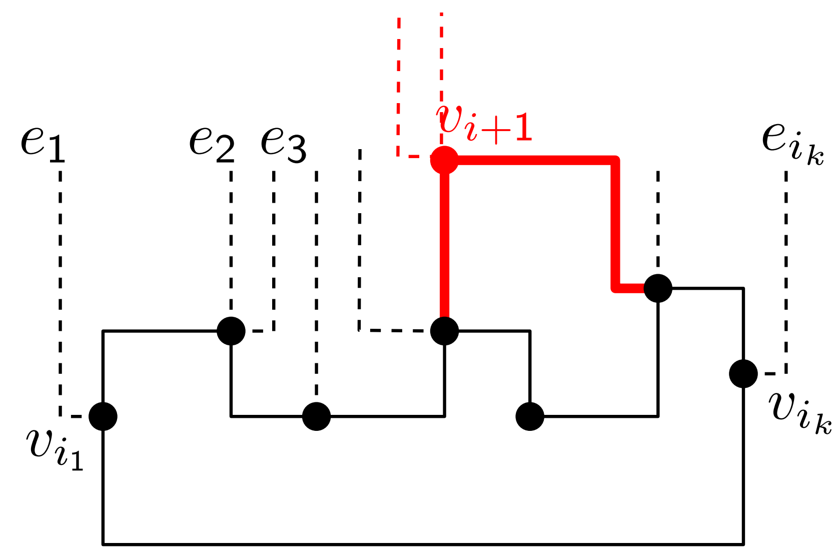



# Biedl & Kant Orthogonal Drawing Algorithm

**Proof** (Continuation)

- The proof is by induction on $G_i$, $i = 1, \ldots, n$, with $G_n = G$.
- Let $E_i$ be the edges outgoing from the vertices of $G_i$ in the order they appear in the embedded $G$.
- We use as an **invariant** that edges $E_i$ appear in the same order in the orthogonal drawing of $G_i$.
- Since $v_{i+1}$ is on the outer face of $G_i$, it can be placed without creating any crossing.
- The invariant holds after the induction step.





Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

# Biedl & Kant Orthogonal Drawing Algorithm

**Theorem (Biedl & Kant 98)**

A biconnected graph $G$ with vertex-degree at most 4 admits an orthogonal drawing such that:

- Area is $(m - n + 1) \times n + 1$
- Each edge (except maybe for one) has at most 2 bends
- The exceptional edge has at most 3 bends
- The total number if bends is at most $2m - 2n + 4$
- If $G$ is plane, the orthogonal drawing is planar
- Finally, provided an $st$-ordering such a drawing can be constructed in $O(n)$ time.

# Biedl & Kant Orthogonal Drawing Algorithm

**Theorem (Biedl & Kant 98)**

A biconnected graph $G$ with vertex-degree at most 4 admits an orthogonal drawing such that:

- Area is $(m-n+1) \times n+1$
- Each edge (except maybe for one) has at most 2 bends
- The exceptional edge has at most 3 bends
- The total number if bends is at most $2m-2n+4$
- If $G$ is plane, the orthogonal drawing is planar
- Finally, provided an $st$-ordering such a drawing can be constructed in $O(n)$ time.

- For the construction we have used an $st$-ordering of $G$!

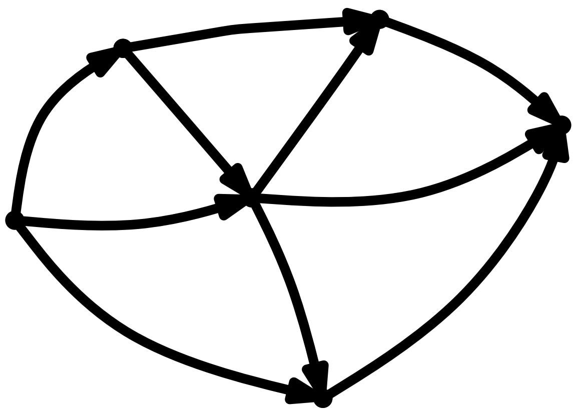


Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

# st-digraph, topological ordering

## Definition: st-digraph

Let $G$ be a directed graph. A vertex $s$ (resp. $t$) is called **source** (resp. **sink**) of $G$ if it has only outgoing (resp. incomming edges). A directed acyclic graph with one source and one sink is called ***st*-digraph**.



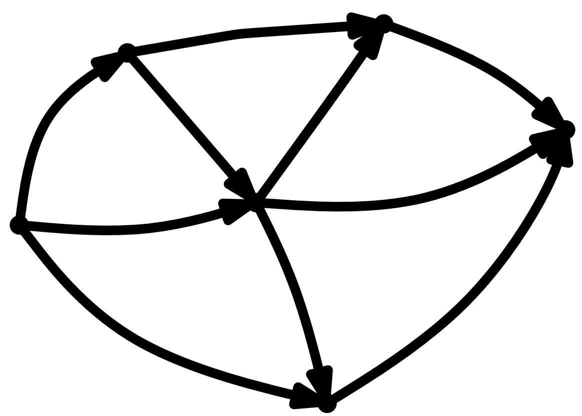
# st-digraph, topological ordering

## Definition: st-digraph

Let $G$ be a directed graph. A vertex $s$ (resp. $t$) is called **source** (resp. **sink**) of $G$ if it has only outgoing (resp. incomming edges). A directed acyclic graph with one source and one sink is called **$st$-digraph**.

## Definition: topological ordering

A **topological ordering** of a directed graph $G$ (with $n$ vertices) is an assignment of numbers $\{1, \ldots, n\}$ to the vertices of $G$, such that for every edge $(u, v)$, number$(v) >$ number$(u)$.



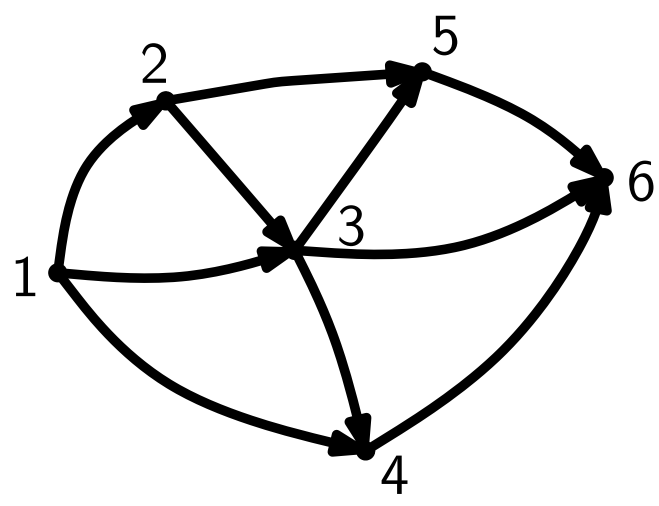
# st-digraph, topological ordering

## Definition: st-digraph

Let $G$ be a directed graph. A vertex $s$ (resp. $t$) is called **source** (resp. **sink**) of $G$ if it has only outgoing (resp. incomming edges). A directed acyclic graph with one source and one sink is called **$st$-digraph**.

## Definition: topological ordering

A **topological ordering** of a directed graph $G$ (with $n$ vertices) is an assignment of numbers $\{1, \ldots, n\}$ to the vertices of $G$, such that for every edge $(u, v)$, number$(v) >$ number$(u)$.



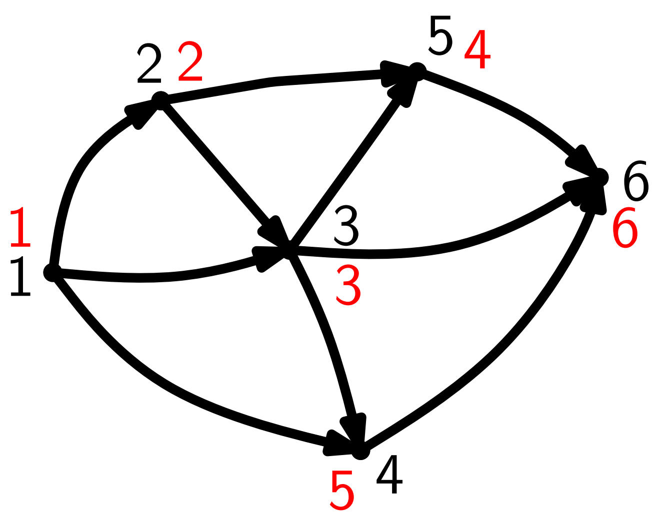
# st-digraph, topological ordering

## Definition: st-digraph

Let $G$ be a directed graph. A vertex $s$ (resp. $t$) is called **source** (resp. **sink**) of $G$ if it has only outgoing (resp. incomming edges). A directed acyclic graph with one source and one sink is called **$st$-digraph**.

## Definition: topological ordering

A **topological ordering** of a directed graph $G$ (with $n$ vertices) is an assignment of numbers $\{1, \ldots, n\}$ to the vertices of $G$, such that for every edge $(u, v)$, number$(v) >$ number$(u)$.



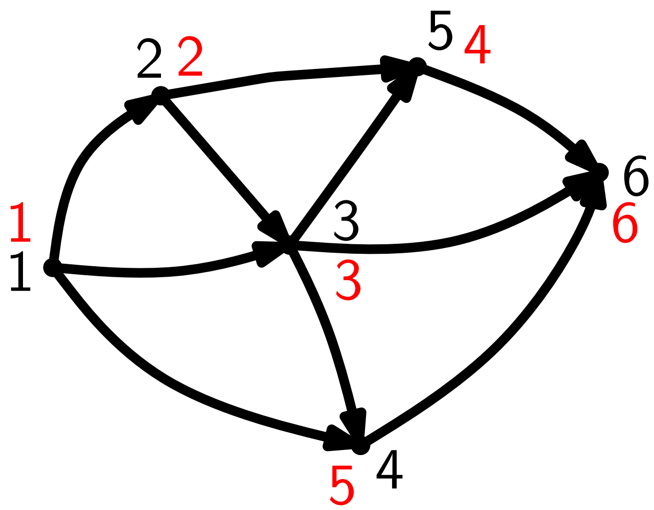
# st-digraph, topological ordering

## Definition: st-digraph

Let $G$ be a directed graph. A vertex $s$ (resp. $t$) is called **source** (resp. **sink**) of $G$ if it has only outgoing (resp. incomming edges). A directed acyclic graph with one source and one sink is called $st$**-digraph**.

## Definition: topological ordering

A **topological ordering** of a directed graph $G$ (with $n$ vertices) is an assignment of numbers $\{1, \ldots, n\}$ to the vertices of $G$, such that for every edge $(u, v)$, number$(v) >$ number$(u)$.




How to construct a topological ordering?

# $st$-ordering

## Construction of an $st$-ordering:

**$G$ is undirected biconnected graph**


Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

# $st$-ordering

## Construction of an $st$-ordering:

**$G$ is undirected biconnected graph** → Orient edges of $G$

# $st$-ordering

## Construction of an $st$-ordering:

**$G$ is undirected biconnected graph**

→ Orient edges of $G$

**$G'$ is an $st$-digraph**

# $st$-ordering

## Construction of an $st$-ordering:

**$G$ is undirected biconnected graph** → (Orient edges of $G$) → **$G'$ is an $st$-digraph** → **Let $v_1, \ldots, v_n$ be a topological ordering of $G'$**



Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

# $st$-ordering

## Construction of an $st$-ordering:

**$G$ is undirected biconnected graph**

→ Orient edges of $G$ →

**$G'$ is an $st$-digraph**

→

**Let $v_1, \ldots, v_n$ be a topological ordering of $G'$**

↓ Since $G'$ is an $st$-digraph, for $v_i$ $(i \neq 1, n)$ $\exists$ $(v_j, v_i)$ and $(v_i, v_k)$. By the property of topological ordering $j < i$ and $i < k$.

# $st$-ordering

## Construction of an $st$-ordering:

**$G$ is undirected biconnected graph**

→ Orient edges of $G$ →

**$G'$ is an $st$-digraph**

→

**Let $v_1, \ldots, v_n$ be a topological ordering of $G'$**

↓ Since $G'$ is an $st$-digraph, for $v_i$ $(i \neq 1, n)$ $\exists$ $(v_j, v_i)$ and $(v_i, v_k)$. By the property of topological ordering $j < i$ and $i < k$.

**$v_1, \ldots, v_n$ is an $st$-ordering of $G$**

# $st$-ordering

## Construction of an $st$-ordering:

**$G$ is undirected biconnected graph** → (Orient edges of $G$) → **$G'$ is an $st$-digraph** → **Let $v_1, \ldots, v_n$ be a topological ordering of $G'$**

↓ Since $G'$ is an $st$-digraph, for $v_i$ $(i \neq 1, n)$ $\exists$ $(v_j, v_i)$ and $(v_i, v_k)$. By the property of topological ordering $j < i$ and $i < k$.

**$v_1, \ldots, v_n$ is an $st$-ordering of $G$**

**EXAMPLE**

# $st$-ordering

## Construction of an $st$-ordering:

**$G$ is undirected biconnected graph** → (**HOW?** Orient edges of $G$) → **$G'$ is an $st$-digraph** → **Let $v_1, \ldots, v_n$ be a topological ordering of $G'$**

↓ Since $G'$ is an $st$-digraph, for $v_i$ $(i \neq 1, n)$ $\exists$ $(v_j, v_i)$ and $(v_i, v_k)$. By the property of topological ordering $j < i$ and $i < k$.

