CS 341: Algorithms
Lec 07: Directed Graphs

Armin Jamshidpey Collin Roberts

Based on lecture notes by Éric Schost

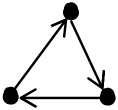
David R. Cheriton School of Computer Science, University of Waterloo

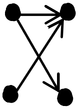
Winter 2025

Directed graphs basics

Definition:

- $G = (V, E)$ as in the undirected case, with the difference that edges are (directed) pairs (v, w)
  - edges also called arcs
  - v is the source node, w is the target
- a path is a sequence $v_1, \ldots, v_k$ of vertices, with (v_i, v_{i+1}) in E for all i. $k = 1$ is OK.
- a cycle is a path $v_1, \ldots, v_k, v_1$, $\boldsymbol{k \geq 2}$
- a directed acyclic graph (DAG) is a directed graph with no cycle





Directed graphs basics

Definition:

- the in-degree of v is the number of edges of the form (u, v)
- the out-degree of v is the number of edges of the form (v, w)

Data structures

- adjacency lists
- adjacency matrix (not symmetric anymore)

BFS and DFS for directed graphs

The algorithms work without any change. We will focus on DFS.
Still true:

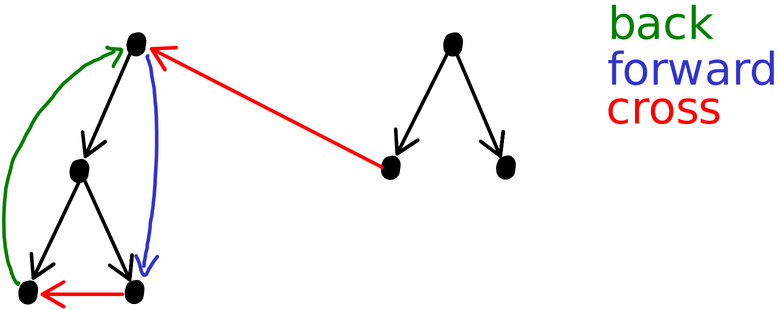
- we obtain a partition of V into vertex-disjoint trees $T_1, \ldots, T_k$
- when we start exploring a vertex v, any w with an unvisited path $v \rightsquigarrow w$ becomes a descendant of v (white path lemma)
- properties of start and finish times

But there can exist edges connecting the trees T_i

Classification of edges

Suppose we have a DFS forest. Edges of G are one of the following:

- **tree edges**
- **back edges:** from descendant to ancestor
- **forward edges:** from ancestor to descendant (but not tree edge)
- **cross edges:** all others



Classification of edges

```
explore(v)
    visited[v] = true
    start[v] = t, t++
    for all w neighbour of v do
        if visited[w] = false
            explore(w)                (v, w) tree edge
    finish[v] = t, t++
```

If w was visited:

- if w not finished, (v, w) **back edge**
- else if start$[v] <$ start$[w] <$ finish$[w]$, (v, w) **forward edge**
- else start$[w] <$ finish$[w] <$ start$[v]$, (v, w) **cross edge**

Testing acyclicity

Claim

G has a cycle if and only if there is a back edge in the DFS forest

Proof

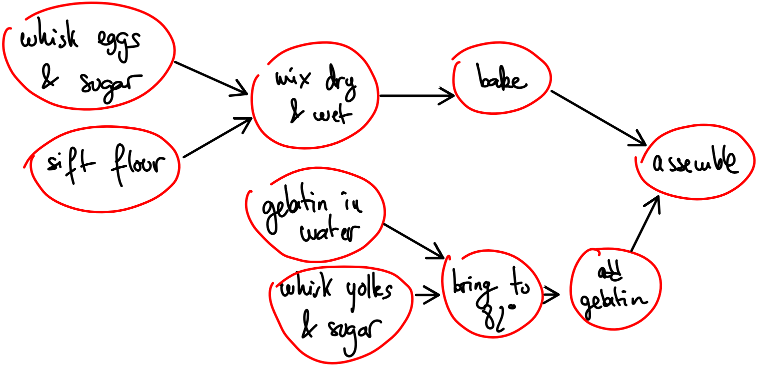
- Suppose there is a back edge (v, w). Then v is a descendant of w, so there is a path $w \rightsquigarrow v$, and a cycle $w \rightsquigarrow v \rightarrow w$
- Suppose there is a cycle $\boldsymbol{v_1, \ldots, v_{k-1}, v_k = v_1}$. Up to renumbering, assume we find v_1 first in the DFS.

