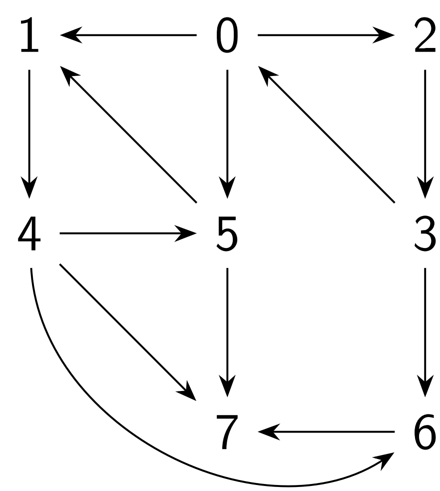
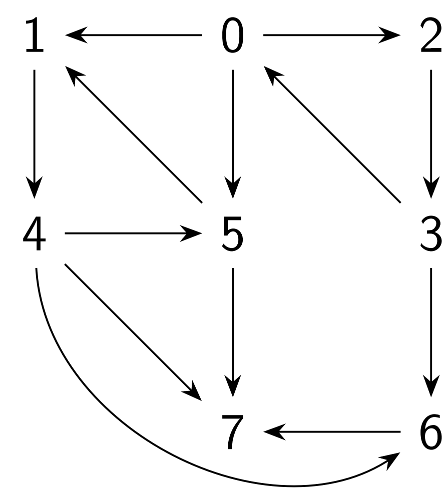
Directed cycles

- In a directed graph, a cycle must follow same direction
  - $0 \to 2 \to 3 \to 0$ is a cycle
  - $0 \to 5 \to 1 \leftarrow 0$ is not



Directed cycles

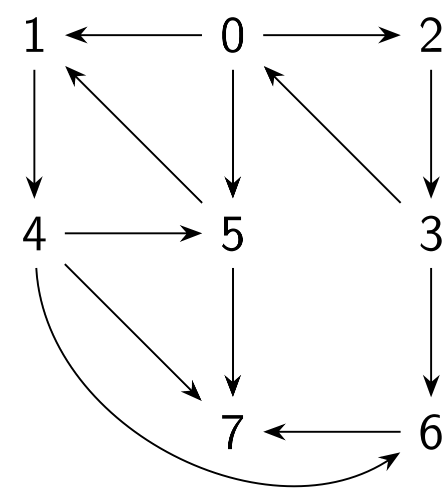
- In a directed graph, a cycle must follow same direction
  - $0 \rightarrow 2 \rightarrow 3 \rightarrow 0$ is a cycle
  - $0 \rightarrow 5 \rightarrow 1 \leftarrow 0$ is not

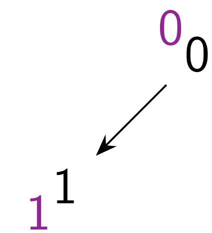


$^{0}0$

Directed cycles

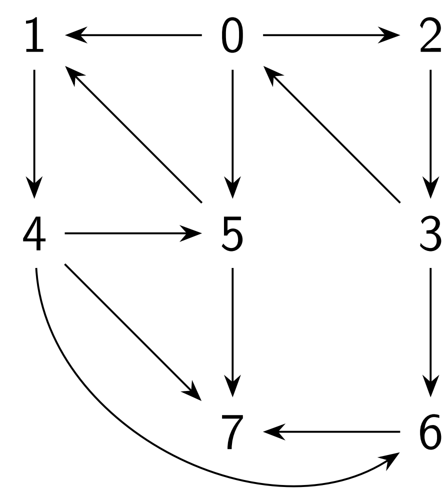
- In a directed graph, a cycle must follow same direction
  - $0 \rightarrow 2 \rightarrow 3 \rightarrow 0$ is a cycle
  - $0 \rightarrow 5 \rightarrow 1 \leftarrow 0$ is not

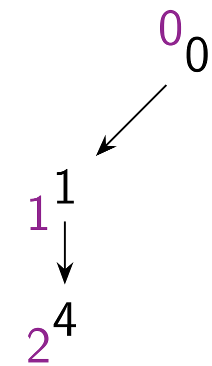




Directed cycles

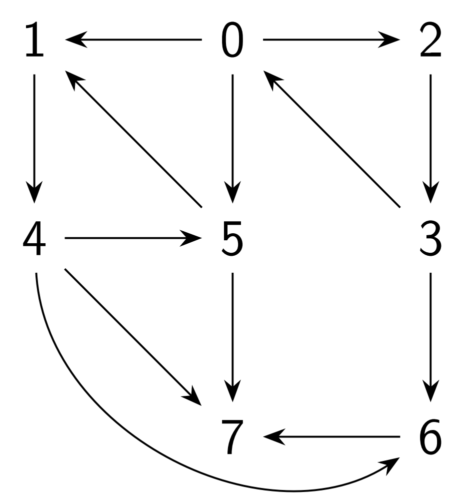
- In a directed graph, a cycle must follow same direction
  - $0 \rightarrow 2 \rightarrow 3 \rightarrow 0$ is a cycle
  - $0 \rightarrow 5 \rightarrow 1 \leftarrow 0$ is not

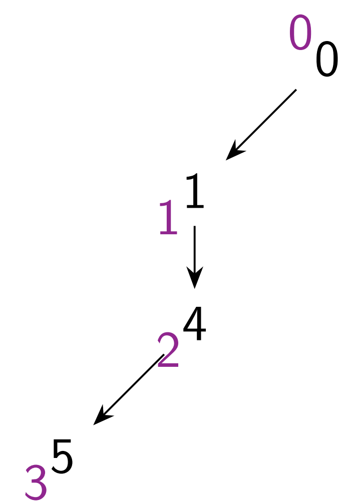




Directed cycles

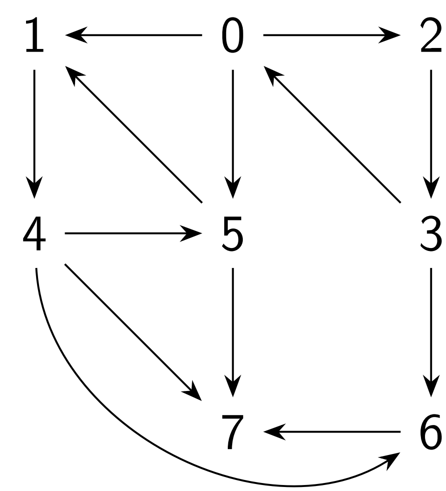
- In a directed graph, a cycle must follow same direction
  - $0 \to 2 \to 3 \to 0$ is a cycle
  - $0 \to 5 \to 1 \leftarrow 0$ is not