**$v_1, \ldots, v_n$ is an $st$-ordering of $G$**

**EXAMPLE**

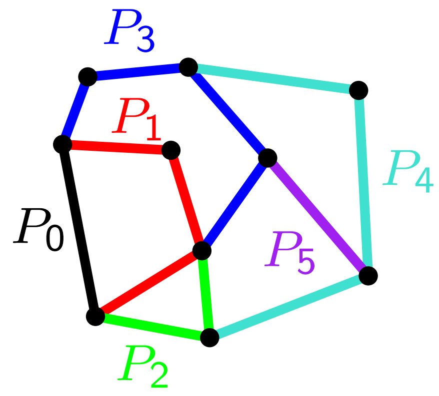
# $st$-ordering

## Definition: Ear decomposition

An ear decomposition $D = (P_0, \dots, P_r)$ of an undirected graph $G = (V, E)$ is a partition of $E$ into an ordered collection of edge disjoint paths $P_0, \dots, P_r$, such that:

- $P_0$ is an edge
- $P_0 \cup P_1$ is a simple cycle
- both end-vertices of $P_i$ belong to $P_0 \cup \dots \cup P_{i-1}$
- no internal vertex of $P_i$ belong to $P_0 \cup \dots \cup P_{i-1}$

An ear decomposition of open if $P_0, \dots, P_r$ are simple paths.



# $st$-ordering

**Lemma (Ear decomposition)**

Let $G = (V, E)$ be a biconnected graph $G$ and let $(s, t) \in E$. $G$ has an open ear decomposition $(P_0, \ldots, P_r)$, where $P_0 = (s, t)$.

# $st$-ordering

**Lemma (Ear decomposition)**

Let $G = (V, E)$ be a biconnected graph $G$ and let $(s, t) \in E$. $G$ has an open ear decomposition $(P_0, \ldots, P_r)$, where $P_0 = (s, t)$.

## Proof

- Let $P_0 = (s, t)$ and $P_1$ be path between $s$ and $t$, it exists since $G$ is biconnected.



Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

# $st$-ordering

**Lemma (Ear decomposition)**

Let $G = (V, E)$ be a biconnected graph $G$ and let $(s, t) \in E$. $G$ has an open ear decomposition $(P_0, \ldots, P_r)$, where $P_0 = (s, t)$.

## Proof

- Let $P_0 = (s, t)$ and $P_1$ be path between $s$ and $t$, it exists since $G$ is biconnected.
- Induction hypothesis: $P_0, \ldots, P_i$ are ears.

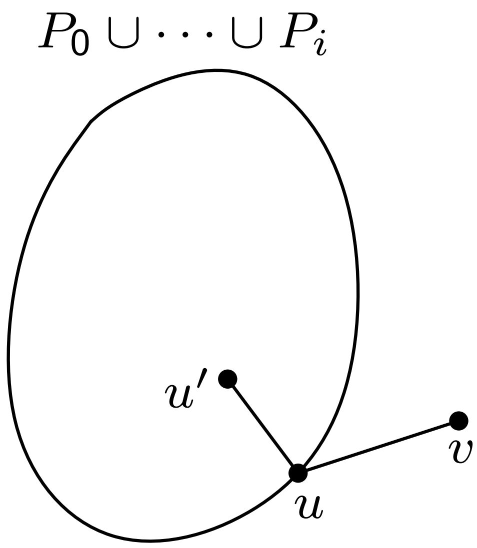
# $st$-ordering

**Lemma (Ear decomposition)**

Let $G = (V, E)$ be a biconnected graph $G$ and let $(s, t) \in E$. $G$ has an open ear decomposition $(P_0, \dots, P_r)$, where $P_0 = (s, t)$.

## Proof

- Let $P_0 = (s, t)$ and $P_1$ be path between $s$ and $t$, it exists since $G$ is biconnected.
- Induction hypothesis: $P_0, \dots, P_i$ are ears.
- Let $(u, v)$ be an edge in $G$ such that $u \in P_0 \cup \dots \cup P_i$ and $v \notin P_0 \cup \dots \cup P_i$. Let $(u, u')$, such that $u' \in P_0 \cup \dots \cup P_i$. Let $P$ be a path between $v$ and $u'$, not passing through $u$. $P$ exists since $G$ is biconnected.



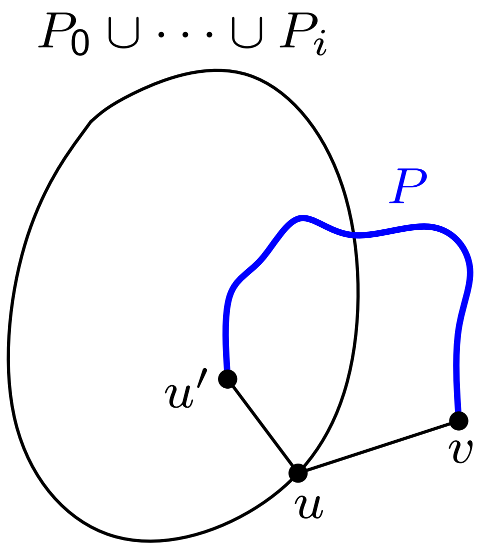
# $st$-ordering

**Lemma (Ear decomposition)**

Let $G = (V, E)$ be a biconnected graph $G$ and let $(s, t) \in E$. $G$ has an open ear decomposition $(P_0, \ldots, P_r)$, where $P_0 = (s, t)$.

## Proof

- Let $P_0 = (s, t)$ and $P_1$ be path between $s$ and $t$, it exists since $G$ is biconnected.
- Induction hypothesis: $P_0, \ldots, P_i$ are ears.
- Let $(u, v)$ be an edge in $G$ such that $u \in P_0 \cup \cdots \cup P_i$ and $v \notin P_0 \cup \cdots \cup P_i$. Let $(u, u')$, such that $u' \in P_0 \cup \cdots \cup P_i$. Let $P$ be a path between $v$ and $u'$, not passing through $u$. $P$ exists since $G$ is biconnected.



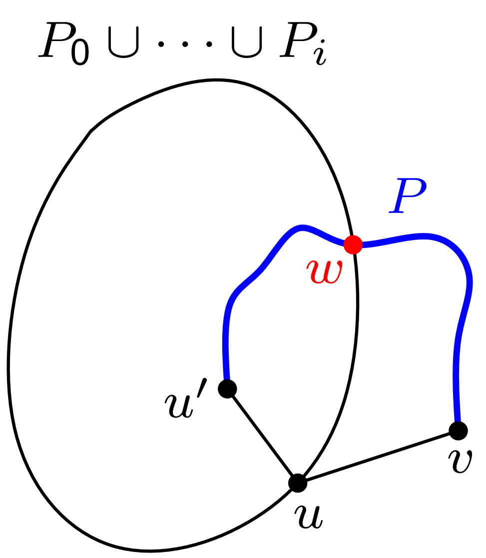
# $st$-ordering

**Lemma (Ear decomposition)**

Let $G = (V, E)$ be a biconnected graph $G$ and let $(s, t) \in E$. $G$ has an open ear decomposition $(P_0, \ldots, P_r)$, where $P_0 = (s, t)$.

## Proof

- Let $P_0 = (s, t)$ and $P_1$ be path between $s$ and $t$, it exists since $G$ is biconnected.
- Induction hypothesis: $P_0, \ldots, P_i$ are ears.
- Let $(u, v)$ be an edge in $G$ such that $u \in P_0 \cup \cdots \cup P_i$ and $v \notin P_0 \cup \cdots \cup P_i$. Let $(u, u')$, such that $u' \in P_0 \cup \cdots \cup P_i$. Let $P$ be a path between $v$ and $u'$, not passing through $u$. $P$ exists since $G$ is biconnected.
- Let $w$ be the first vertex of $P$ that is contained in $P_0 \cup \cdots \cup P_i$. Set $P_{i+1} = (u, v) \cup P(v - \cdots - w)$.



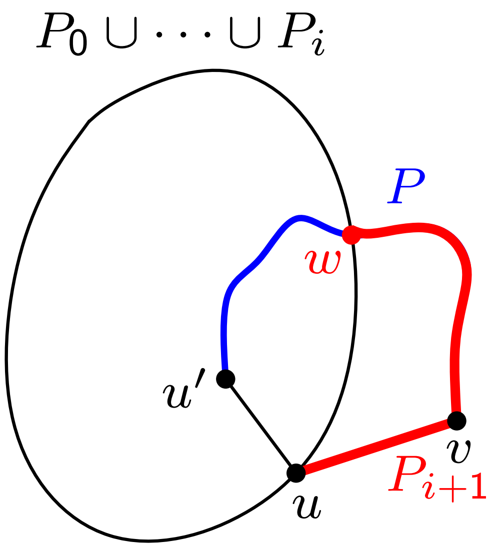
# $st$-ordering

**Lemma (Ear decomposition)**

Let $G = (V, E)$ be a biconnected graph $G$ and let $(s,t) \in E$. $G$ has an open ear decomposition $(P_0, \dots, P_r)$, where $P_0 = (s,t)$.

## Proof

- Let $P_0 = (s,t)$ and $P_1$ be path between $s$ and $t$, it exists since $G$ is biconnected.
- Induction hypothesis: $P_0, \dots, P_i$ are ears.
- Let $(u,v)$ be an edge in $G$ such that $u \in P_0 \cup \dots \cup P_i$ and $v \notin P_0 \cup \dots \cup P_i$. Let $(u,u')$, such that $u' \in P_0 \cup \dots \cup P_i$. Let $P$ be a path between $v$ and $u'$, not passing through $u$. $P$ exists since $G$ is biconnected.
- Let $w$ be the first vertex of $P$ that is contained in $P_0 \cup \dots \cup P_i$. Set $P_{i+1} = (u,v) \cup P(v - \dots - w)$.



# $st$-ordering

**Lemma ($st$-orientation)**

Let $G = (V, E)$ be a biconnected graph $G$ and let $(s, t) \in E$. There is an orientation $G'$ of $G$ which represents an $st$-digraph. $G'$ is called $st$-orientation of $G$.

# $st$-ordering

**Lemma ($st$-orientation)**

Let $G = (V, E)$ be a biconnected graph $G$ and let $(s, t) \in E$. There is an orientation $G'$ of $G$ which represents an $st$-digraph. $G'$ is called $st$-orientation of $G$.

## Proof

- Let $D = (P_0, \ldots, P_r)$ be an ear decomposition of $G = (V, E)$. Notice that $G = P_0 \cup \cdots \cup P_r$.

# $st$-ordering

**Lemma ($st$-orientation)**

Let $G = (V, E)$ be a biconnected graph $G$ and let $(s, t) \in E$. There is an orientation $G'$ of $G$ which represents an $st$-digraph. $G'$ is called $st$-orientation of $G$.

**Proof**

- Let $D = (P_0, \dots, P_r)$ be an ear decomposition of $G = (V, E)$. Notice that $G = P_0 \cup \dots \cup P_r$.
- Let $G_i = P_0 \cup \dots \cup P_i$. We prove that $G_i$ has an $st$-orientation by induction on $i$.

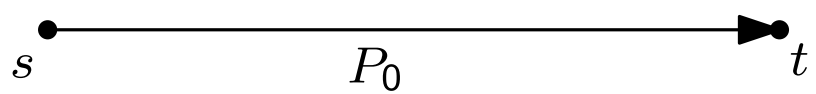
# $st$-ordering

**Lemma ($st$-orientation)**

Let $G = (V, E)$ be a biconnected graph $G$ and let $(s, t) \in E$. There is an orientation $G'$ of $G$ which represents an $st$-digraph. $G'$ is called $st$-orientation of $G$.

**Proof**

- Let $D = (P_0, \dots, P_r)$ be an ear decomposition of $G = (V, E)$. Notice that $G = P_0 \cup \dots \cup P_r$.
- Let $G_i = P_0 \cup \dots \cup P_i$. We prove that $G_i$ has an $st$-orientation by induction on $i$.



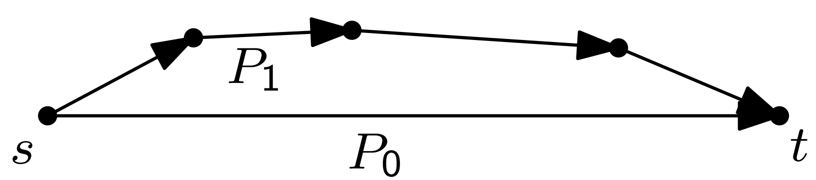
# $st$-ordering

**Lemma ($st$-orientation)**

Let $G = (V, E)$ be a biconnected graph $G$ and let $(s, t) \in E$. There is an orientation $G'$ of $G$ which represents an $st$-digraph. $G'$ is called $st$-orientation of $G$.

## Proof

- Let $D = (P_0, \ldots, P_r)$ be an ear decomposition of $G = (V, E)$. Notice that $G = P_0 \cup \cdots \cup P_r$.
- Let $G_i = P_0 \cup \cdots \cup P_i$. We prove that $G_i$ has an $st$-orientation by induction on $i$.



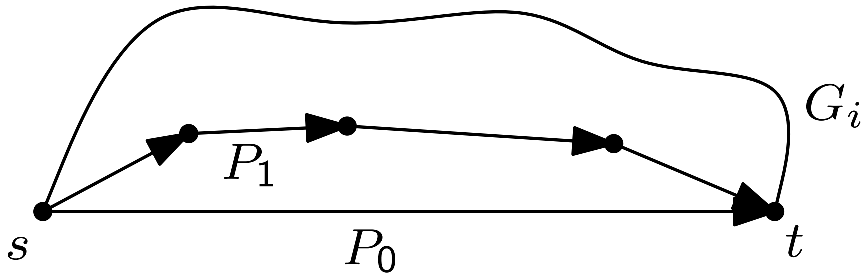
# $st$-ordering

**Lemma ($st$-orientation)**

Let $G = (V, E)$ be a biconnected graph $G$ and let $(s, t) \in E$. There is an orientation $G'$ of $G$ which represents an $st$-digraph. $G'$ is called $st$-orientation of $G$.