  Starting from v_1, we will reach v_{k-1} (white path lemma). We check the edge (v_{k-1}, v_1), and v_1 is not finished. So back edge.

Consequence: acyclicity test in $\boldsymbol{O(n+m)}$

Topological ordering

Definition: Suppose $G = (V, E)$ is a DAG. A **topological order** is an ordering $<$ of V such that for any edge (v, w), we have $v < w$.



No such order if there are cycles.

From a DFS forest

Observation:

- start times do not help
- finish times do, but we have to reverse their order

From a DFS forest

Claim

Suppose that V is ordered using the reverse of the finishing order: $v < w \iff \textsf{finish}[w] < \textsf{finish}[v]$.

This is a topological order.

Proof. Have to prove: for any edge (v, w), $\textsf{finish}[w] < \textsf{finish}[v]$.

- if we discover v before w, w will become a descendant of v (white path lemma), and we finish exploring it before we finish v.
- if we discover w before v, because there is no path $w \rightsquigarrow v$ (G is a DAG), we will finish w before we start v.

Consequence: topological order in $O(n + m)$.

Strong connectivity

Definition. A directed graph G is strongly connected if for all v, w in G, there is a path $\boldsymbol{v} \rightsquigarrow \boldsymbol{w}$ (and thus a path $\boldsymbol{w} \rightsquigarrow \boldsymbol{v}$).

Observation

G is strongly connected iff there exists s such that for all w, there are paths $\boldsymbol{s} \rightsquigarrow \boldsymbol{w}$ and $\boldsymbol{w} \rightsquigarrow \boldsymbol{s}$.

Proof.

- $\Longrightarrow$ is obvious.
- For $\Longleftarrow$, take vertices v, w. We have paths $v \rightsquigarrow s$ and $s \rightsquigarrow w$, so $v \rightsquigarrow w$. Same thing with $w \rightsquigarrow v$.

Testing strong connectivity

Algorithm:

- call **explore twice**, starting from a same vertex s
- edges reversed the second time

Correctness:

- first run tells whether for all v, there is a path $s \rightsquigarrow v$
- second one tells whether for all v, there is a path $s \rightsquigarrow v$ in the reverse graph (which is a path $v \rightsquigarrow s$ in G)

Consequence: test in $\boldsymbol{O(n+m)}$

Structure of directed graphs

Definition: a **strongly connected component** of G is

- a subgraph of G
- which is strongly connected
- but not contained in a larger strongly connected subgraph of G.

Exercise

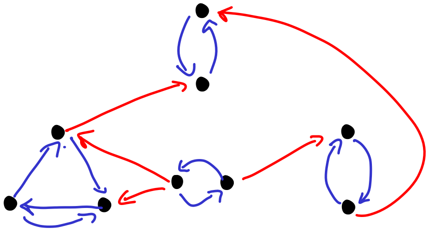
The vertices of strongly connected components form a partition of V.

Exercise

v and w are in the same strongly connected component if and only if there are paths $v \rightsquigarrow w$ and $w \rightsquigarrow v$.

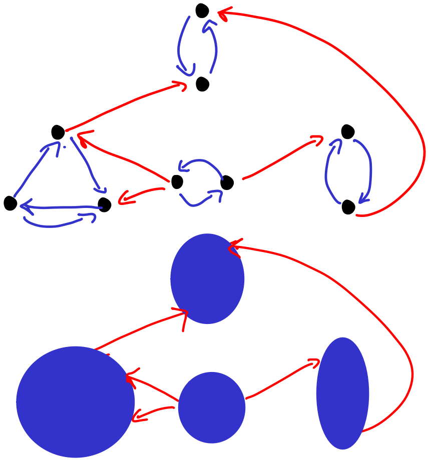
Structure of directed graphs

A directed graph G can be seen as a DAG of disjoint strongly connected components.



Structure of directed graphs

A directed graph G can be seen as a DAG of disjoint strongly connected components.



Kosaraju's algorithm for strongly connected components

Definition: for a directed graph $G = (V, E)$, the **reverse** (or **transpose**) graph $G^T = (V, E^T)$ is the graph with same vertices, and reversed edges.