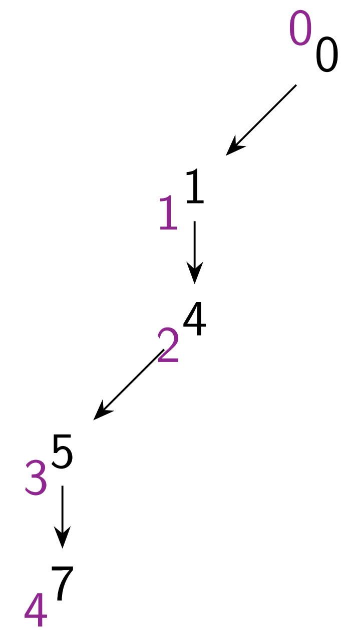




Directed cycles

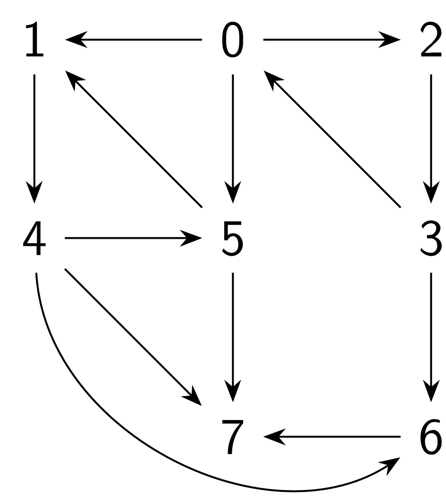
- In a directed graph, a cycle must follow same direction
  - $0 \rightarrow 2 \rightarrow 3 \rightarrow 0$ is a cycle
  - $0 \rightarrow 5 \rightarrow 1 \leftarrow 0$ is not

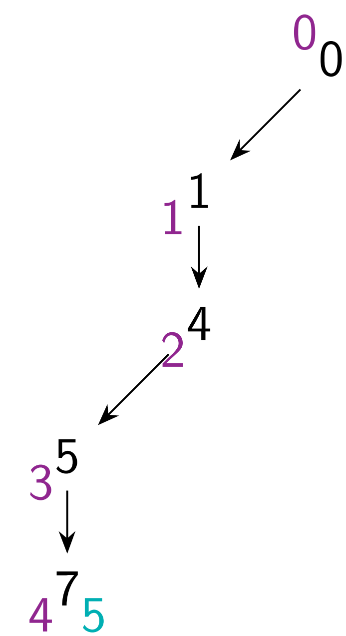




Directed cycles

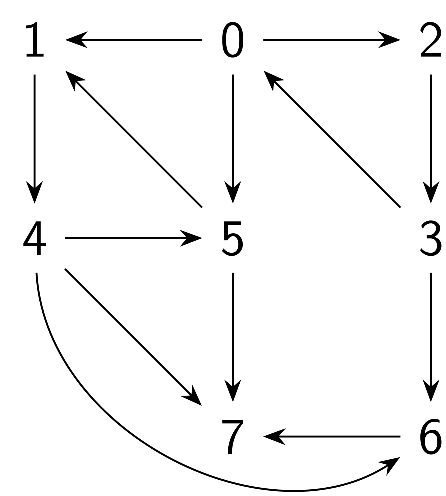
- In a directed graph, a cycle must follow same direction
  - $0 \to 2 \to 3 \to 0$ is a cycle
  - $0 \to 5 \to 1 \leftarrow 0$ is not

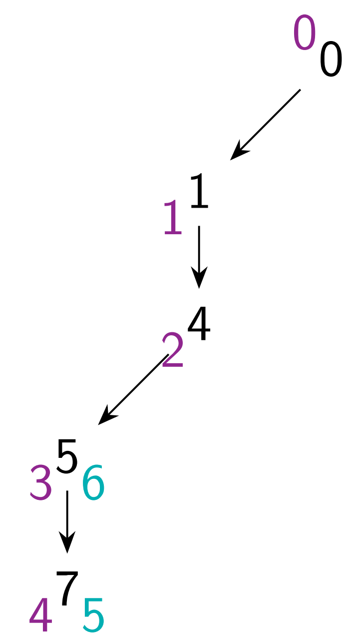




Directed cycles

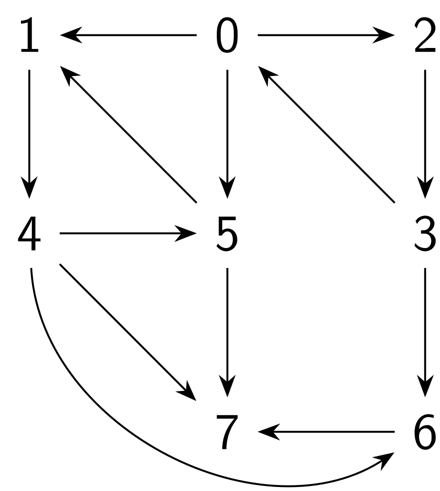
- In a directed graph, a cycle must follow same direction
  - $0 \rightarrow 2 \rightarrow 3 \rightarrow 0$ is a cycle
  - $0 \rightarrow 5 \rightarrow 1 \leftarrow 0$ is not

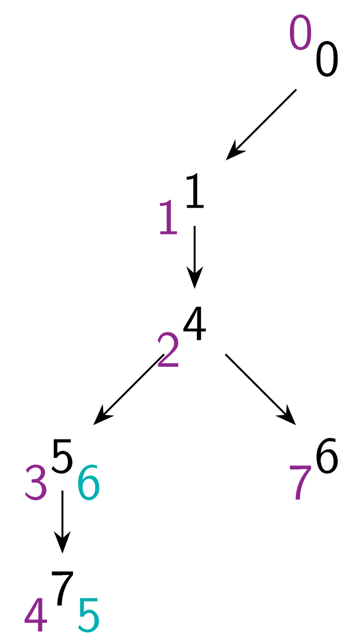




Directed cycles

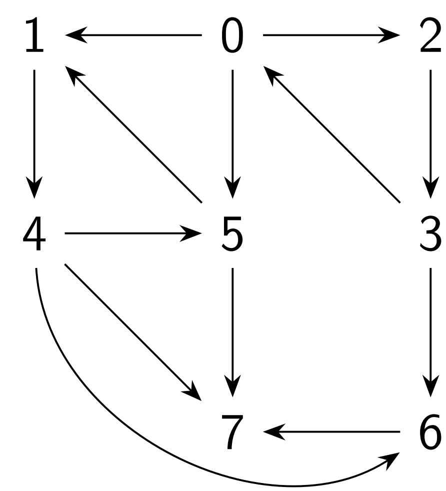
- In a directed graph, a cycle must follow same direction
  - $0 \rightarrow 2 \rightarrow 3 \rightarrow 0$ is a cycle
  - $0 \rightarrow 5 \rightarrow 1 \leftarrow 0$ is not