**Proof**

- Let $D = (P_0, \dots, P_r)$ be an ear decomposition of $G = (V, E)$. Notice that $G = P_0 \cup \dots \cup P_r$.
- Let $G_i = P_0 \cup \dots \cup P_i$. We prove that $G_i$ has an $st$-orientation by induction on $i$.





Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

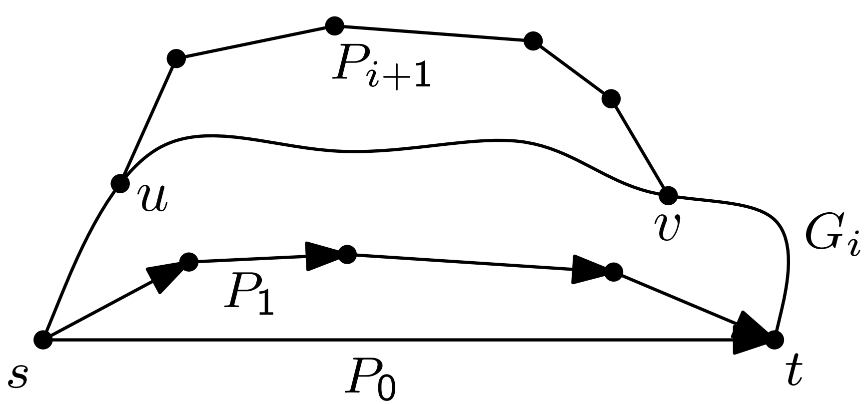
# $st$-ordering

**Lemma ($st$-orientation)**

Let $G = (V, E)$ be a biconnected graph $G$ and let $(s, t) \in E$. There is an orientation $G'$ of $G$ which represents an $st$-digraph. $G'$ is called $st$-orientation of $G$.

**Proof**

- Let $D = (P_0, \ldots, P_r)$ be an ear decomposition of $G = (V, E)$. Notice that $G = P_0 \cup \cdots \cup P_r$.
- Let $G_i = P_0 \cup \cdots \cup P_i$. We prove that $G_i$ has an $st$-orientation by induction on $i$.



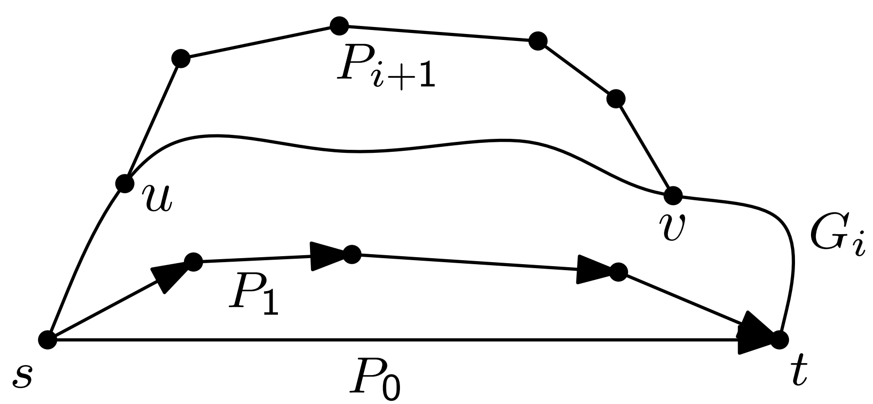
# $st$-ordering

**Lemma ($st$-orientation)**

Let $G = (V, E)$ be a biconnected graph $G$ and let $(s, t) \in E$. There is an orientation $G'$ of $G$ which represents an $st$-digraph. $G'$ is called $st$-orientation of $G$.

## Proof

- Let $D = (P_0, \dots, P_r)$ be an ear decomposition of $G = (V, E)$. Notice that $G = P_0 \cup \dots \cup P_r$.
- Let $G_i = P_0 \cup \dots \cup P_i$. We prove that $G_i$ has an $st$-orientation by induction on $i$.




- Distinguish two cases based on whether $u$ and $v$ are connected by a directed path or not.

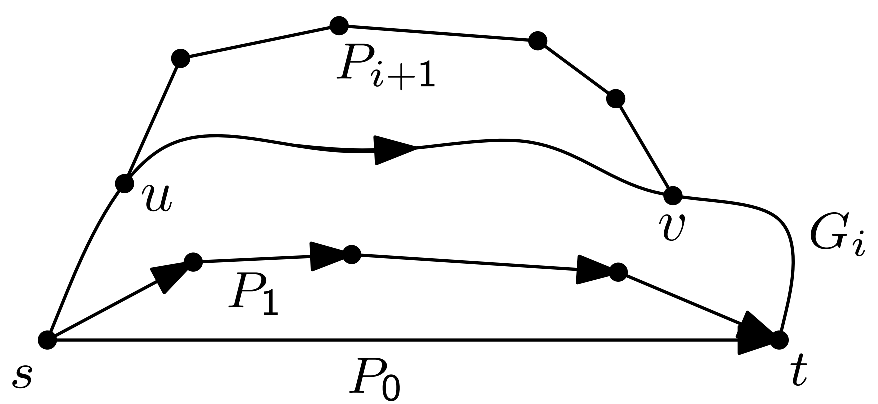
# $st$-ordering

**Lemma ($st$-orientation)**

Let $G = (V, E)$ be a biconnected graph $G$ and let $(s, t) \in E$. There is an orientation $G'$ of $G$ which represents an $st$-digraph. $G'$ is called $st$-orientation of $G$.

**Proof**

- Let $D = (P_0, \dots, P_r)$ be an ear decomposition of $G = (V, E)$. Notice that $G = P_0 \cup \dots \cup P_r$.
- Let $G_i = P_0 \cup \dots \cup P_i$. We prove that $G_i$ has an $st$-orientation by induction on $i$.




- Distinguish two cases based on whether $u$ and $v$ are connected by a directed path or not.

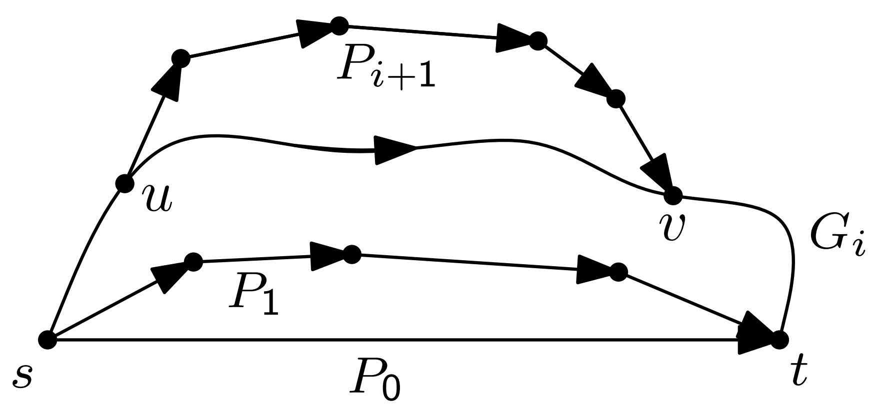
# $st$-ordering

**Lemma ($st$-orientation)**

Let $G = (V, E)$ be a biconnected graph $G$ and let $(s, t) \in E$. There is an orientation $G'$ of $G$ which represents an $st$-digraph. $G'$ is called $st$-orientation of $G$.

**Proof**

- Let $D = (P_0, \dots, P_r)$ be an ear decomposition of $G = (V, E)$. Notice that $G = P_0 \cup \dots \cup P_r$.
- Let $G_i = P_0 \cup \dots \cup P_i$. We prove that $G_i$ has an $st$-orientation by induction on $i$.




- Distinguish two cases based on whether $u$ and $v$ are connected by a directed path or not.

# $st$-ordering

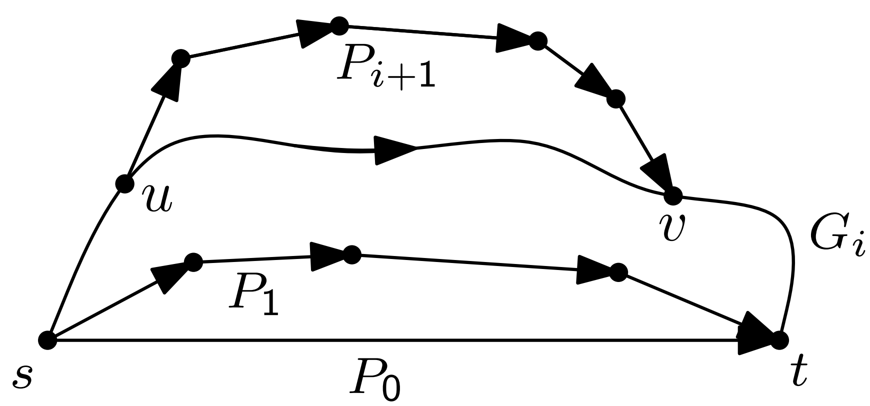
**Lemma ($st$-orientation)**

Let $G = (V, E)$ be a biconnected graph $G$ and let $(s, t) \in E$. There is an orientation $G'$ of $G$ which represents an $st$-digraph. $G'$ is called $st$-orientation of $G$.

**Proof**

- Let $D = (P_0, \dots, P_r)$ be an ear decomposition of $G = (V, E)$. Notice that $G = P_0 \cup \dots \cup P_r$.
- Let $G_i = P_0 \cup \dots \cup P_i$. We prove that $G_i$ has an $st$-orientation by induction on $i$.

EXAMPLE




- Distinguish two cases based on whether $u$ and $v$ are connected by a directed path or not.

## Construction of an $st$-ordering:

**$G$ is undirected biconnected graph**

→ **HOW?** Orient edges of $G$ →

**$G'$ is an $st$-digraph**

→

**Let $v_1, \dots, v_n$ be a topological ordering of $G'$**

↓ Since $G'$ is an $st$-digraph, for $v_i$ $(i \neq 1, n)$ $\exists$ $(v_j, v_i)$ and $(v_i, v_k)$. By the property of topological ordering $j < i$ and $i < k$.

**$v_1, \dots, v_n$ is an $st$-ordering of $G$**

## Construction of an $st$-ordering:

**$G$ is undirected biconnected graph**

**HOW?**

Orient edges of $G$

**$G'$ is an $st$-digraph**

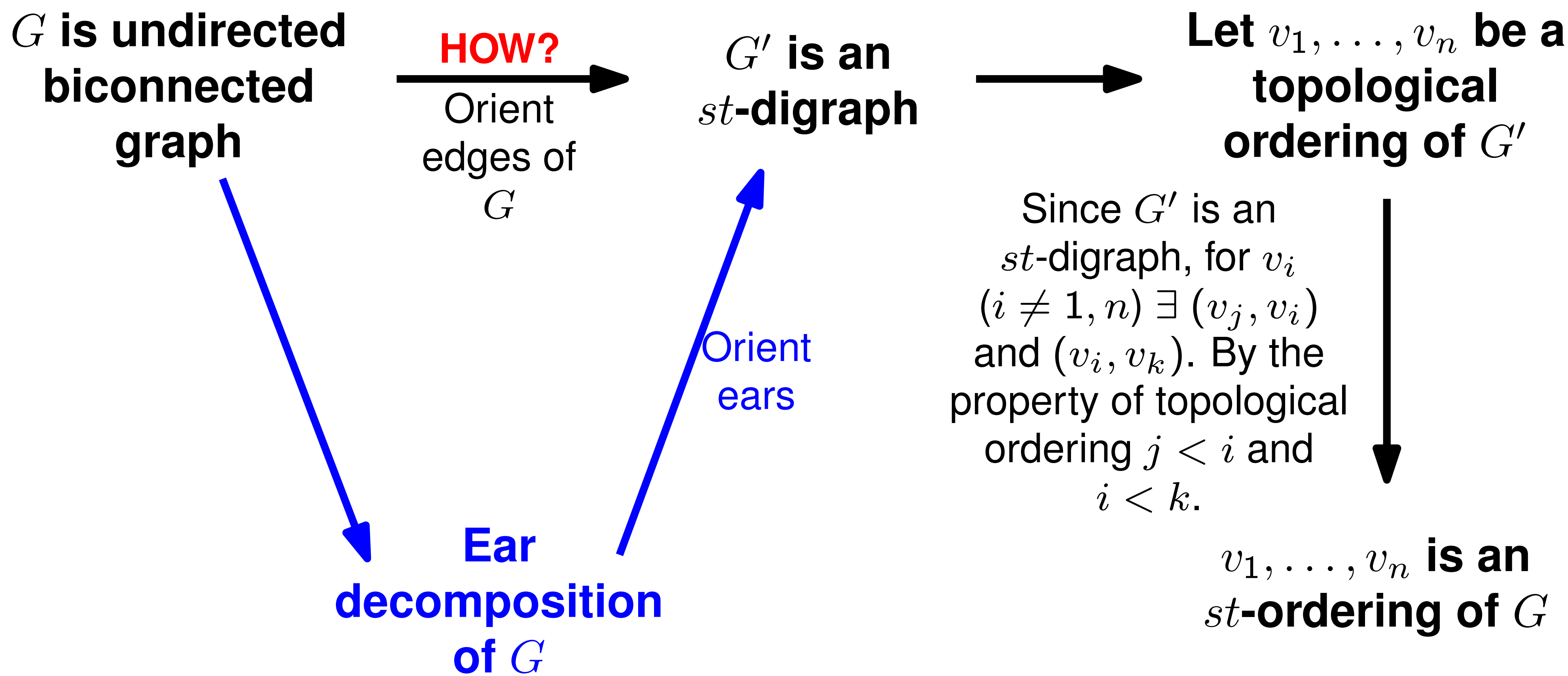
**Let $v_1, \ldots, v_n$ be a topological ordering of $G'$**

Since $G'$ is an $st$-digraph, for $v_i$ $(i \neq 1, n)$ $\exists$ $(v_j, v_i)$ and $(v_i, v_k)$. By the property of topological ordering $j < i$ and $i < k$.