SCC(G)
1. run a DFS on G and record finish times
2. run a DFS on G^T, with vertices ordered in **decreasing finish time**
3. return the trees in the DFS forest of G^T

Complexity: $\boldsymbol{O(n+m)}$ (don't forget the time to reverse G)

Exercise

check that the strongly connected components of G and G^T are the same

Correctness

Want to prove: for any vertices v, w, the following are equivalent.

(1) v and w are in the same strongly connected component of G

(2) v and w are in the same tree in the DFS forest of G^T (with vertices ordered in decreasing finish time)

Correctness

Want to prove: for any vertices v, w, the following are equivalent.

(1) v and w are in the same strongly connected component of G

(2) v and w are in the same tree in the DFS forest of G^T (with vertices ordered in decreasing finish time)

Proof of $1 \implies 2$ (order of the vertices does not matter here)
Let C be the strongly connected component of G that contains v and w

Correctness

Want to prove: for any vertices v, w, the following are equivalent.

(1) v and w are in the same strongly connected component of G

(2) v and w are in the same tree in the DFS forest of G^T (with vertices ordered in decreasing finish time)

Proof of $1 \implies 2$ (order of the vertices does not matter here)
Let C be the strongly connected component of G that contains v and w

Let s be the first vertex of C that we visit in the DFS of G^T

- there is a path $s \leadsto v$ in G^T
- all vertices on this path are in C (easy)
- so they are all unvisited when we arrive at s
- so v becomes a descendant of s **white path lemma**
- same for w

Correctness

Proof of $2 \implies 1$.
Let T be the tree in the DFS forest of G^T that contains v and w, with root s

We prove that for every vertex t in T, s **and** t **are in the same strongly connected component of** G**.**

Correctness

Proof of $2 \implies 1$.
Let T be the tree in the DFS forest of G^T that contains v and w, with root s

We prove that for every vertex t in T, s **and** t **are in the same strongly connected component of** G**.**

(1) for all t in T, there is a path $\boldsymbol{s} \rightsquigarrow \boldsymbol{t}$ in G^T, so there is a path $\boldsymbol{t} \rightsquigarrow \boldsymbol{s}$ in G

Correctness

Proof of $2 \implies 1$.

Let T be the tree in the DFS forest of G^T that contains v and w, with root s

We prove that for every vertex t in T, **s and t are in the same strongly connected component of G.**

(1) for all t in T, there is a path $\boldsymbol{s} \rightsquigarrow \boldsymbol{t}$ in G^T, so there is a path $\boldsymbol{t} \rightsquigarrow \boldsymbol{s}$ in G

(2) now we prove: for all t in T, t is a descendant of s in the DFS forest of G (this gives a path $\boldsymbol{s} \rightsquigarrow \boldsymbol{t}$ in G)

Correctness

Want to prove: for all t in T, t is a descendant of s in the DFS forest of G.

Correctness

Want to prove: for all t in T, t is a descendant of s in the DFS forest of G.

By induction: suppose it is true for some t in T, and prove it is true for its children. So let u be a child of t in T.

Correctness

Want to prove: for all t in T, t is a descendant of s in the DFS forest of G.

By induction: suppose it is true for some t in T, and prove it is true for its children. So let u be a child of t in T.

- $\mathsf{start}[s] \leq \mathsf{start}[t] < \mathsf{finish}[t] \leq \mathsf{finish}[s]$ induction assumption
- by definition of s, $\mathsf{finish}[u] < \mathsf{finish}[s]$

Correctness

Want to prove: for all t in T, t is a descendant of s in the DFS forest of G.

By induction: suppose it is true for some t in T, and prove it is true for its children. So let u be a child of t in T.

- $\text{start}[s] \leq \text{start}[t] < \text{finish}[t] \leq \text{finish}[s]$ induction assumption
- by definition of s, $\text{finish}[u] < \text{finish}[s]$, so our options are
  (1) $\text{start}[s] < \text{start}[u] < \text{finish}[u] < \text{finish}[s]$ $\quad$ [( )]
  (2) $\text{start}[u] < \text{finish}[u] < \text{start}[s] < \text{finish}[s]$ $\quad$ ( ) [ ]

Correctness

Want to prove: for all t in T, t is a descendant of s in the DFS forest of G.