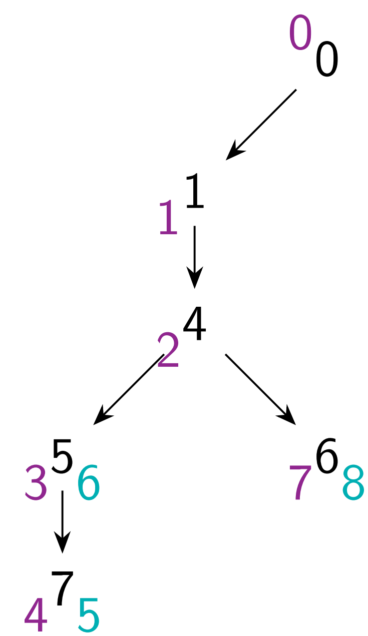




Directed cycles

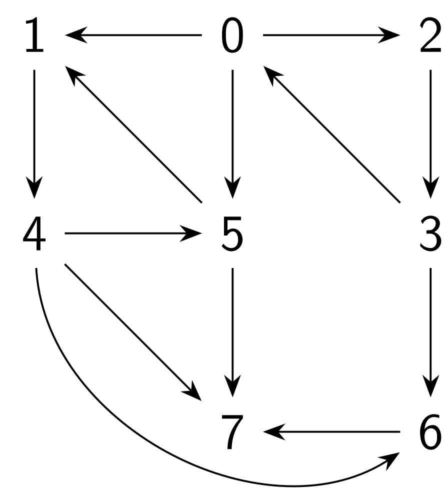
- In a directed graph, a cycle must follow same direction
  - $0 \rightarrow 2 \rightarrow 3 \rightarrow 0$ is a cycle
  - $0 \rightarrow 5 \rightarrow 1 \leftarrow 0$ is not

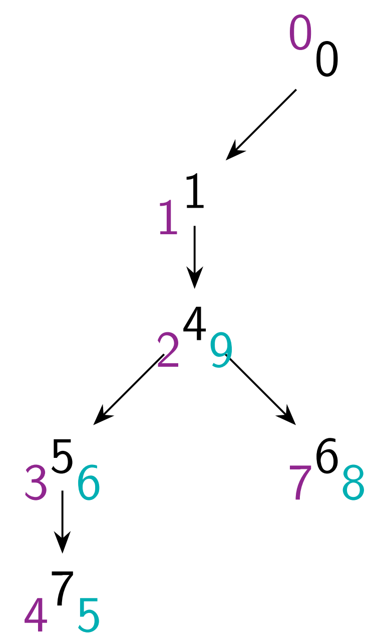




Directed cycles

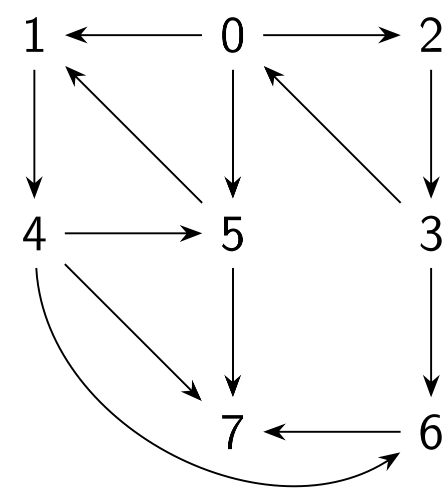
- In a directed graph, a cycle must follow same direction
  - $0 \to 2 \to 3 \to 0$ is a cycle
  - $0 \to 5 \to 1 \leftarrow 0$ is not

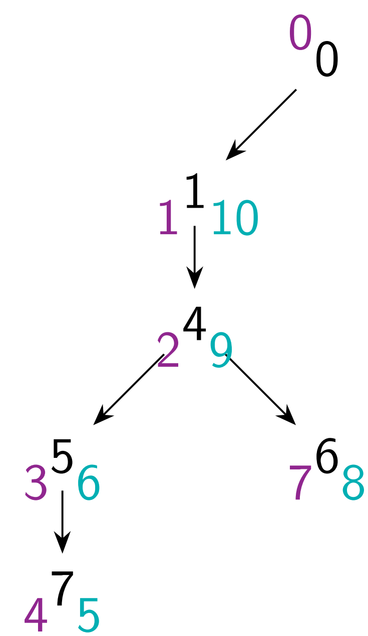




Directed cycles

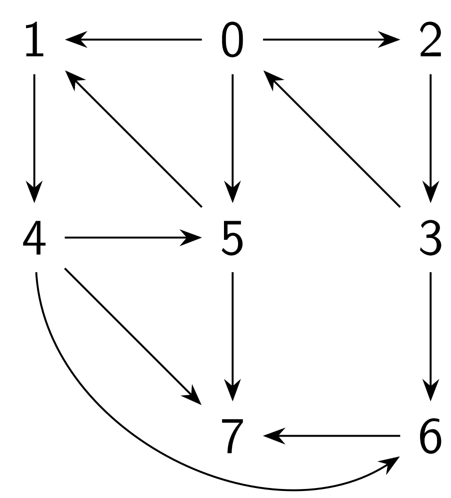
- In a directed graph, a cycle must follow same direction
  - $0 \to 2 \to 3 \to 0$ is a cycle
  - $0 \to 5 \to 1 \leftarrow 0$ is not

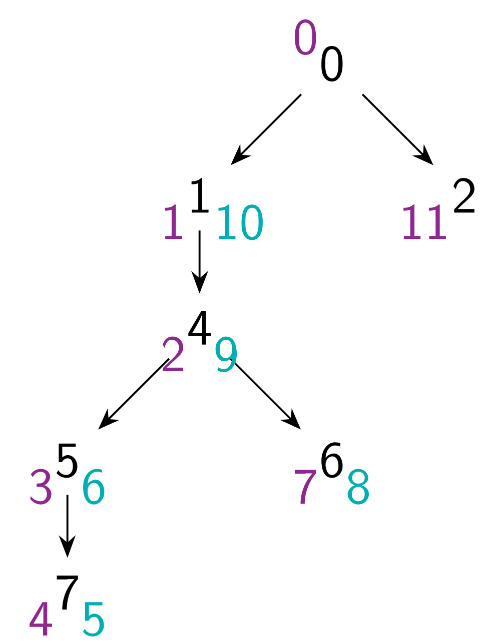




Directed cycles

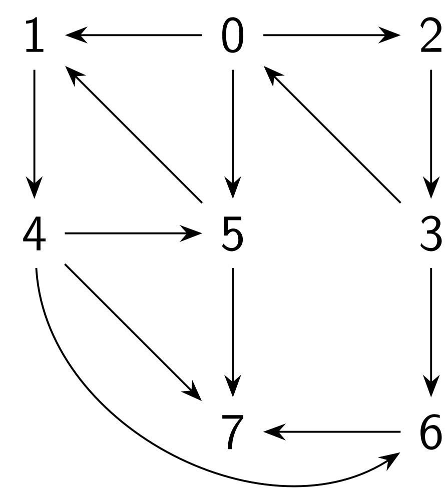
- In a directed graph, a cycle must follow same direction
  - $0 \rightarrow 2 \rightarrow 3 \rightarrow 0$ is a cycle
  - $0 \rightarrow 5 \rightarrow 1 \leftarrow 0$ is not

