

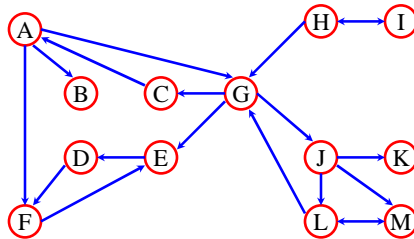
Topological Sort

CLRS Reading: Sections 22.4, pages 549 -- 552
Problem Set: Assignment #9 due Tuesday, May 6

V - 1

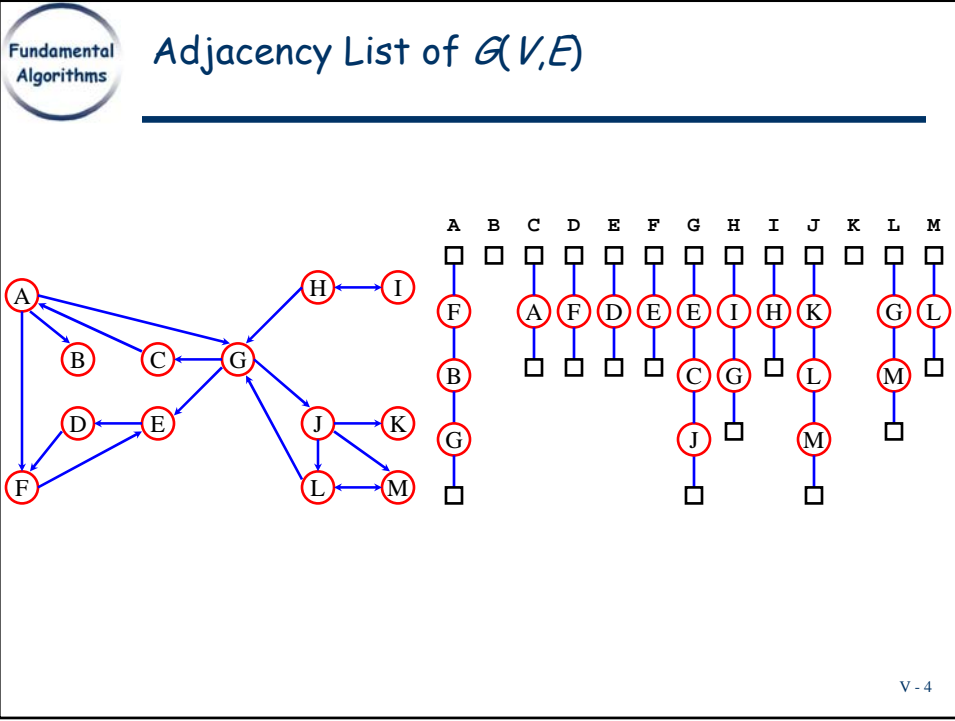
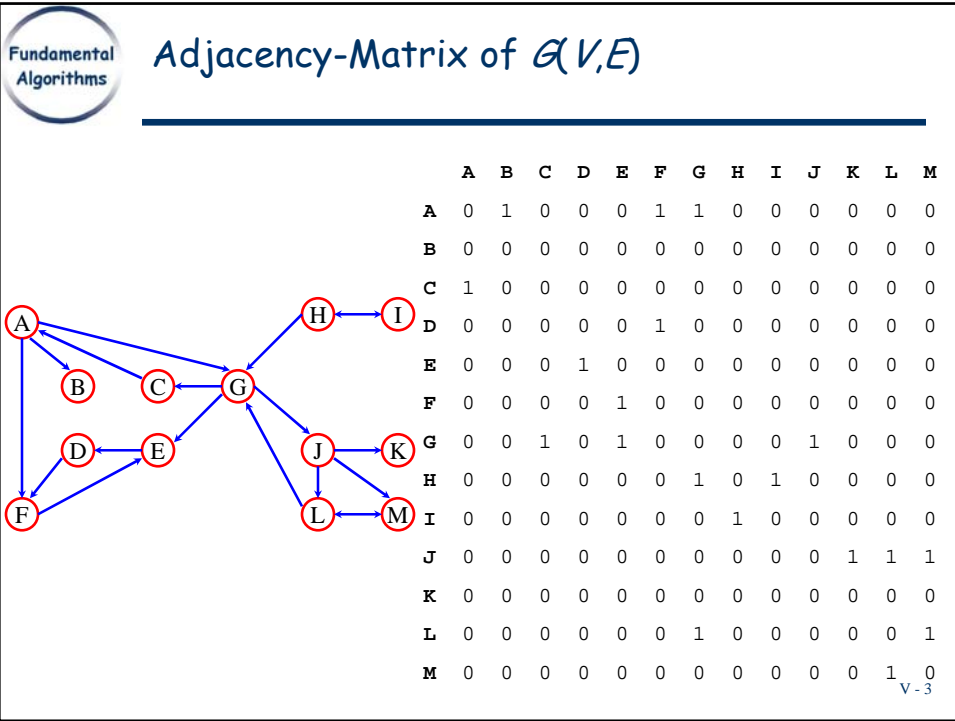
Digraphs

- *Directed graphs, digraphs, are graphs in which edges connecting nodes are one-way.*



- Directed graphs may be represented using either an adjacency matrix or an adjacency list.