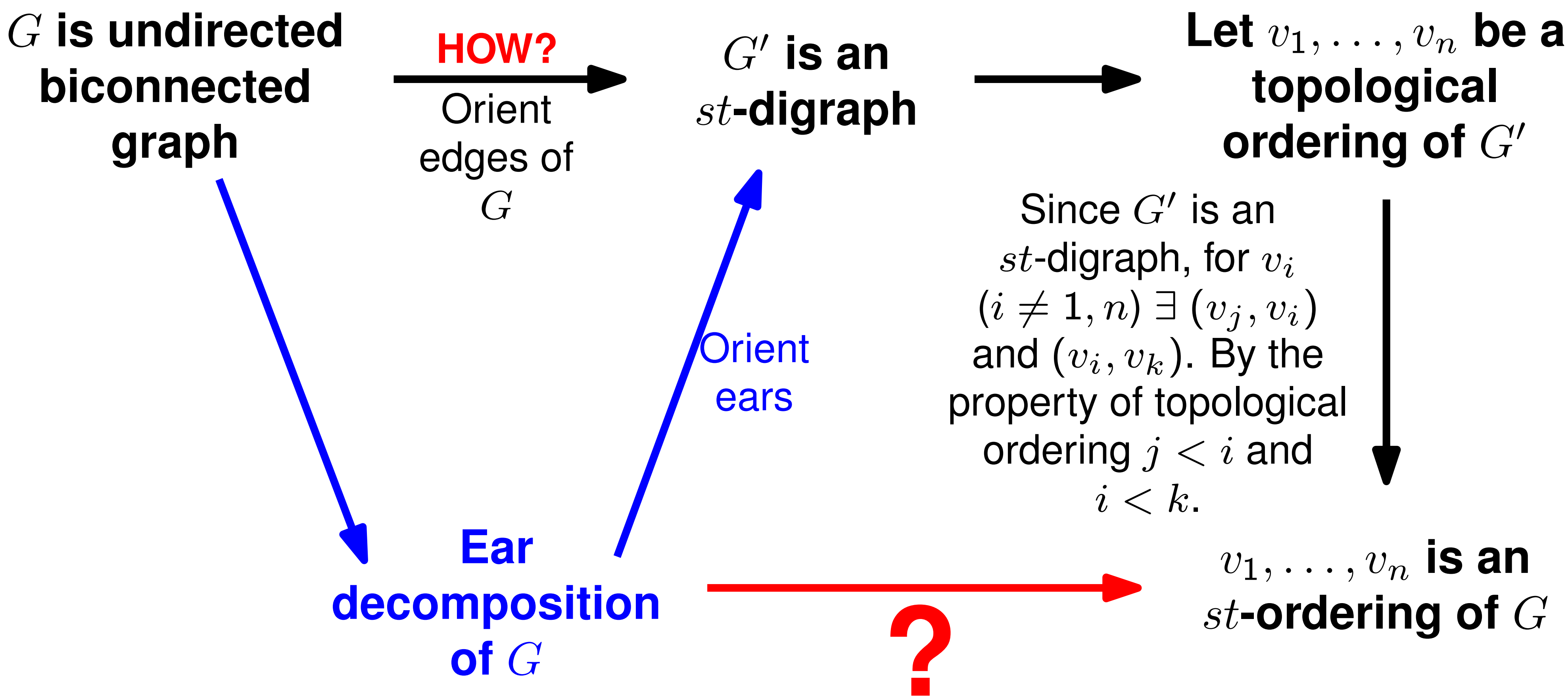
**$v_1, \ldots, v_n$ is an $st$-ordering of $G$**

**Ear decomposition of $G$**

## Construction of an $st$-ordering:



## Construction of an $st$-ordering:



# $st$-ordering

## Direct construction of $st$-ordering from ear decomposition

# $st$-ordering

## Direct construction of $st$-ordering from ear decomposition

- We construct it incrementally, considering $G_i = P_0 \cup \cdots \cup P_i$, $i = 0, \ldots, r$.



Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

# $st$-ordering

## Direct construction of $st$-ordering from ear decomposition

- We construct it incrementally, considering $G_i = P_0 \cup \cdots \cup P_i$, $i = 0, \ldots, r$.
- For $G_1$, let $P_1 = \{u_1, \ldots, u_p\}$, here $u_1 = s$ and $u_p = t$. The sequence $L = \{u_1, \ldots, u_p\}$ is an $st$-ordering of $G_1$.


Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

# $st$-ordering

## Direct construction of $st$-ordering from ear decomposition

- We construct it incrementally, considering $G_i = P_0 \cup \dots \cup P_i$, $i = 0, \dots, r$.
- For $G_1$, let $P_1 = \{u_1, \dots, u_p\}$, here $u_1 = s$ and $u_p = t$. The sequence $L = \{u_1, \dots, u_p\}$ is an $st$-ordering of $G_1$.
- Assume that $L$ contains an $st$-ordering of $G_i$ and let ear $P_{i+1} = \{v_1, \dots, v_q\}$. We insert vertices $v_1, \dots, v_q$ to $L$ after vertex $v_1$ (or before $v_q$).

# $st$-ordering

## Direct construction of $st$-ordering from ear decomposition

- We construct it incrementally, considering $G_i = P_0 \cup \cdots \cup P_i$, $i = 0, \ldots, r$.
- For $G_1$, let $P_1 = \{u_1, \ldots, u_p\}$, here $u_1 = s$ and $u_p = t$. The sequence $L = \{u_1, \ldots, u_p\}$ is an $st$-ordering of $G_1$.
- Assume that $L$ contains an $st$-ordering of $G_i$ and let ear $P_{i+1} = \{v_1, \ldots, v_q\}$. We insert vertices $v_1, \ldots, v_q$ to $L$ after vertex $v_1$ (or before $v_q$).

EXAMPLE

# $st$-ordering

## Direct construction of $st$-ordering from ear decomposition

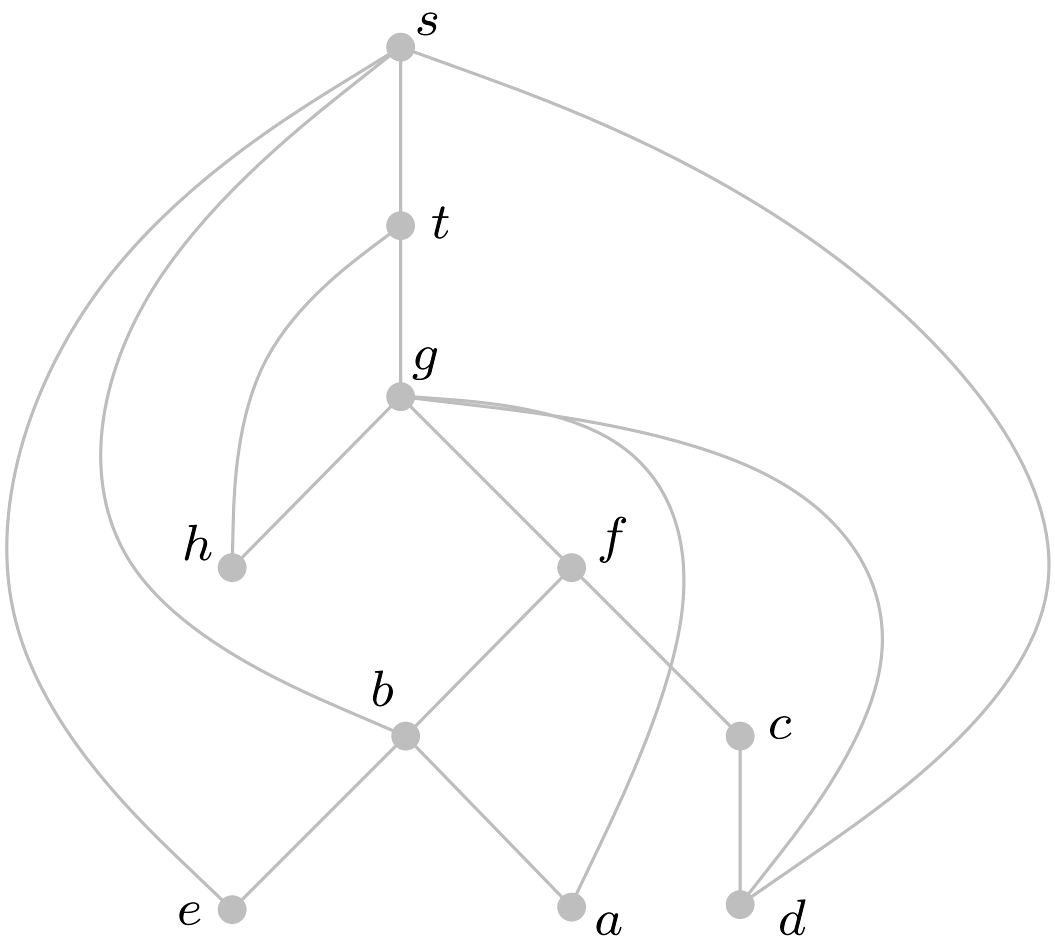
- We construct it incrementally, considering $G_i = P_0 \cup \cdots \cup P_i$, $i = 0, \ldots, r$.
- For $G_1$, let $P_1 = \{u_1, \ldots, u_p\}$, here $u_1 = s$ and $u_p = t$. The sequence $L = \{u_1, \ldots, u_p\}$ is an $st$-ordering of $G_1$.
- Assume that $L$ contains an $st$-ordering of $G_i$ and let ear $P_{i+1} = \{v_1, \ldots, v_q\}$. We insert vertices $v_1, \ldots, v_q$ to $L$ after vertex $v_1$ (or before $v_q$).
- **Why this is an $st$-ordering?** Let $G'_{i+1}$ be an $st$-orientation of $G_i$ as constructed in the previous proof. $L$ is a topological ordering of $G'_{i+1}$ and therefore an $st$-ordering of $G_i$

**EXAMPLE**

# $st$-ordering

## Algorithm: $st$-ordering (example)

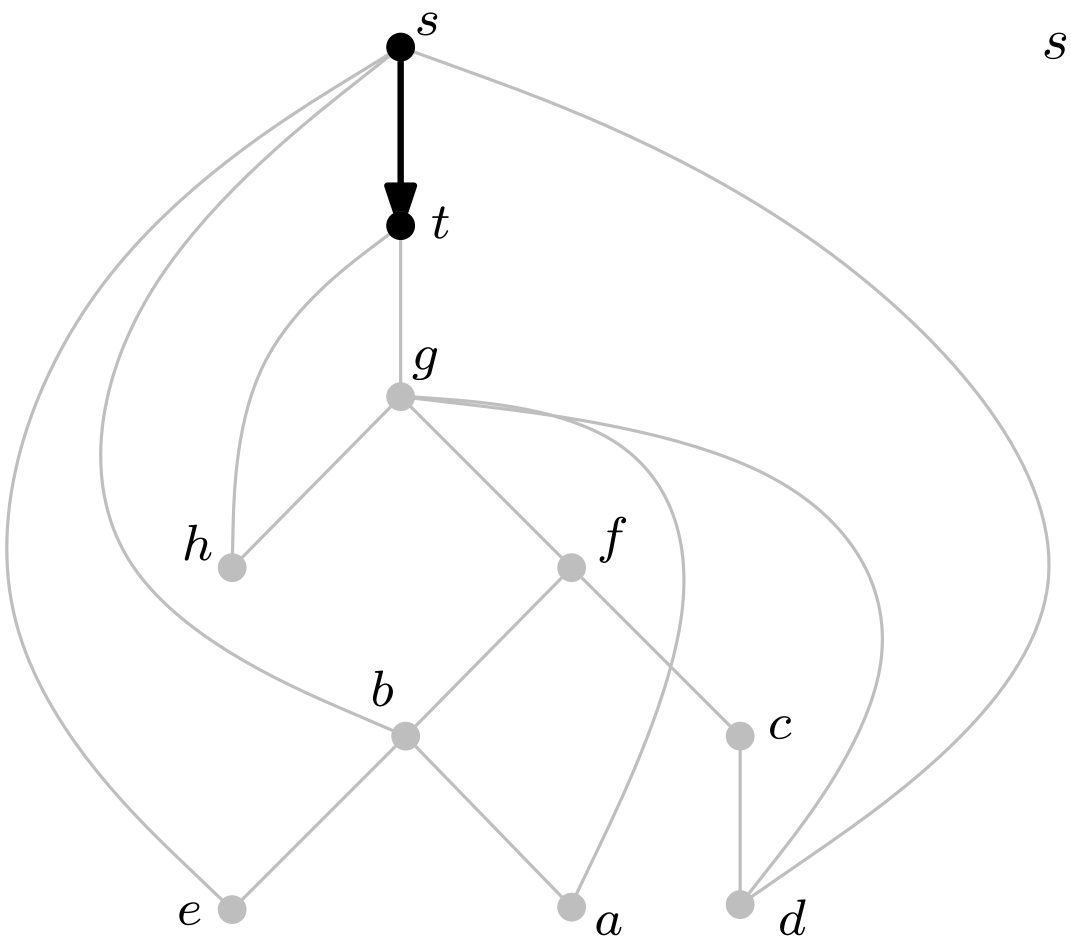
(Implementation details - Based on DFS)



# $st$-ordering

## Algorithm: $st$-ordering (example)

(Implementation details - Based on DFS)