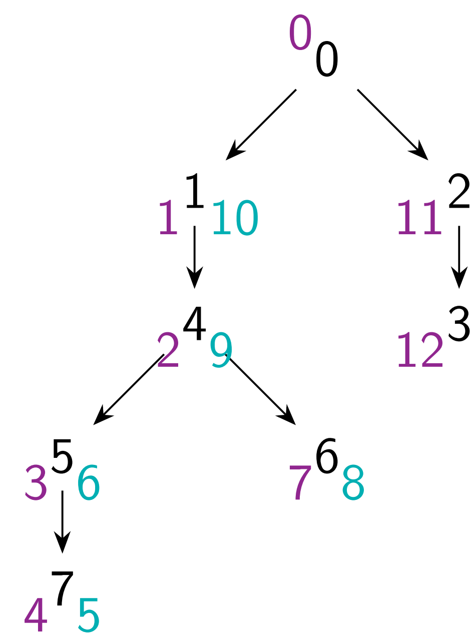




Directed cycles

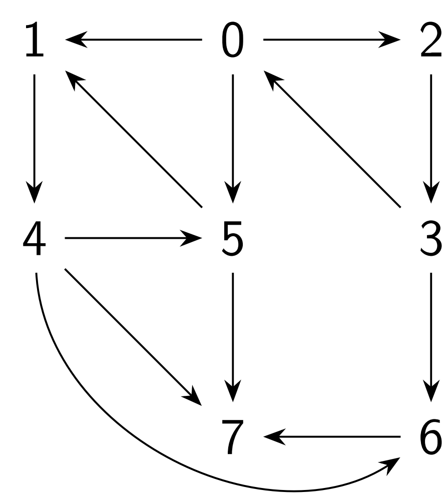
- In a directed graph, a cycle must follow same direction
  - $0 \to 2 \to 3 \to 0$ is a cycle
  - $0 \to 5 \to 1 \leftarrow 0$ is not

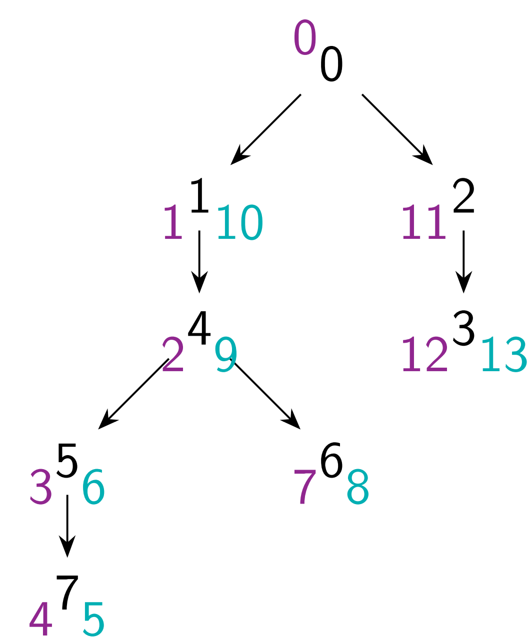




Directed cycles

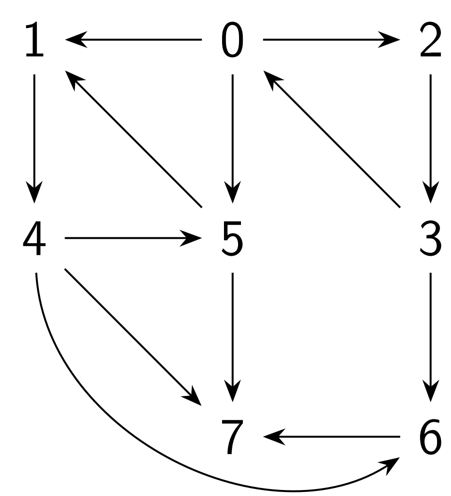
- In a directed graph, a cycle must follow same direction
  - $0 \rightarrow 2 \rightarrow 3 \rightarrow 0$ is a cycle
  - $0 \rightarrow 5 \rightarrow 1 \leftarrow 0$ is not

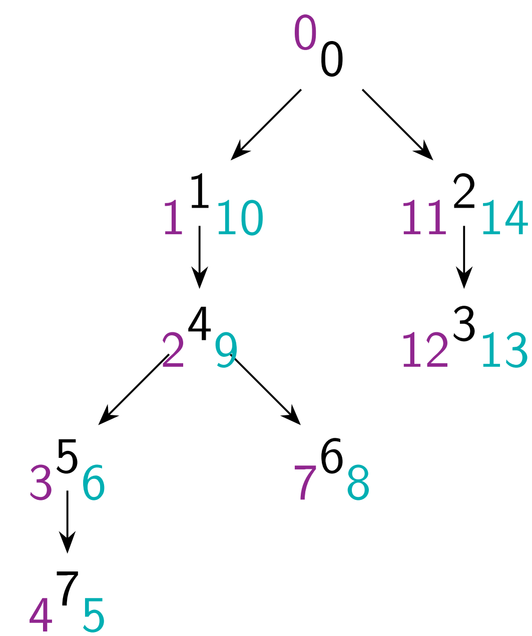




Directed cycles

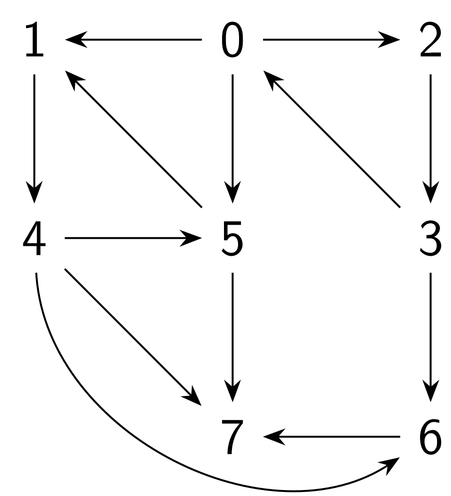
- In a directed graph, a cycle must follow same direction
  - $0 \rightarrow 2 \rightarrow 3 \rightarrow 0$ is a cycle
  - $0 \rightarrow 5 \rightarrow 1 \leftarrow 0$ is not

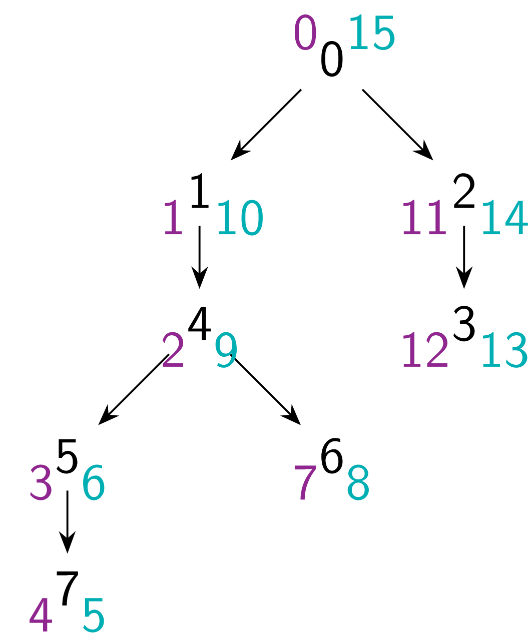




Directed cycles

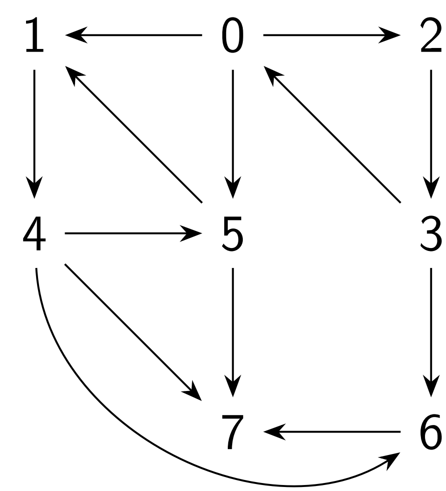
- In a directed graph, a cycle must follow same direction
  - $0 \rightarrow 2 \rightarrow 3 \rightarrow 0$ is a cycle
  - $0 \rightarrow 5 \rightarrow 1 \leftarrow 0$ is not