By induction: suppose it is true for some t in T, and prove it is true for its children. So let u be a child of t in T.

- $\mathsf{start}[s] \le \mathsf{start}[t] < \mathsf{finish}[t] \le \mathsf{finish}[s]$ induction assumption
- by definition of s, $\mathsf{finish}[u] < \mathsf{finish}[s]$, so our options are
  - (1) $\mathsf{start}[s] < \mathsf{start}[u] < \mathsf{finish}[u] < \mathsf{finish}[s]$ [( )]
  - (2) $\mathsf{start}[u] < \mathsf{finish}[u] < \mathsf{start}[s] < \mathsf{finish}[s]$ ( ) [ ]
- if (2), with our induction assumption, we get $\mathsf{start}[u] < \mathsf{start}[t]$
- since (t, u) is in T, (u, t) is in G. With previous item, we get: t is a descendant of u in the DFS of G (white path)

Correctness

Want to prove: for all t in T, t is a descendant of s in the DFS forest of G.

By induction: suppose it is true for some t in T, and prove it is true for its children. So let u be a child of t in T.

- $\text{start}[s] \le \text{start}[t] < \text{finish}[t] \le \text{finish}[s]$ induction assumption
- by definition of s, $\text{finish}[u] < \text{finish}[s]$, so our options are
  (1) $\text{start}[s] < \text{start}[u] < \text{finish}[u] < \text{finish}[s]$ [( )]
  (2) $\text{start}[u] < \text{finish}[u] < \text{start}[s] < \text{finish}[s]$ ( ) [ ]
- if (2), with our induction assumption, we get $\text{start}[u] < \text{start}[t]$
- since (t, u) is in T, (u, t) is in G. With previous item, we get: t is a descendant of u in the DFS of G (white path)
- this gives $\text{start}[u] < \textbf{start}[t] < \text{finish}[t] < \textbf{finish}[u]$

Correctness

Want to prove: for all t in T, t is a descendant of s in the DFS forest of G.

By induction: suppose it is true for some t in T, and prove it is true for its children. So let u be a child of t in T.

- $\text{start}[s] \le \text{start}[t] < \text{finish}[t] \le \text{finish}[s]$ induction assumption
- by definition of s, $\text{finish}[u] < \text{finish}[s]$, so our options are
  (1) $\text{start}[s] < \text{start}[u] < \text{finish}[u] < \text{finish}[s]$ [( )]
  (2) $\text{start}[u] < \text{finish}[u] < \text{start}[s] < \text{finish}[s]$ ( ) [ ]
- if (2), with our induction assumption, we get $\text{start}[u] < \text{start}[t]$
- since (t, u) is in T, (u, t) is in G. With previous item, we get: t is a descendant of u in the DFS of G (white path)
- this gives $\text{start}[u] <$ **$\text{start}[t]$** $< \text{finish}[t] <$ **$\text{finish}[u]$**
- but also **$\text{finish}[u]$** $< \text{start}[s] <$ **$\text{start}[t]$** from (2) and induction assumption

Correctness

Want to prove: for all t in T, t is a descendant of s in the DFS forest of G.

By induction: suppose it is true for some t in T, and prove it is true for its children. So let u be a child of t in T.

- $\text{start}[s] \le \text{start}[t] < \text{finish}[t] \le \text{finish}[s]$ induction assumption
- by definition of s, $\text{finish}[u] < \text{finish}[s]$, so our options are
  (1) $\text{start}[s] < \text{start}[u] < \text{finish}[u] < \text{finish}[s]$ [( )]
  (2) $\text{start}[u] < \text{finish}[u] < \text{start}[s] < \text{finish}[s]$ ( ) [ ]
- if (2), with our induction assumption, we get $\text{start}[u] < \text{start}[t]$
- since (t, u) is in T, (u, t) is in G. With previous item, we get: t is a descendant of u in the DFS of G (white path)
- this gives $\text{start}[u] < \textbf{start}[t] < \text{finish}[t] < \textbf{finish}[u]$
- but also $\textbf{finish}[u] < \text{start}[s] < \textbf{start}[t]$ from (2) and induction assumption
- so (2) impossible, and we must have (1)
- shows that u is a descendant of s in the DFS forest of G