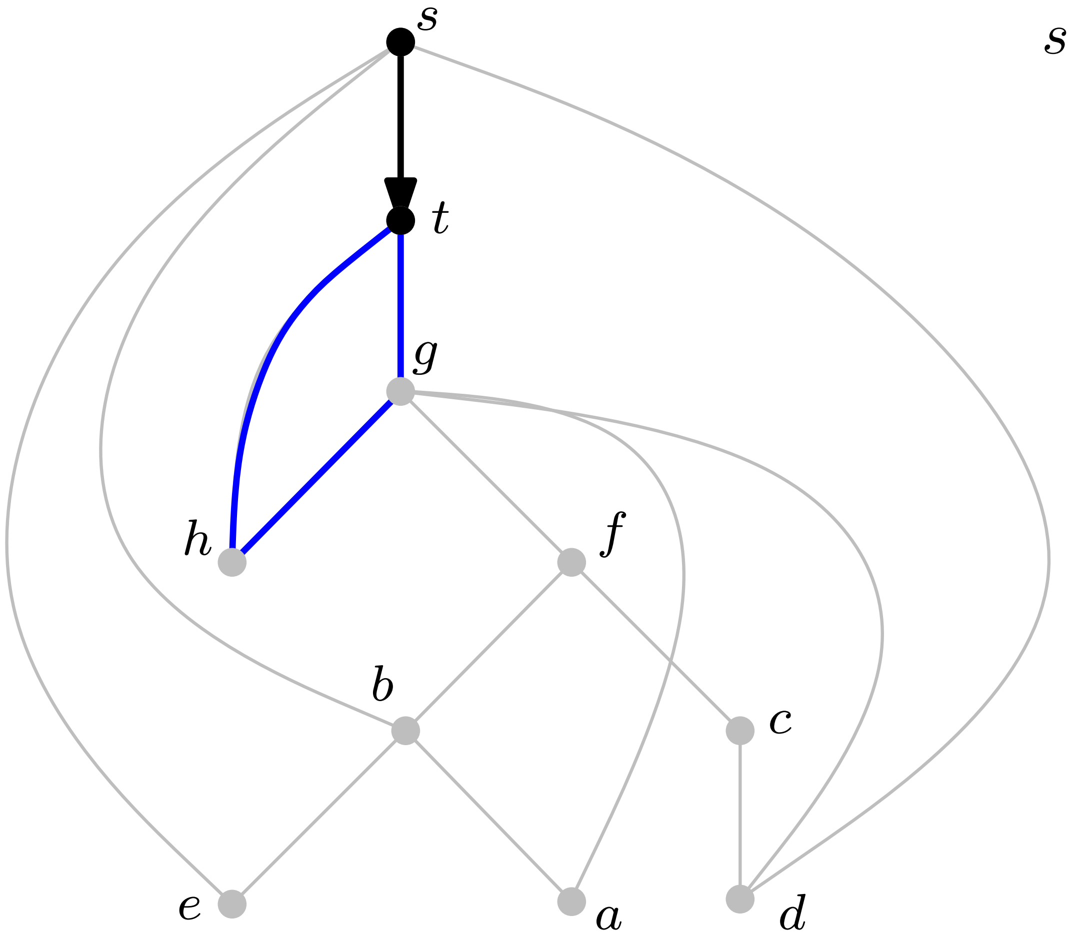

$s, t$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

# $st$-ordering

## Algorithm: $st$-ordering (example)
(Implementation details - Based on DFS)



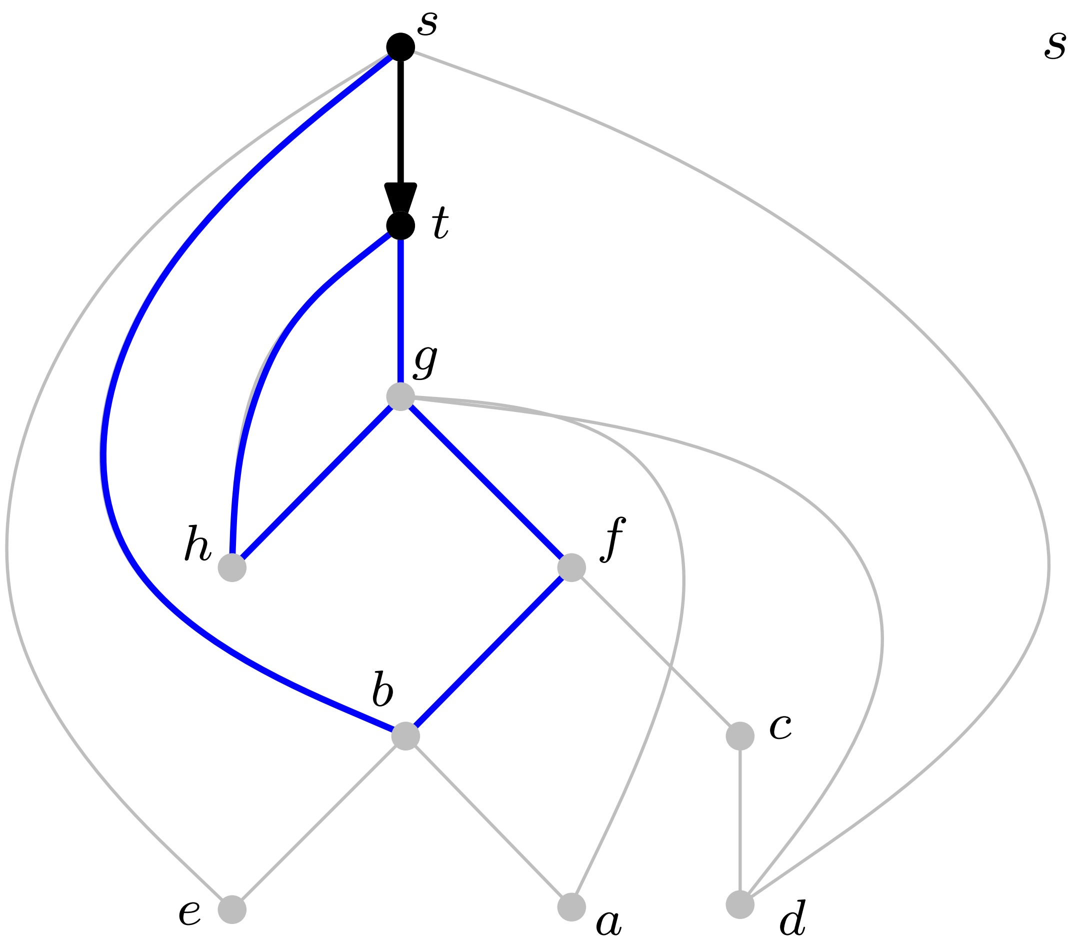

$s, t$


Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

# $st$-ordering

## Algorithm: $st$-ordering (example)

(Implementation details - Based on DFS)

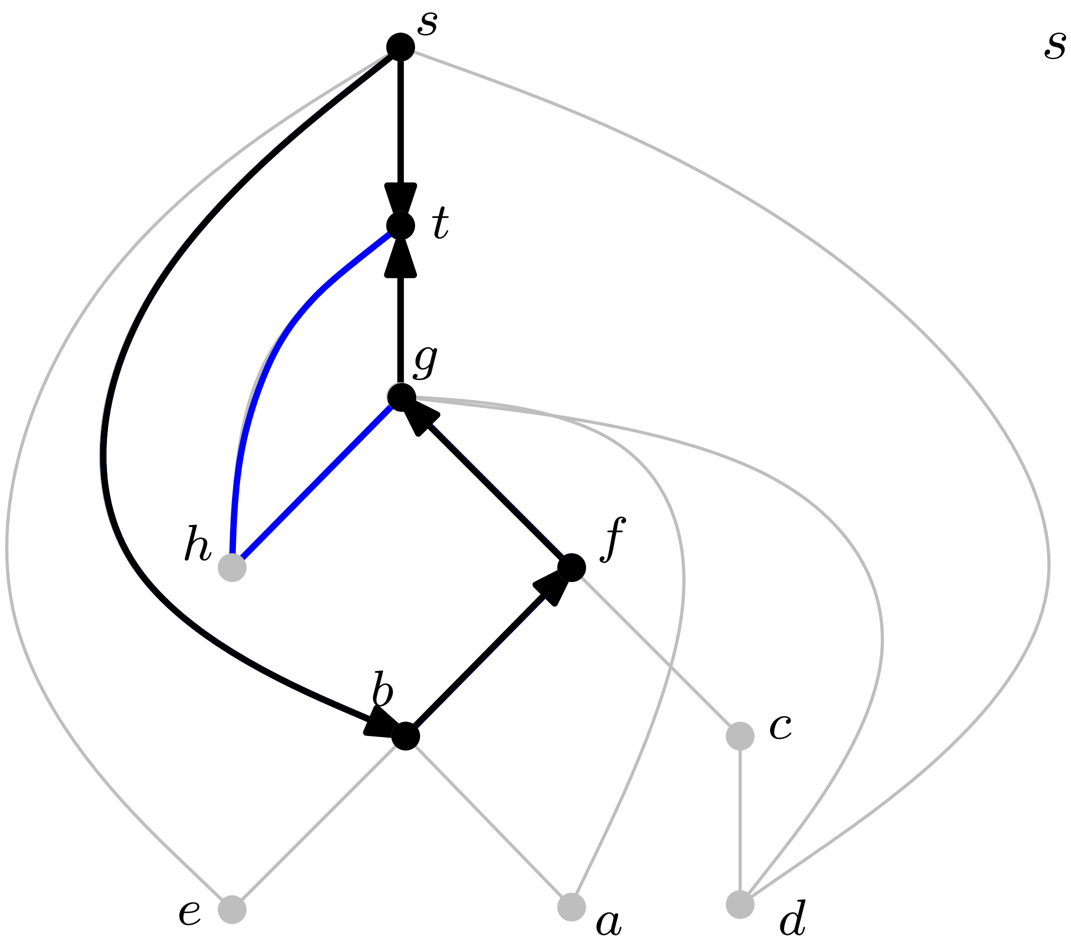



$s, t$

# $st$-ordering

## Algorithm: $st$-ordering (example)

(Implementation details - Based on DFS)




$s, t$

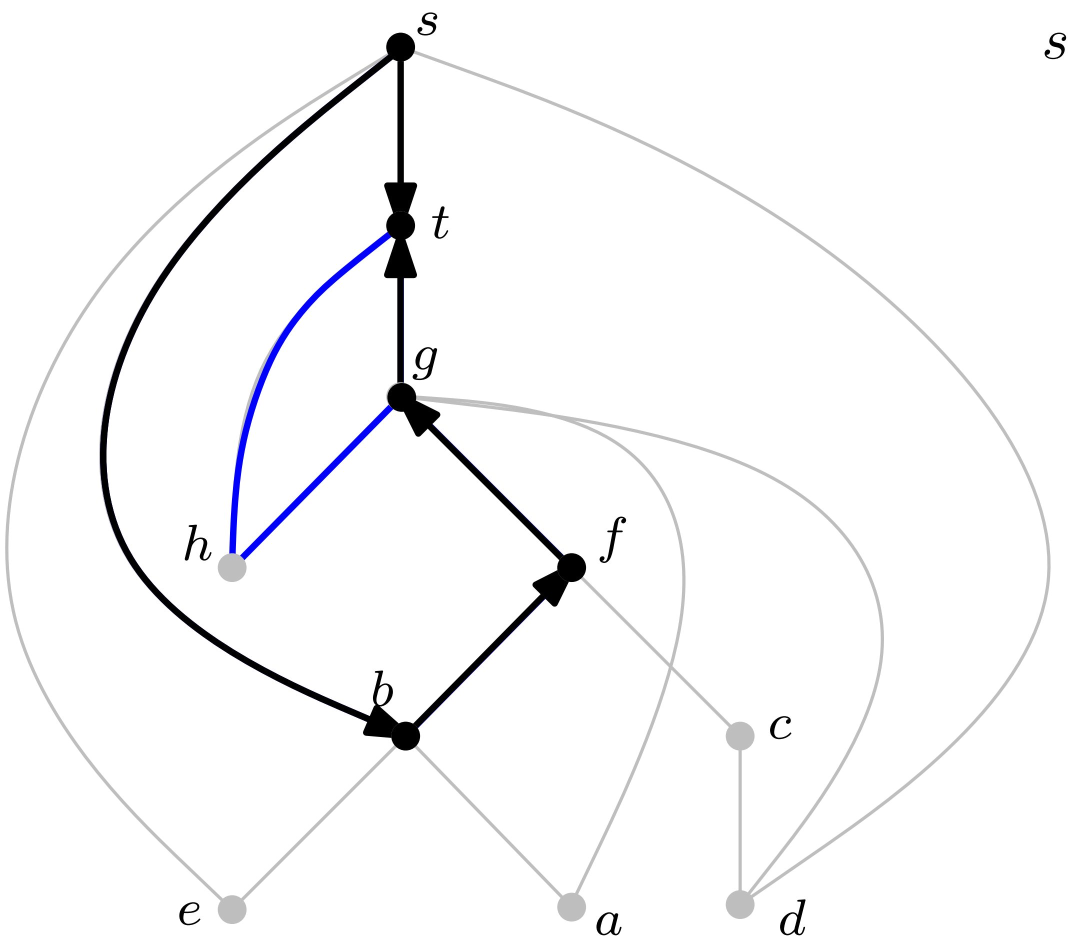


Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

# $st$-ordering

## Algorithm: $st$-ordering (example)

(Implementation details - Based on DFS)