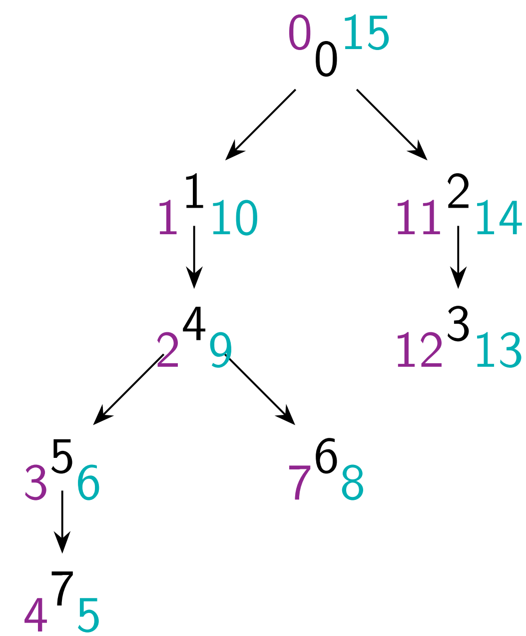




Directed cycles

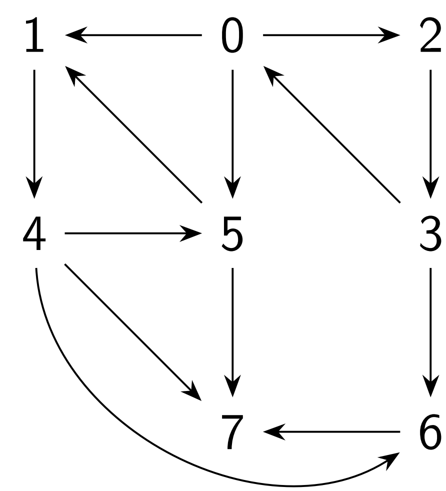
- In a directed graph, a cycle must follow same direction
  - $0 \rightarrow 2 \rightarrow 3 \rightarrow 0$ is a cycle
  - $0 \rightarrow 5 \rightarrow 1 \leftarrow 0$ is not
- Tree edges

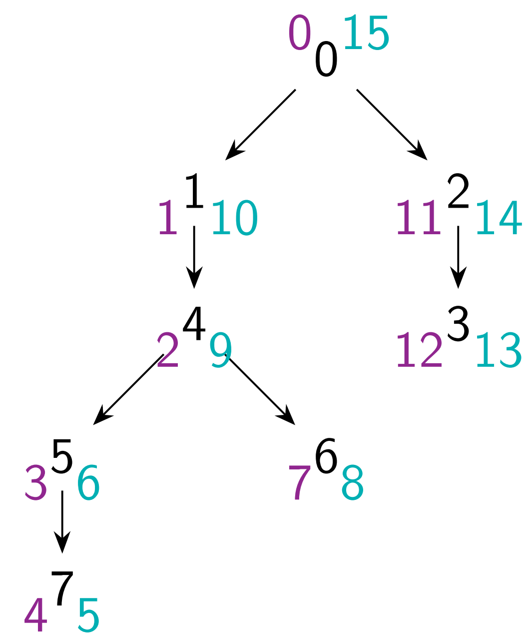




Directed cycles

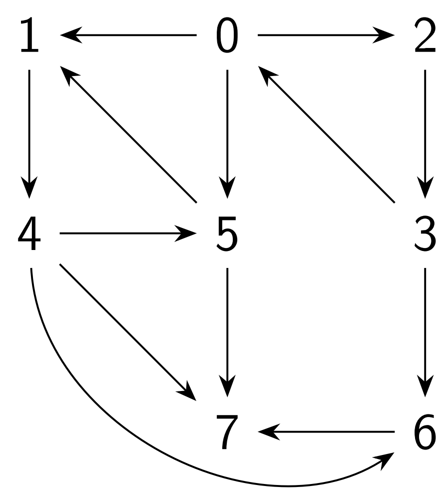
- In a directed graph, a cycle must follow same direction
  - $0 \rightarrow 2 \rightarrow 3 \rightarrow 0$ is a cycle
  - $0 \rightarrow 5 \rightarrow 1 \leftarrow 0$ is not
- Tree edges
- Different types of non-tree edges

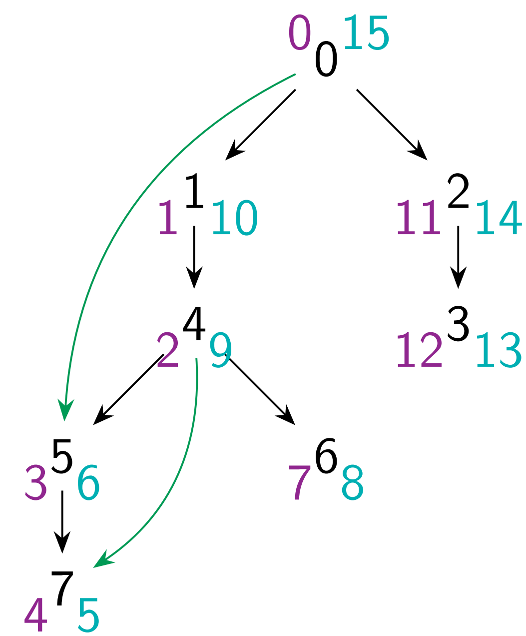




Directed cycles

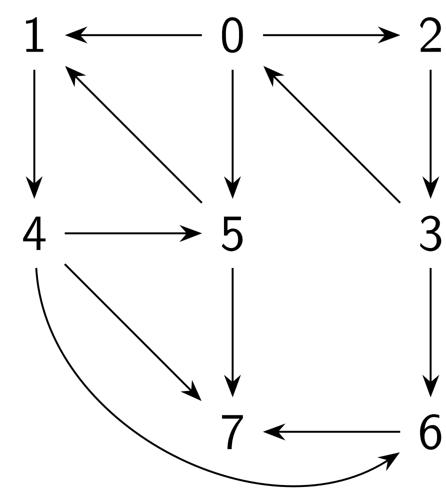
- In a directed graph, a cycle must follow same direction
  - $0 \to 2 \to 3 \to 0$ is a cycle
  - $0 \to 5 \to 1 \leftarrow 0$ is not
- Tree edges
- Different types of non-tree edges
  - Forward edges