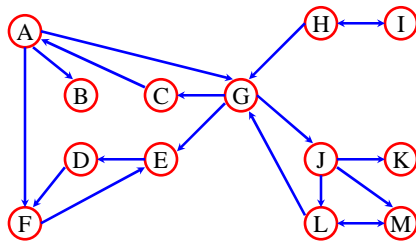
V - 2



Setting Up Depth-First Search

```

DFS(G)
  for each u ∈ V[G]
    do color[u] ← WHITE
       parent[u] ← NIL
  time ← 0
  for each u ∈ V[G]
    do if color[u] = WHITE
       then DFS-Visit[u]
  
```

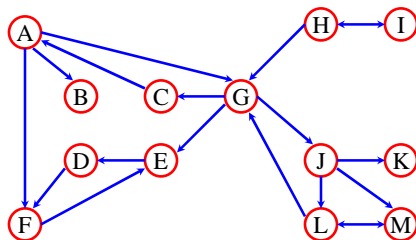


V-5

DFS-Visit(u)

```

DFS-Visit(u)
  color[u] ← GRAY
  d[u] ← time ← time+1
  for each v ∈ Adj[u]
    do if color[v] = WHITE
       then parent[v] ← u
          DFS-Visit(v)
  color[u] ← BLACK
  f[u] ← time ← time+1
  
```



V-6

Fundamental Algorithms

Depth-First Search Forest*

Tree edge

Back edge

Cross edge

Forward edge

*Which edges betray a directed cycle?

V - 7

Fundamental Algorithms

Professor Bumstead Gets Dressed*

undershorts

socks

watch

pants

shoes

belt

shirt

tie

jacket

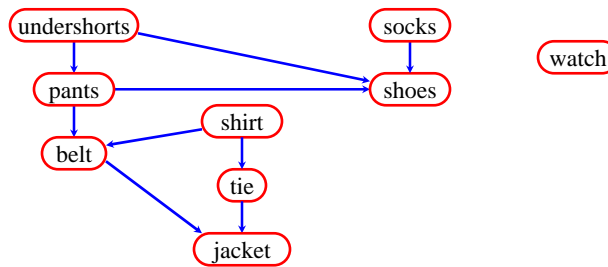
*The good Professor wishes to order the vertices so that no vertex comes before any vertex that points to it.

V - 8



Topological Sort

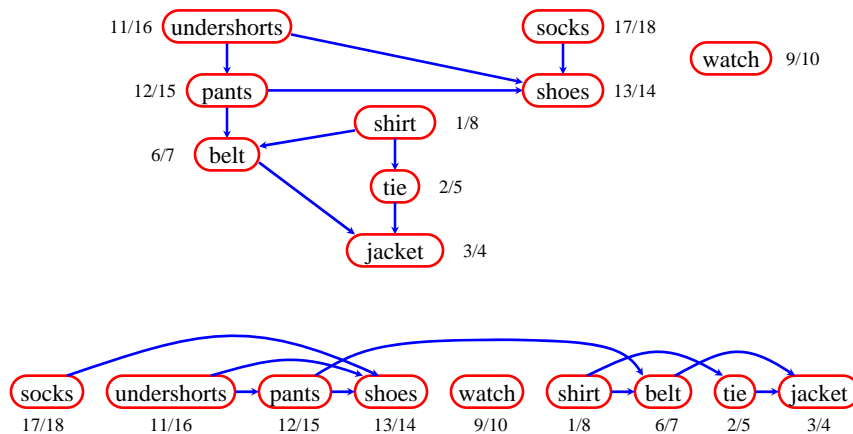
```
Topological-Sort( $G$ )  
  call DFS( $G$ ) to compute finishing times  $f[v]$   
  for each vertex  $v$   
  as each vertex is finished, insert it onto the  
  front of the linked list  
  return the linked list of vertices.
```



V - 9



All Dressed Up*



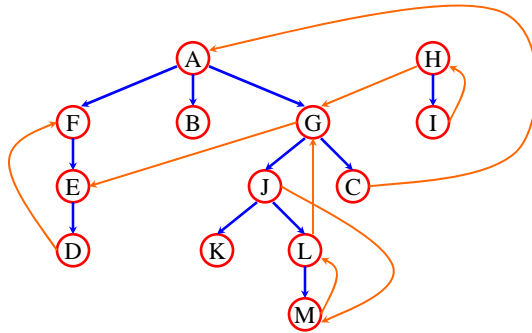
*In order for the Professor's algorithm to work, what must be true about the digraph?

V - 10

Recognizing DAGs

Lemma

A directed graph G is acyclic if and only if a depth-first search of G yields no back edges.



V - 11

Proof of Correctness

Theorem

Topological-Sort(G) produces a topological sort of a directed acyclic graph.

V - 12