$s, \underline{b}, \underline{f}, \underline{g}, t$

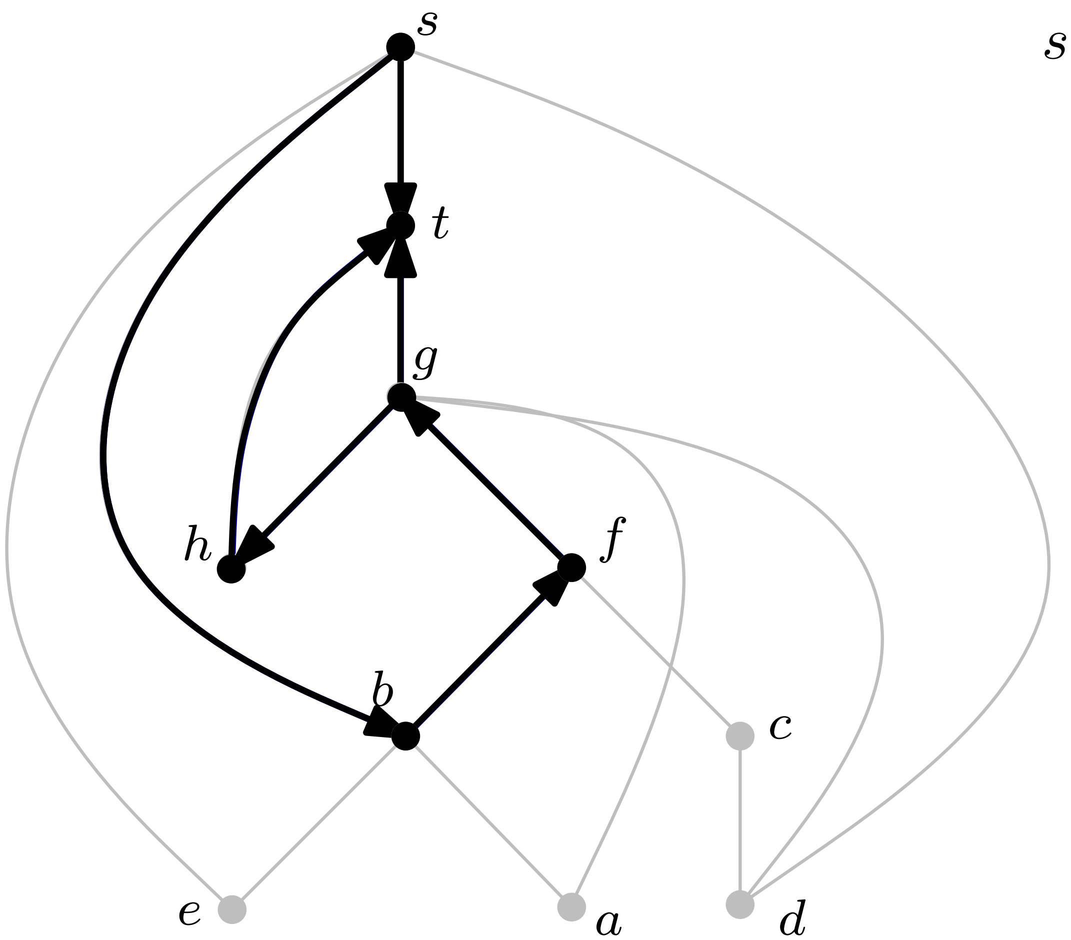


Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

# $st$-ordering

## Algorithm: $st$-ordering (example)

(Implementation details - Based on DFS)




$s, \underline{b}, \underline{f}, \underline{g}, t$

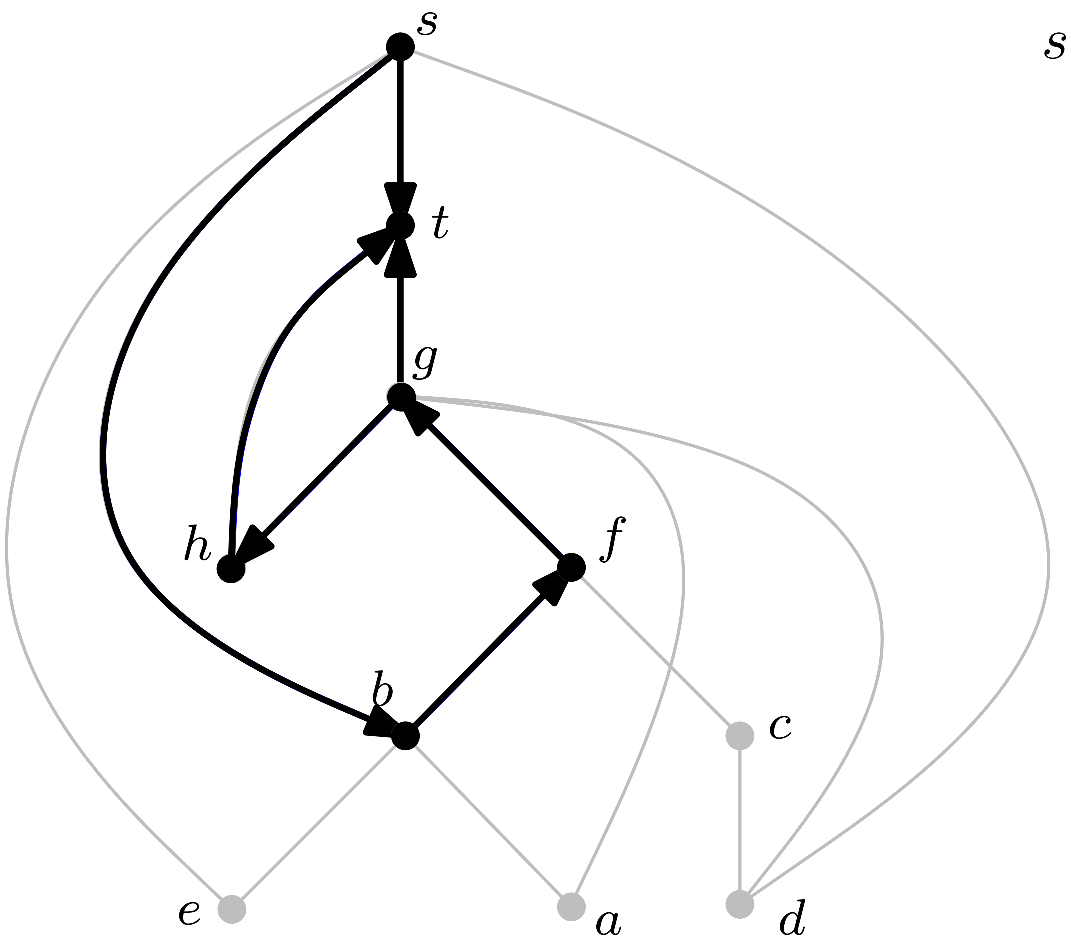


Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

# $st$-ordering

## Algorithm: $st$-ordering (example)

(Implementation details - Based on DFS)



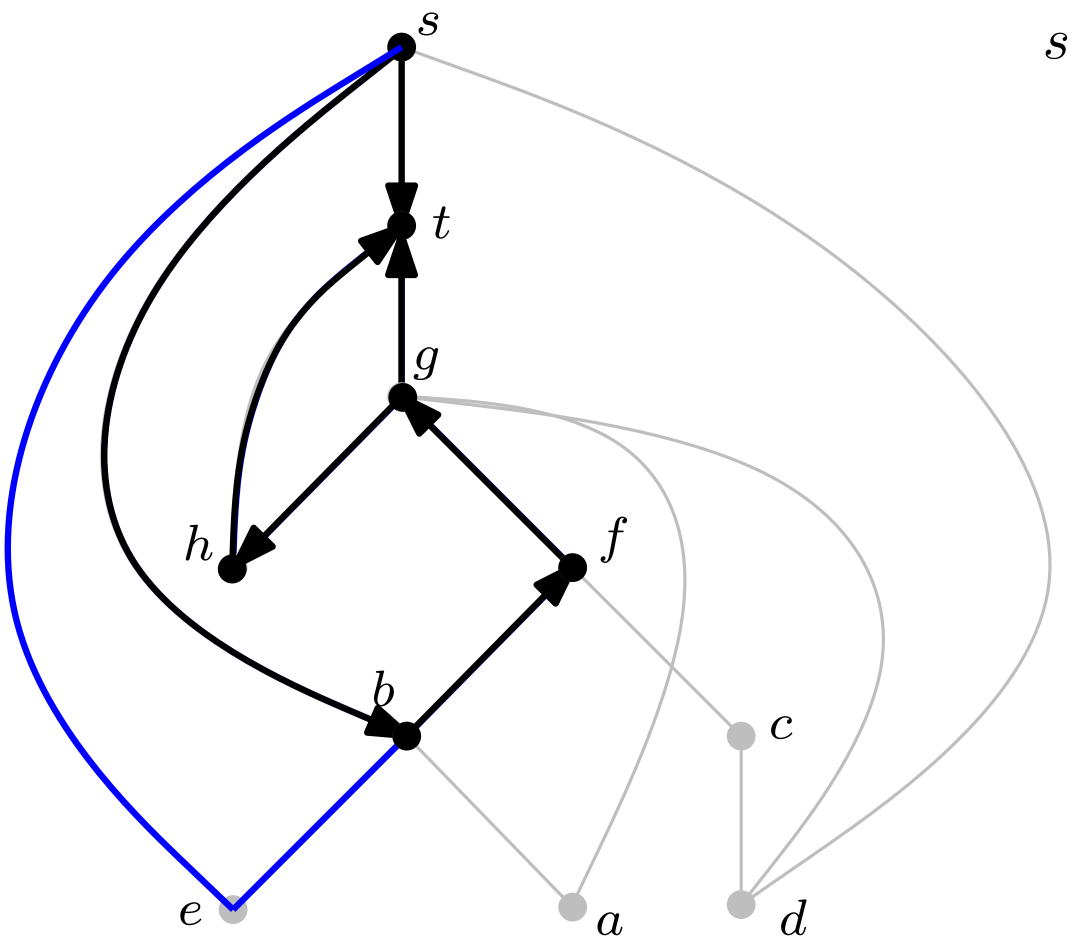

$s, b, f, g, \underline{h}, t$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

# $st$-ordering

## Algorithm: $st$-ordering (example)
(Implementation details - Based on DFS)




$s, b, f, g, \underline{h}, t$

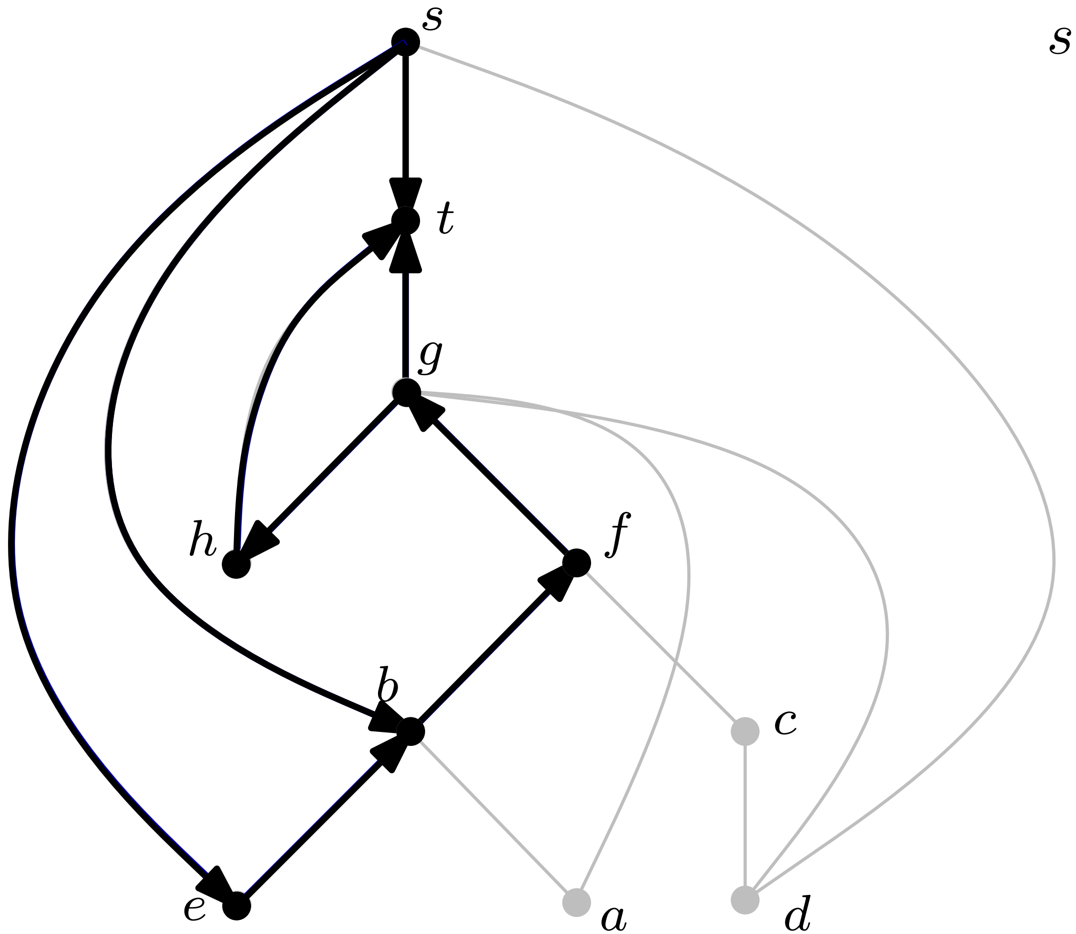


Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

# $st$-ordering

## Algorithm: $st$-ordering (example)

(Implementation details - Based on DFS)