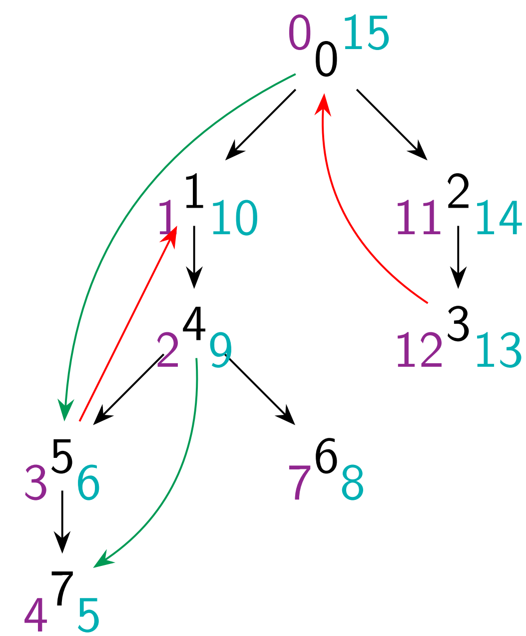




Directed cycles

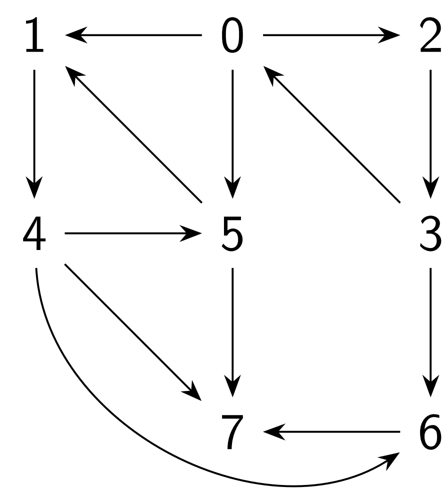
- In a directed graph, a cycle must follow same direction
  - $0 \rightarrow 2 \rightarrow 3 \rightarrow 0$ is a cycle
  - $0 \rightarrow 5 \rightarrow 1 \leftarrow 0$ is not
- Tree edges
- Different types of non-tree edges
  - Forward edges
  - Back edges

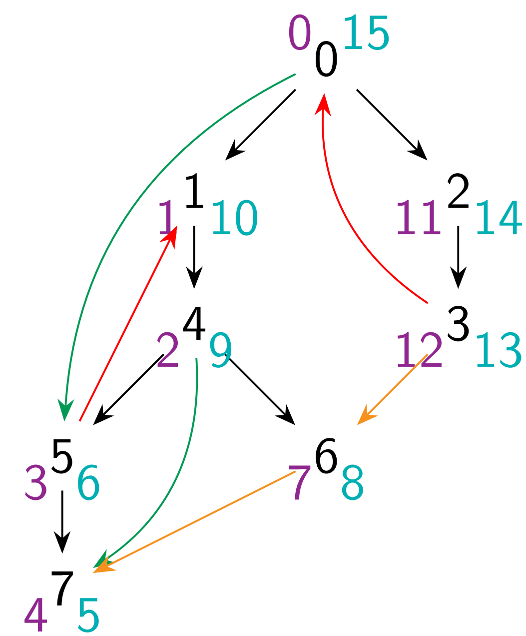




Directed cycles

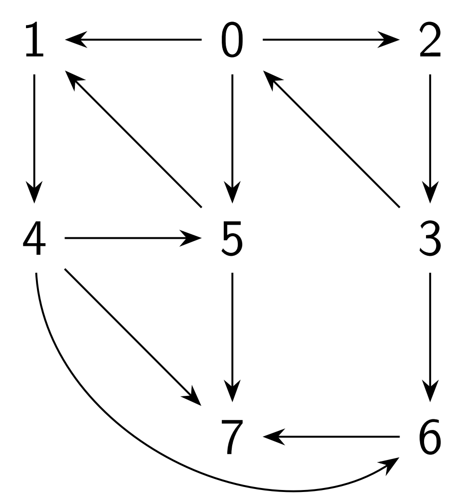
- In a directed graph, a cycle must follow same direction
  - $0 \rightarrow 2 \rightarrow 3 \rightarrow 0$ is a cycle
  - $0 \rightarrow 5 \rightarrow 1 \leftarrow 0$ is not
- Tree edges
- Different types of non-tree edges
  - Forward edges
  - Back edges
  - Cross edges

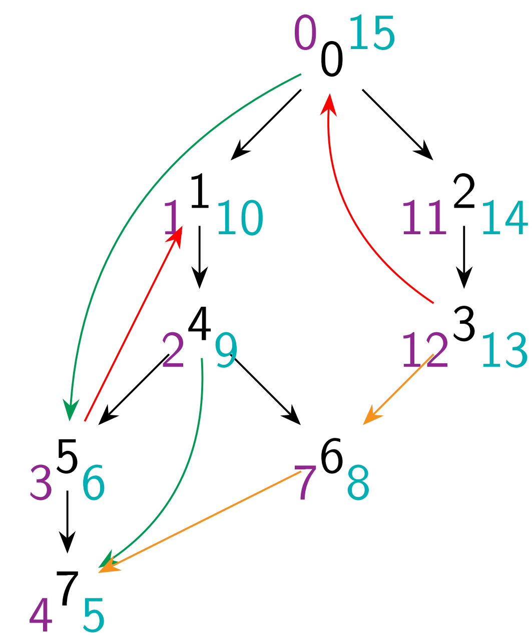




Directed cycles

- In a directed graph, a cycle must follow same direction
  - $0 \to 2 \to 3 \to 0$ is a cycle
  - $0 \to 5 \to 1 \leftarrow 0$ is not
- Tree edges
- Different types of non-tree edges
  - Forward edges
  - Back edges
  - Cross edges

