16.2.2
Topological ordering

Total recall: Order on a set

Order or **strict total order** on a set X is a binary relation $\prec$ on X, such that

1. Transitivity: $\forall x.y, z \in X \quad x \prec y$ and $y \prec z \implies x \prec z$.
2. For any $x, y \in X$, exactly one of the following holds:
   $x \prec y$, $y \prec x$ or $x = y$.

Cannot have $x_1, \ldots, x_m \in X$, such that $x_1 \prec X_2, \ldots, x_{m-1} \prec x_m, x_m \prec x_1$, because...

Order on a (finite) set X: listing the elements of X from smallest to largest.

Total recall: Order on a set

Order or **strict total order** on a set X is a binary relation $\prec$ on X, such that

1. Transitivity: $\forall x.y, z \in X \quad x \prec y$ and $y \prec z \implies x \prec z$.
2. For any $x, y \in X$, exactly one of the following holds: $x \prec y$, $y \prec x$ or $x = y$.

Cannot have $x_1, \ldots, x_m \in X$, such that $x_1 \prec X_2, \ldots, x_{m-1} \prec x_m, x_m \prec x_1$, because...

Order on a (finite) set X: listing the elements of X from smallest to largest.

Total recall: Order on a set

Order or **strict total order** on a set X is a binary relation $\prec$ on X, such that

1. Transitivity: $\forall x.y, z \in X \quad x \prec y$ and $y \prec z \implies x \prec z$.
2. For any $x, y \in X$, exactly one of the following holds: $x \prec y$, $y \prec x$ or $x = y$.

Cannot have $x_1, \ldots, x_m \in X$, such that $x_1 \prec X_2, \ldots, x_{m-1} \prec x_m, x_m \prec x_1$, because...

Order on a (finite) set X: listing the elements of X from smallest to largest.

Convention about writing edges

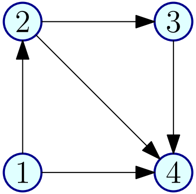
1. Undirected graph edges:

$$uv = \{u, v\} = vu \in E$$

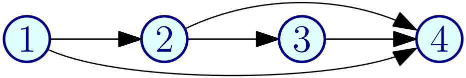
2. Directed graph edges:

$$u \to v \qquad \equiv \qquad (u, v) \qquad \equiv \qquad (u \to v)$$

Topological Ordering/Sorting



Graph G



Topological Ordering of G

Definition

A **topological ordering**/**topological sorting** of $G = (V, E)$ is an ordering $\prec$ on V such that if $(u \to v) \in E$ then $u \prec v$.

Informal equivalent definition:

One can order the vertices of the graph along a line (say the x-axis) such that all edges are from left to right.

DAGs and Topological Sort

Lemma

A directed graph G can be topologically ordered $\iff$ G is a DAG.

Need to show both directions.

DAGs and Topological Sort

Lemma

A directed graph G is a DAG $\implies$ *G can be topologically ordered.*

Proof.

Consider the following algorithm:

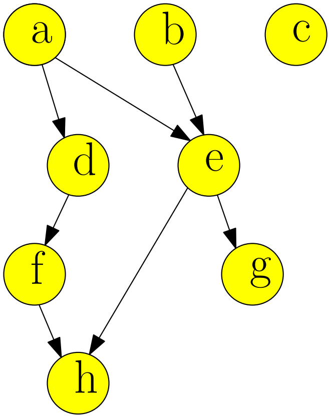
1. Pick a source u, output it.
2. Remove u and all edges out of u.
3. Repeat until graph is empty.

Exercise: prove this gives topological sort. □

Topological ordering in linear time

Exercise: show algorithm can be implemented in $O(m + n)$ time.

Topological Sort: Example



DAGs and Topological Sort

Lemma

A directed graph G *can be topologically ordered* $\implies$ G *is a* DAG.

Proof.

Proof by contradiction. Suppose G is not a DAG and has a topological ordering $\prec$. G has a cycle

$$C = u_1 \to u_2 \to \cdots \to u_k \to u_1.$$

Then $u_1 \prec u_2 \prec \ldots \prec u_k \prec u_1$

$\implies u_1 \prec u_1$.

A contradiction (to $\prec$ being an order). Not possible to topologically order the vertices. □

DAGs and Topological Sort

Lemma

A directed graph G can be topologically ordered $\implies$ G is a DAG.

Proof.

Proof by contradiction. Suppose G is not a DAG and has a topological ordering $\prec$. G has a cycle

$$C = u_1 \to u_2 \to \cdots \to u_k \to u_1.$$

Then $u_1 \prec u_2 \prec \ldots \prec u_k \prec u_1$
$\implies u_1 \prec u_1$.
A contradiction (to $\prec$ being an order). Not possible to topologically order the vertices. □

Regular sorting and DAGs

DAGs and Topological Sort

1. **Note:** A DAG G may have many different topological sorts.

2. **Exercise:** What is a DAG with the most number of distinct topological sorts for a given number n of vertices?

3. **Exercise:** What is a DAG with the least number of distinct topological sorts for a given number n of vertices?

THE END

...

(for now)