$s, b, f, g, \underline{h}, t$

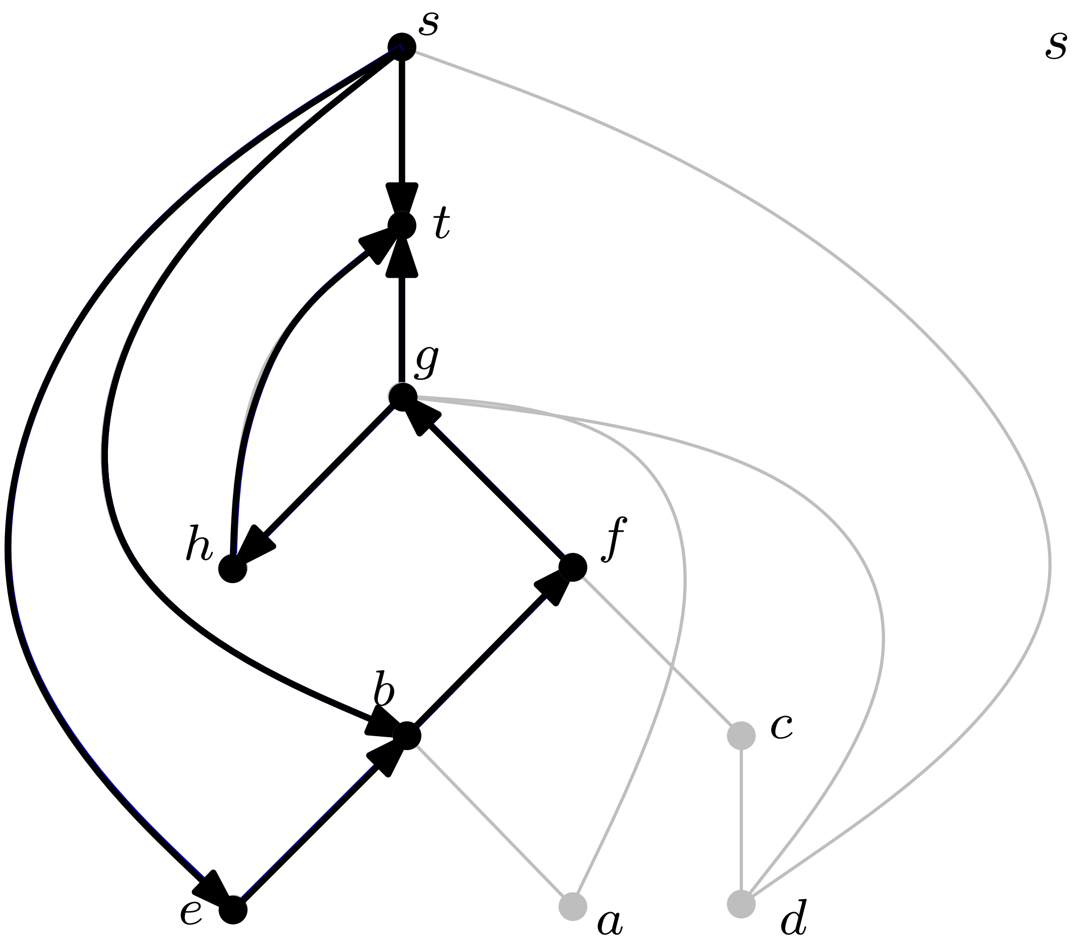


Institut für Theoretische Informatik
Lehrstuhl Algorithmik I

# $st$-ordering

## Algorithm: $st$-ordering (example)

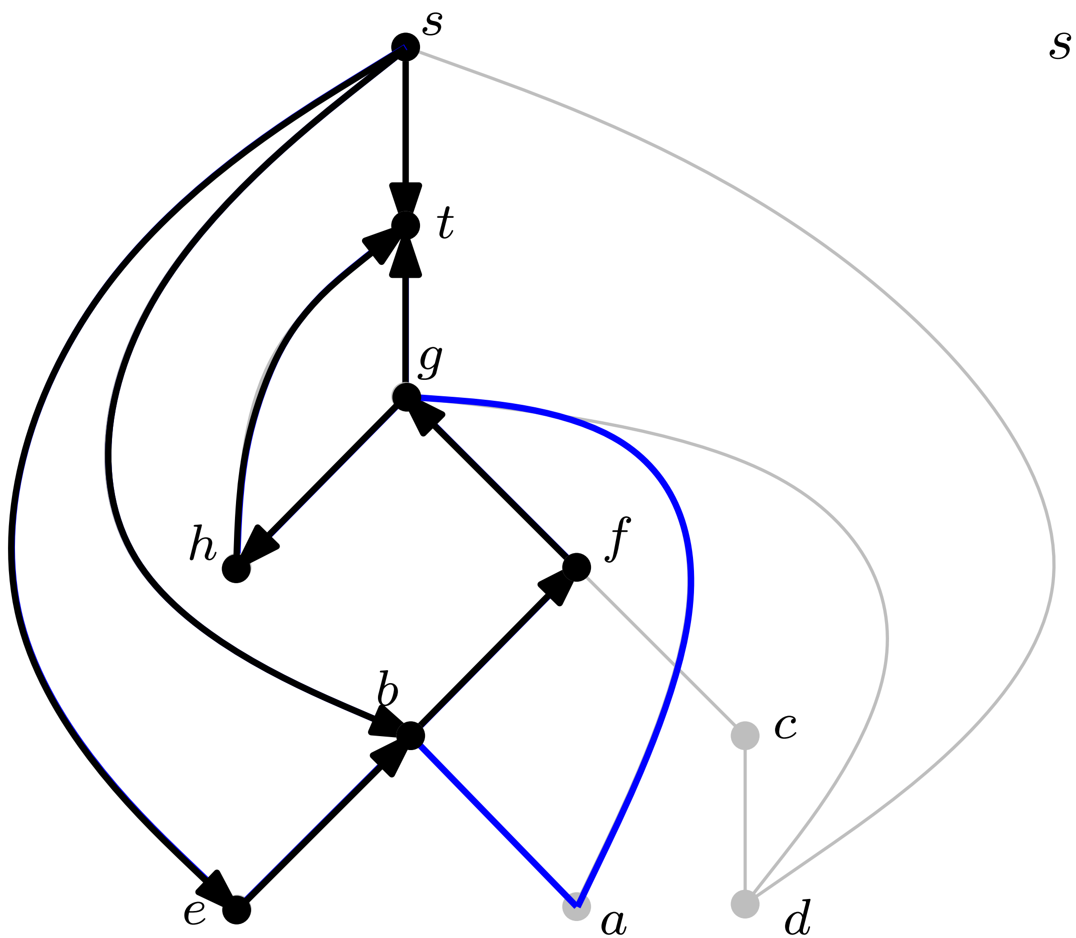
(Implementation details - Based on DFS)




$s, \underline{e}, b, f, g, h, t$

# $st$-ordering

## Algorithm: $st$-ordering (example)
(Implementation details - Based on DFS)

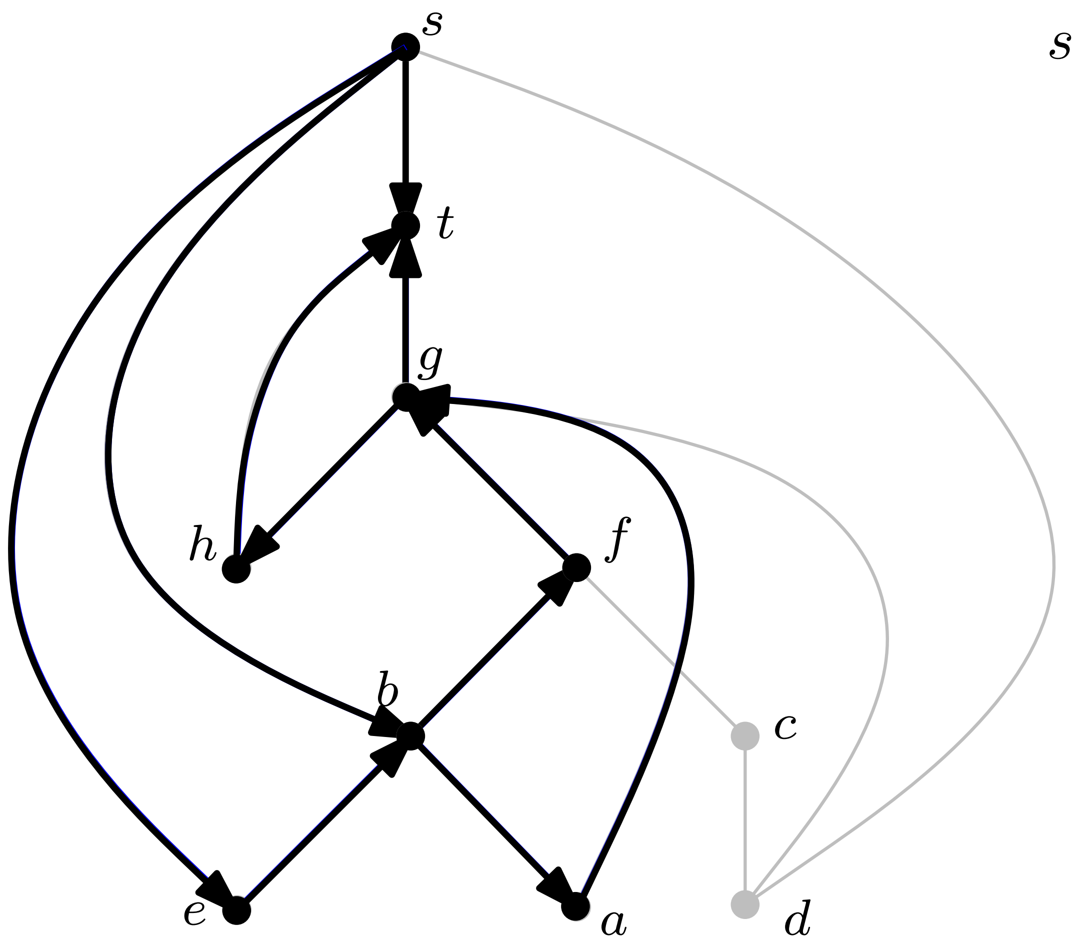



$s, \underline{e}, b, f, g, h, t$

# $st$-ordering

## Algorithm: $st$-ordering (example)

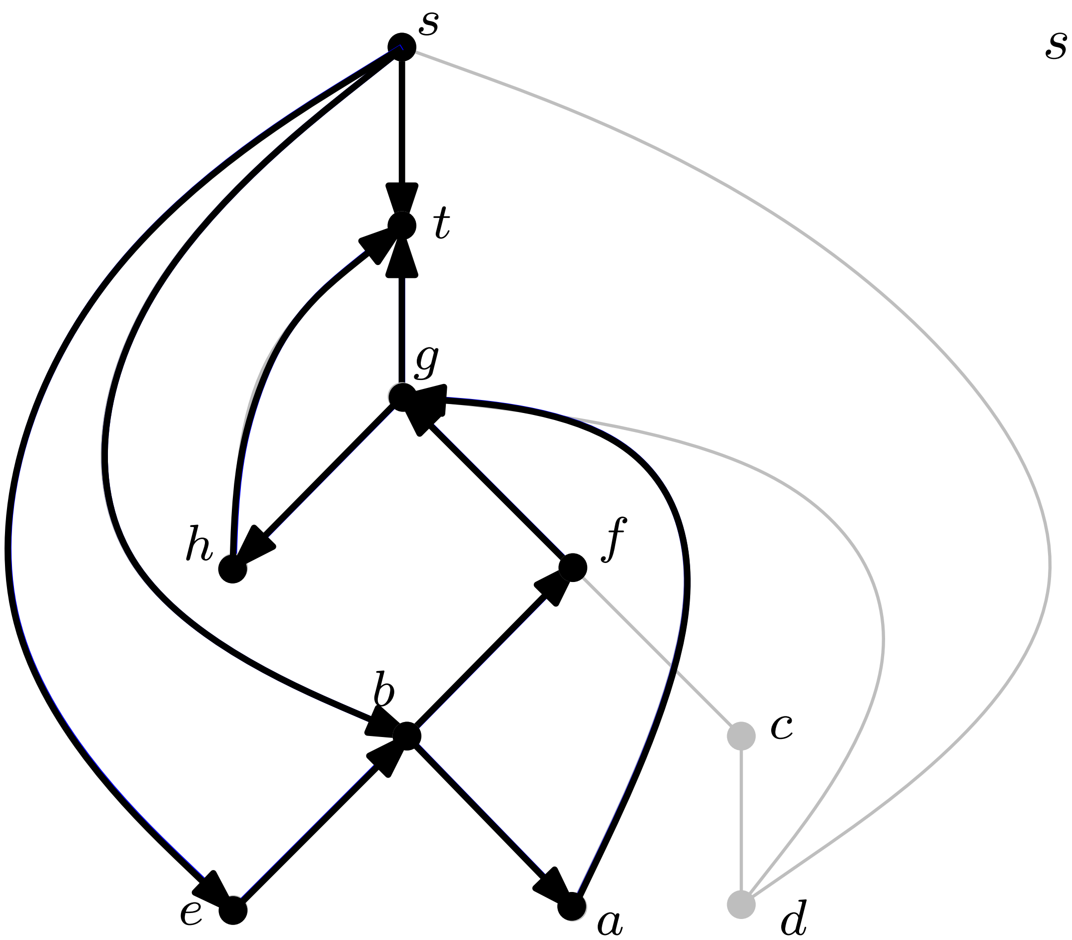
(Implementation details - Based on DFS)