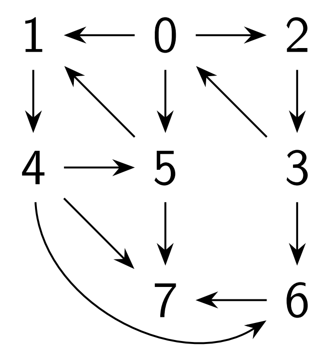


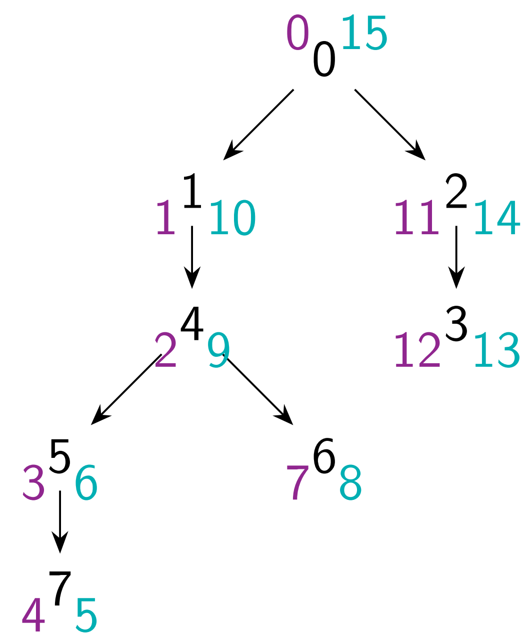


- Only back edges correspond to cycles

Classifying non-tree edges in directed graphs

- Use pre/post numbers

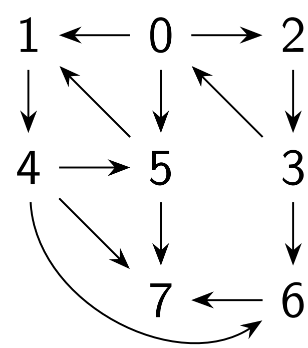


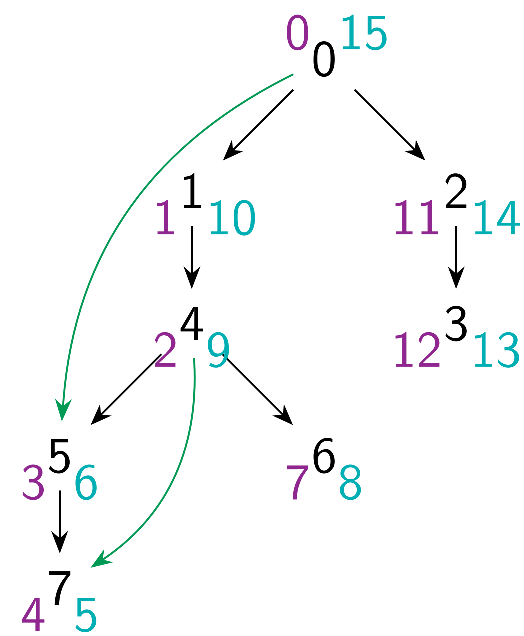


Classifying non-tree edges in directed graphs

- Use pre/post numbers
- Tree edge/forward edge (u, v)

  Interval $[\text{pre}(u), \text{post}(u)]$ contains $[\text{pre}(v), \text{post}(v)]$



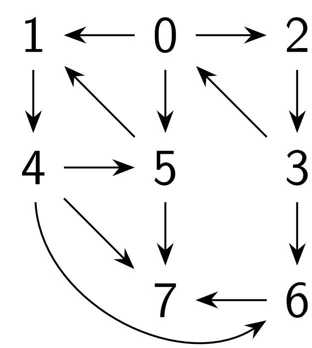


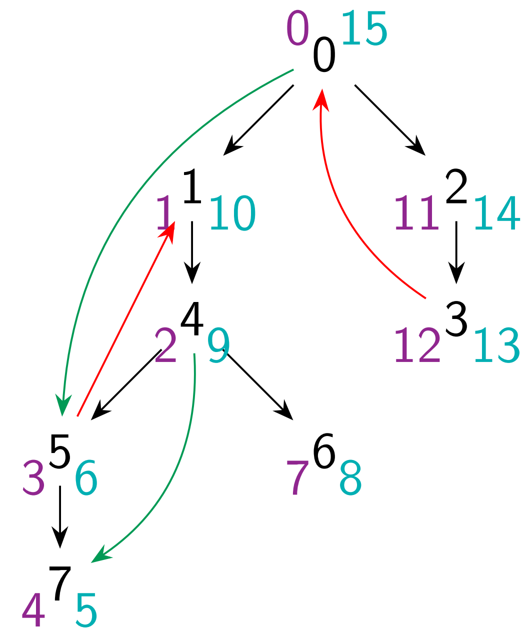
Classifying non-tree edges in directed graphs

- Use pre/post numbers
- Tree edge/forward edge (u, v)

  Interval $[\text{pre}(u), \text{post}(u)]$ contains $[\text{pre}(v), \text{post}(v)]$
- Back edge (u, v)

  Interval $[\text{pre}(v), \text{post}(v)]$ contains $[\text{pre}(u), \text{post}(u)]$





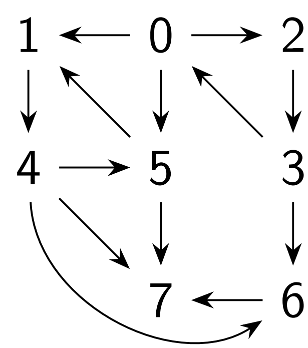
Classifying non-tree edges in directed graphs

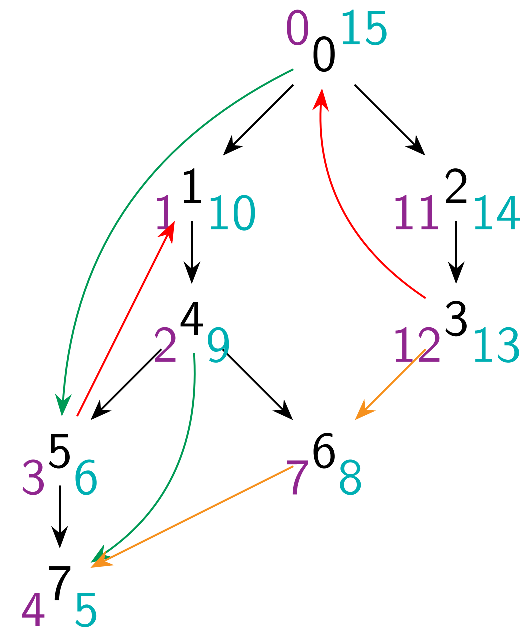
- Use pre/post numbers
- Tree edge/forward edge (u, v)

  Interval $[\mathrm{pre}(u), \mathrm{post}(u)]$ contains $[\mathrm{pre}(v), \mathrm{post}(v)]$
- Back edge (u, v)

  Interval $[\mathrm{pre}(v), \mathrm{post}(v)]$ contains $[\mathrm{pre}(u), \mathrm{post}(u)]$
- Cross edge (u, v)

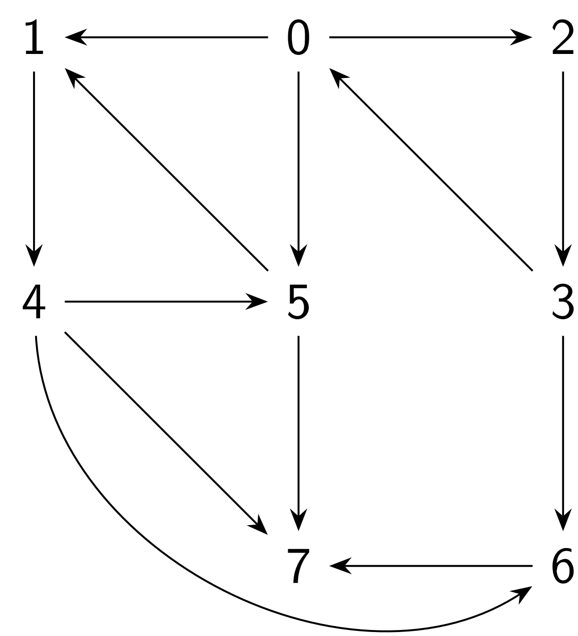
  Intervals $[\mathrm{pre}(u), \mathrm{post}(u)]$ and $[\mathrm{pre}(v), \mathrm{post}(v)]$ are disjoint





