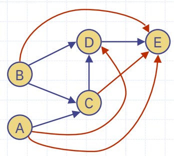
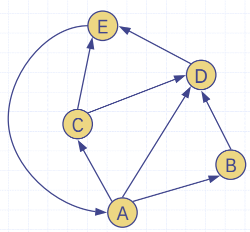
Digraphs



Outline and Reading

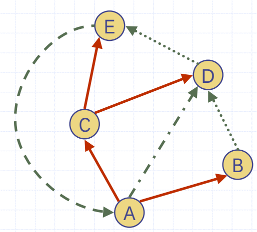
Digraphs

- A digraph is a directed graph whose edges are all directed
- Applications
  - one-way streets
  - flights
  - task scheduling



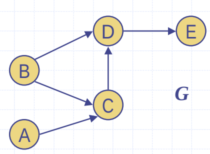
Directed DFS

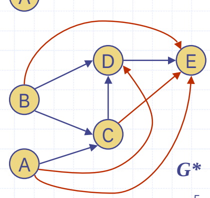
- We can specialize the traversal algorithms (DFS and BFS) to digraphs by traversing edges only along their direction
- In the directed DFS algorithm, we have four types of edges
  - discovery edges
  - back edges
  - forward edges
  - cross edges
- A directed DFS starting a a vertex s determines the vertices reachable from s



Transitive Closure

- Given a digraph G, the transitive closure of G is the digraph G^* such that
  - G^* has the same vertices as G
  - if G has a directed path from u to v ($u \neq v$), G^* has a directed edge from u to v
- The transitive closure provides reachability information about a digraph
- We can compute the transitive closure in time $O(n(n+m))$ by repeated applications of directed DFS



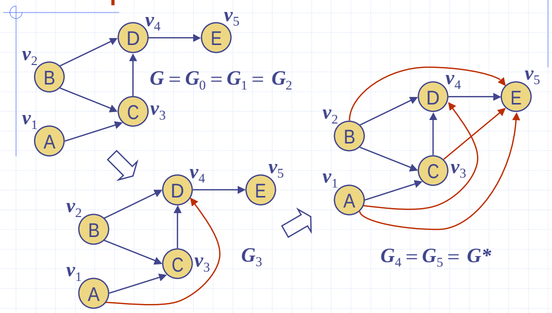


Floyd-Warshall's Algorithm

- Floyd-Warshall's algorithm numbers the vertices of a digraph G as $v_1, \ldots, v_n$ and computes a series of digraphs $G_0, \ldots, G_n$
  - $G_0 = G$
  - G_k has a directed edge (v_i, v_j) if G has a directed path from v_i to v_j with intermediate vertices in the set $\{v_1, \ldots, v_k\}$
- We have that $G_n = G^*$
- In phase k, digraph G_k is computed from G_{k-1}

```
Algorithm FloydWarshall(G)
  Input digraph G
  Output transitive closure G* of G
  i ← 1
  for all v ∈ G.vertices()
    denote v as v_i
    i ← i + 1
  G_0 ← G
  for k ← 1 to n do
    G_k ← G_{k−1}
     for i ← 1 to n (i ≠ k) do
       for j ← 1 to n (j ≠ i, k) do
         if G_{k−1}.areAdjacent(v_i, v_k) ∧
              G_{k−1}.areAdjacent(v_k, v_j)
           if ¬G_k.areAdjacent(v_i, v_j)
              G_k.insertDirectedEdge(v_i, v_j , k)
    return G_n
```

Example



DAGs and Topological Ordering

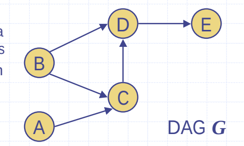
- A directed acyclic graph (DAG) is a digraph that has no directed cycles
- A topological ordering of a digraph is a numbering

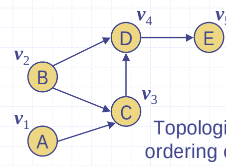
  $v_1, \ldots, v_n$

  of the vertices such that for every edge (v_i, v_j), we have $i < j$
- Example: in a task scheduling digraph, a topological ordering a task sequence that satisfies the precedence constraints

Theorem

A digraph admits a topological ordering if and only if it is a DAG





Topological ordering of G