$s, \underline{e}, b, f, g, h, t$

# $st$-ordering

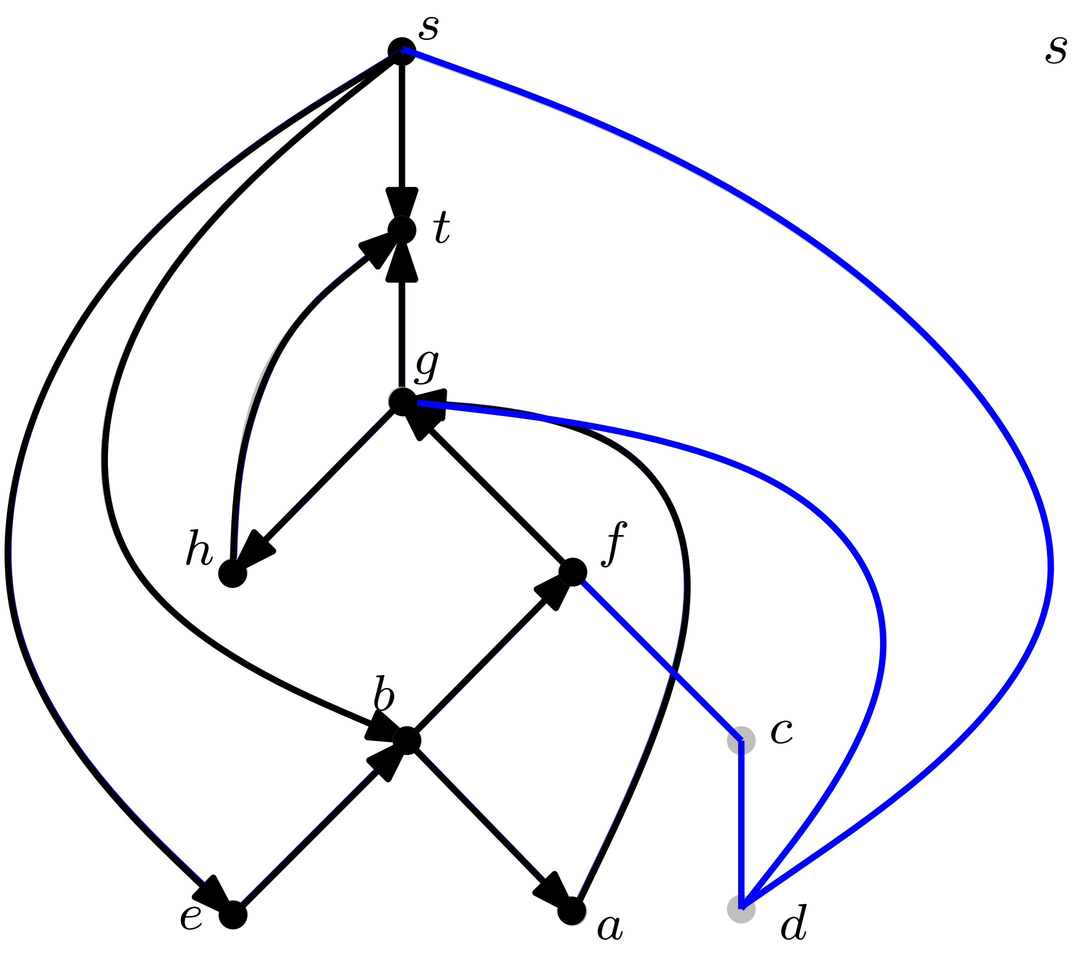
## Algorithm: $st$-ordering (example)
(Implementation details - Based on DFS)




$s, e, b, \underline{a}, f, g, h, t$

# $st$-ordering

## Algorithm: $st$-ordering (example)
(Implementation details - Based on DFS)

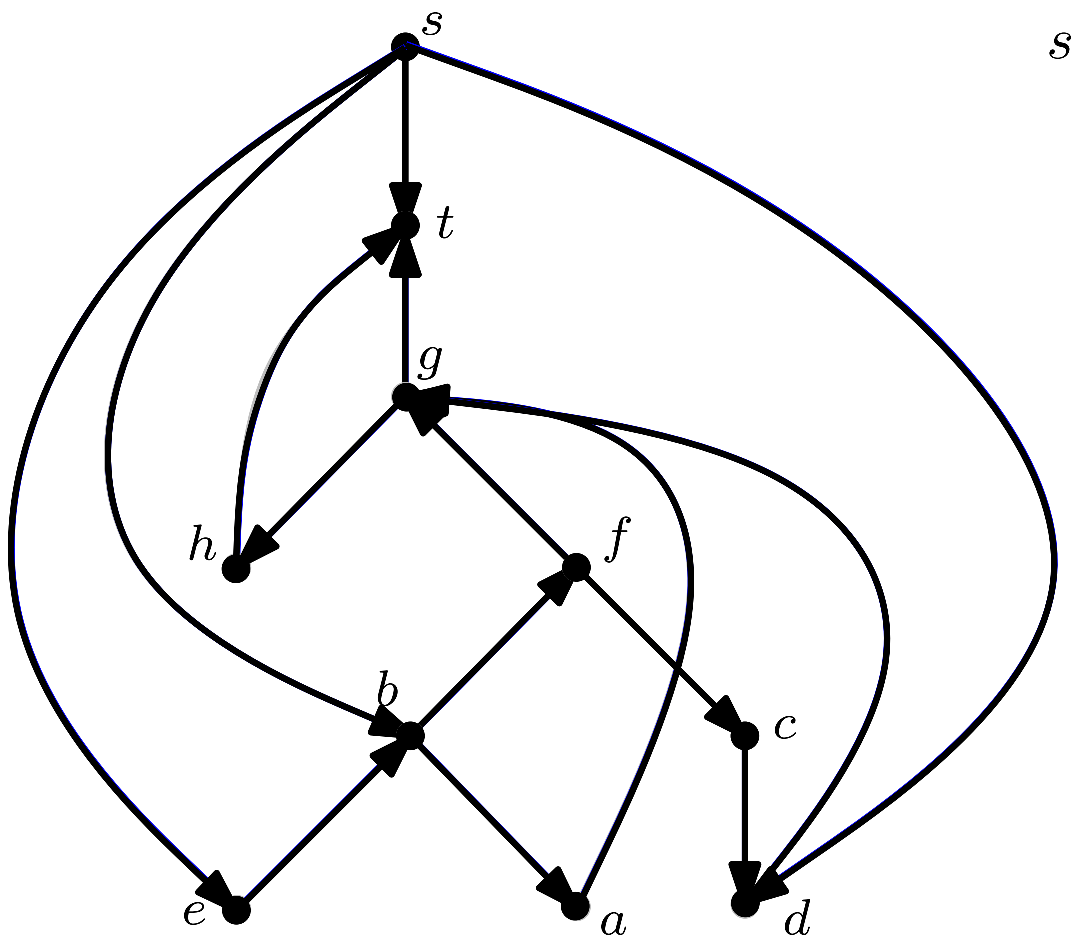



$s, e, b, \underline{a}, f, g, h, t$

# $st$-ordering

## Algorithm: $st$-ordering (example)

(Implementation details - Based on DFS)

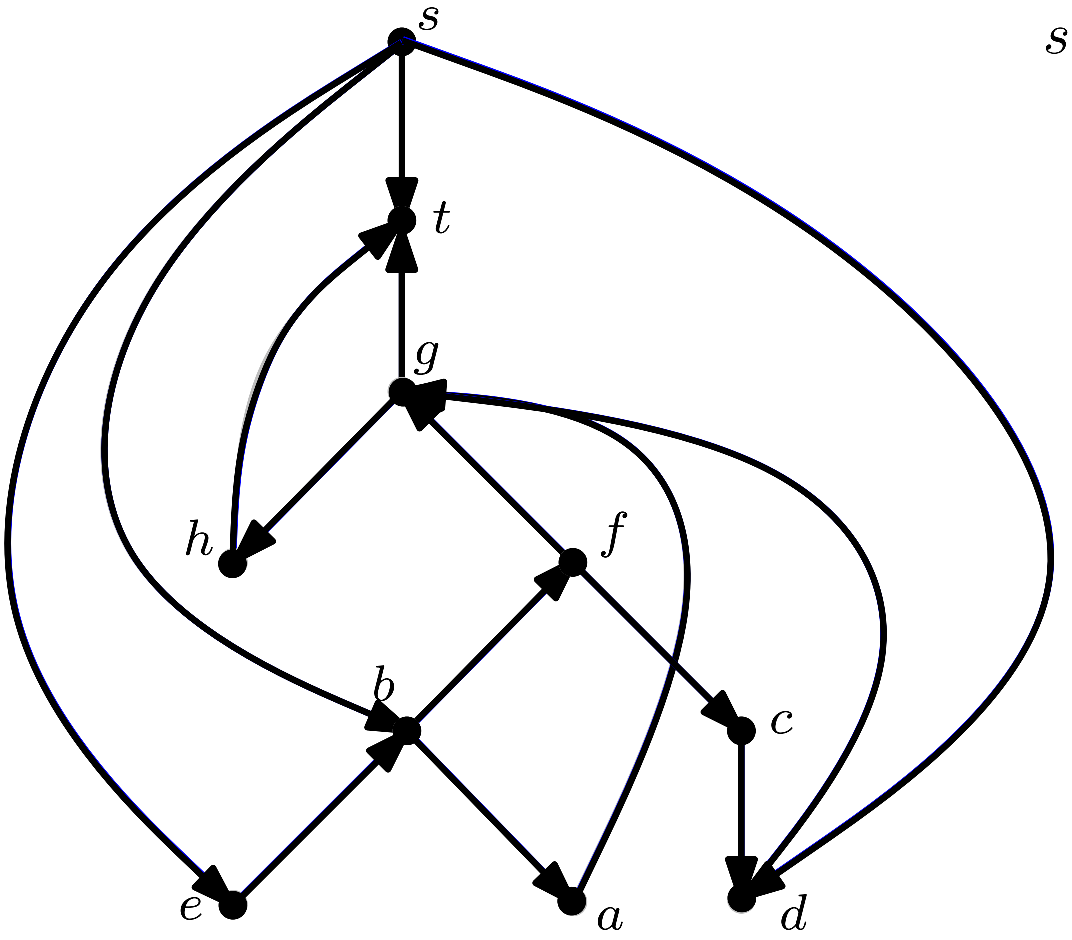



$s, e, b, \underline{a}, f, g, h, t$

# $st$-ordering

## Algorithm: $st$-ordering (example)

(Implementation details - Based on DFS)




$s, e, b, a, f, \underline{c}, \underline{d}, g, h, t$

# $st$-ordering

## Algorithm $st$-ordering

**Data:** Undirected biconnected graph $G = (V, E)$, edge $\{s, t\} \in E$
**Result:** List $L$ of nodes representing an $st$-ordering of $G$)

```
dfs(vertex v) begin
    i ← i + 1;  DFS[v] ← i;
    while there exists non-enumerated e = {v, w} do
        DFS[e] ← DFS[v];
        if w not enumerated then
            CHILDEDGE[v] ← e; PARENT[w] ← v;
            dfs(w);
        else
            {w, x} ← CHILDEDGE[w]; D[{w, x}] ← D[{w, x}] ∪ {e};
            if x ∈ L then process_ears(w → x);
            ;

begin
    initialize L as {s, t};
    DFS[s] ← 1;  i ← 1; DFS[{s, t}] ← 1; CHILDEDGE[s] ← {s, t};
    dfs(t);
```

# $st$-ordering

## Function *process_ears*

**process_ears(tree edge $w \rightarrow x$) begin**

- **foreach** $v \hookrightarrow w \in D[w \rightarrow x]$ **do**
  - $u \leftarrow v$;
  - **while** $u \notin L$ **do** $u \leftarrow PARENT[u]$;
  - ;
  - $P \leftarrow (u \stackrel{*}{\rightarrow} v \hookrightarrow w)$;
  - **if** $w \rightarrow x$ *is oriented from* $w$ *to* $x$ *(resp.from* $x$ *to* $w$*)* **then**
    - orient $P$ from $w$ to $u$ (resp. from $u$ to $w$);
    - paste the inner nodes of $P$ to $L$
    - before (resp. after) $u$ ;
  - **foreach** *tree edge* $w' \rightarrow x'$ *of* $P$ **do** $process\_ears(w' \rightarrow x')$; ;
- $D[\{w, x\}] \leftarrow \emptyset$;

# $st$-ordering

**Theorem**

The described algorithm produces an $st$-ordering of a given biconnected graph $G = (V, E)$ in $O(E)$ time.

# $st$-ordering

**Theorem**

The described algorithm produces an $st$-ordering of a given biconnected graph $G = (V, E)$ in $O(E)$ time.

**Theorem (Biedl & Kant 98)**

A biconnected graph $G$ with vertex-degree at most 4 admits an orthogonal drawing such that:

- Area is $(m - n + 1) \times n + 1$
- Each edge (except maybe for one) has at most 2 bends
- The exceptional edge has at most 3 bends
- The total number if bends is at most $2m - 2n + 4$
- If $G$ is plane, the orthogonal drawing is planar
- Finally, provided an $st$-ordering such a drawing can be constructed in $O(n)$ time.

# $st$-ordering

**Theorem**

The described algorithm produces an $st$-ordering of a given biconnected graph $G = (V, E)$ in $O(E)$ time.

**Theorem (Biedl & Kant 98)**

A biconnected graph $G$ with vertex-degree at most 4 admits an orthogonal drawing such that:

- Area is $(m - n + 1) \times n + 1$
- Each edge (except maybe for one) has at most 2 bends
- The exceptional edge has at most 3 bends
- The total number if bends is at most $2m - 2n + 4$
- If $G$ is plane, the orthogonal drawing is planar
- Finally, provided an $st$-ordering such a drawing can be constructed in $O(n)$ time.

Together imply an $O(n)$ algorithm for constructing an orthogonal drawing.