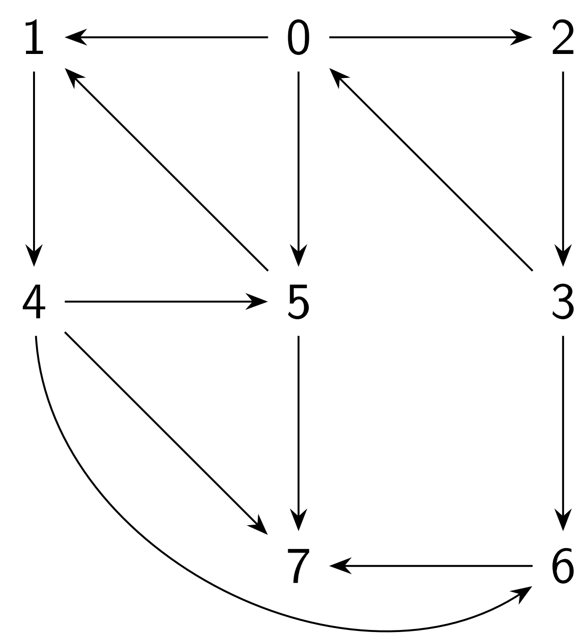
Connectivity in directed graphs

- Take directions into account



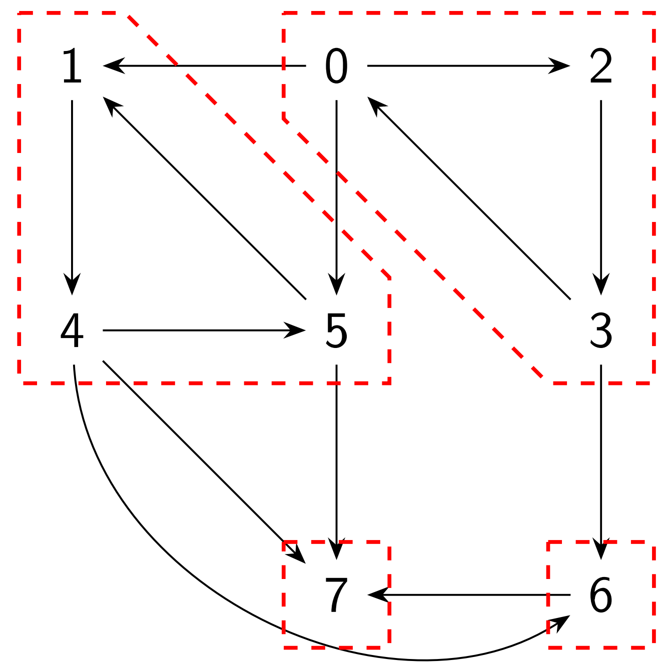
Connectivity in directed graphs

- Take directions into account
- Vertices i and j are strongly connected if there is a path from i to j and a path from j to i



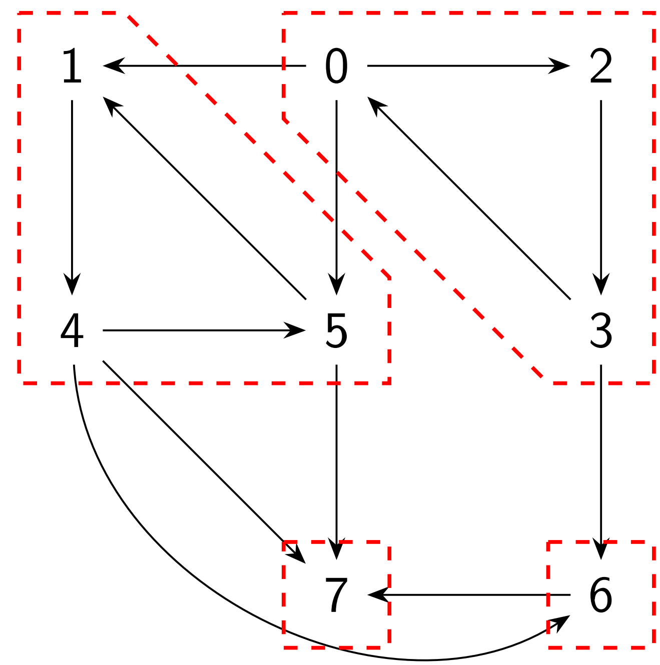
Connectivity in directed graphs

- Take directions into account
- Vertices i and j are strongly connected if there is a path from i to j and a path from j to i
- Directed graphs can be decomposed into strongly connected components (SCCs)
  - Within an SCC, each pair of vertices is strongly connected



Connectivity in directed graphs

- Take directions into account
- Vertices i and j are strongly connected if there is a path from i to j and a path from j to i
- Directed graphs can be decomposed into strongly connected components (SCCs)
  - Within an SCC, each pair of vertices is strongly connected
- DFS numbering can be used to compute SCCs



Summary

- BFS and DFS can be used to identify connected components in an undirected graph
  - BFS and DFS identify an underlying tree, non-tree edges generate cycles

Summary

- BFS and DFS can be used to identify connected components in an undirected graph
  - BFS and DFS identify an underlying tree, non-tree edges generate cycles
- In a directed graph, non-tree edges can be forward / back / cross
  - Only back edges generate cycles
  - Classify non-tree edges using DFS numbering

Summary

- BFS and DFS can be used to identify connected components in an undirected graph
  - BFS and DFS identify an underlying tree, non-tree edges generate cycles
- In a directed graph, non-tree edges can be forward / back / cross
  - Only back edges generate cycles
  - Classify non-tree edges using DFS numbering
- Directed graphs decompose into strongly connected components
  - DFS numbering can be used to compute SCC decomposition

Summary

- BFS and DFS can be used to identify connected components in an undirected graph
  - BFS and DFS identify an underlying tree, non-tree edges generate cycles
- In a directed graph, non-tree edges can be forward / back / cross
  - Only back edges generate cycles
  - Classify non-tree edges using DFS numbering
- Directed graphs decompose into strongly connected components
  - DFS numbering can be used to compute SCC decomposition
- DFS numbering can also be used to identify other features such as articulation points (cut vertices) and bridges (cut edges)

Summary

- BFS and DFS can be used to identify connected components in an undirected graph
  - BFS and DFS identify an underlying tree, non-tree edges generate cycles
- In a directed graph, non-tree edges can be forward / back / cross
  - Only back edges generate cycles
  - Classify non-tree edges using DFS numbering
- Directed graphs decompose into strongly connected components
  - DFS numbering can be used to compute SCC decomposition
- DFS numbering can also be used to identify other features such as articulation points (cut vertices) and bridges (cut edges)
- Directed acyclic graphs are useful for representing dependencies
  - Given course prerequisites, find a valid sequence to complete a programme