Depth-first Search (DFS)

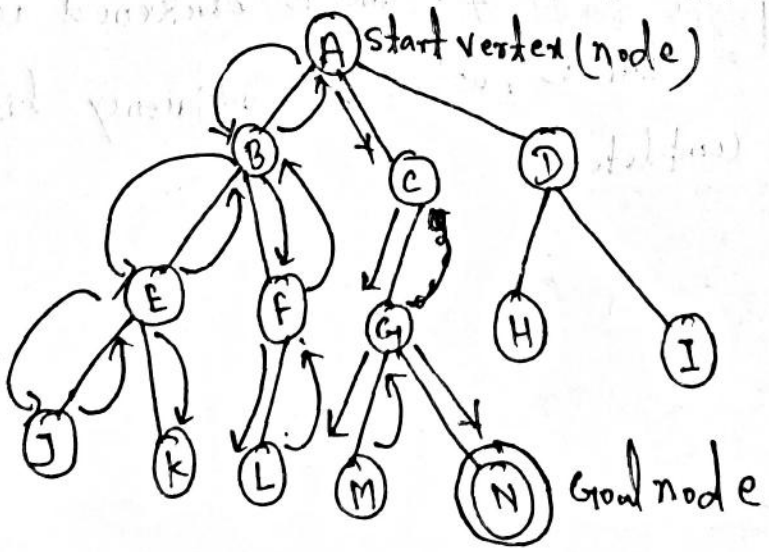
→ DFS is one of the simplest algorithm for searching a Graph.

→ The strategy followed by depth-first search is, as its name implies, to search "deeper" in the graph whenever possible.

→ In depth-first search, edges are explored out of the most recently discovered vertex v that still has unexplored edges leaving it.

→ When all of v's edges have been explored, the search "backtracks" to explore edges leaving the vertex from which v was discovered. This process continues until we have discovered all the vertices that are rechable from the original source vertex.

→ DFS algorithm use Stack data structure



→ if any undiscovered vertices remain, then one of them is selected as a new source and the search is repeated from that source. This entire process is repeated until all vertices are discovered.

→ Dfs algorithm works on both directed and undirected graph.

→ Depth-first search may search from multiple sources.

Note: → (i) The procedure DFS below records when it discovers vertex u in the variable d[u] and when it finishes vertex u in the variable f[u].
d[u] denote the: discovery time.
f[u] denote the: finishing time

(ii) Each vertex is initially white, is grayed when it is discovered in the search, and is blackened when it is finished, that is, when its adjacency list has been examined completely.

Algorithm

```
DFS(G)
1. for each vertex u ∈ V[G]
2.       do colour[u] ← white
3.          π[u] ← NIL
4  time ← 0
5  for each vertex u ∈ V[G]
6        do if colour[u] = WHITE
7                then DFS-VISIT(u)

DFS-VISIT(u)
1 colour[u] ← GRAY
2 time ← time + 1
3 d[u] ← time
4 for each v ∈ Adj[u]
5       do if colour[v] = WHITE
6             then π[v] ← u
7                  DFS-VISIT(v)
8 colour[u] ← BLACK
9 f(u) ← time ← time + 1
```

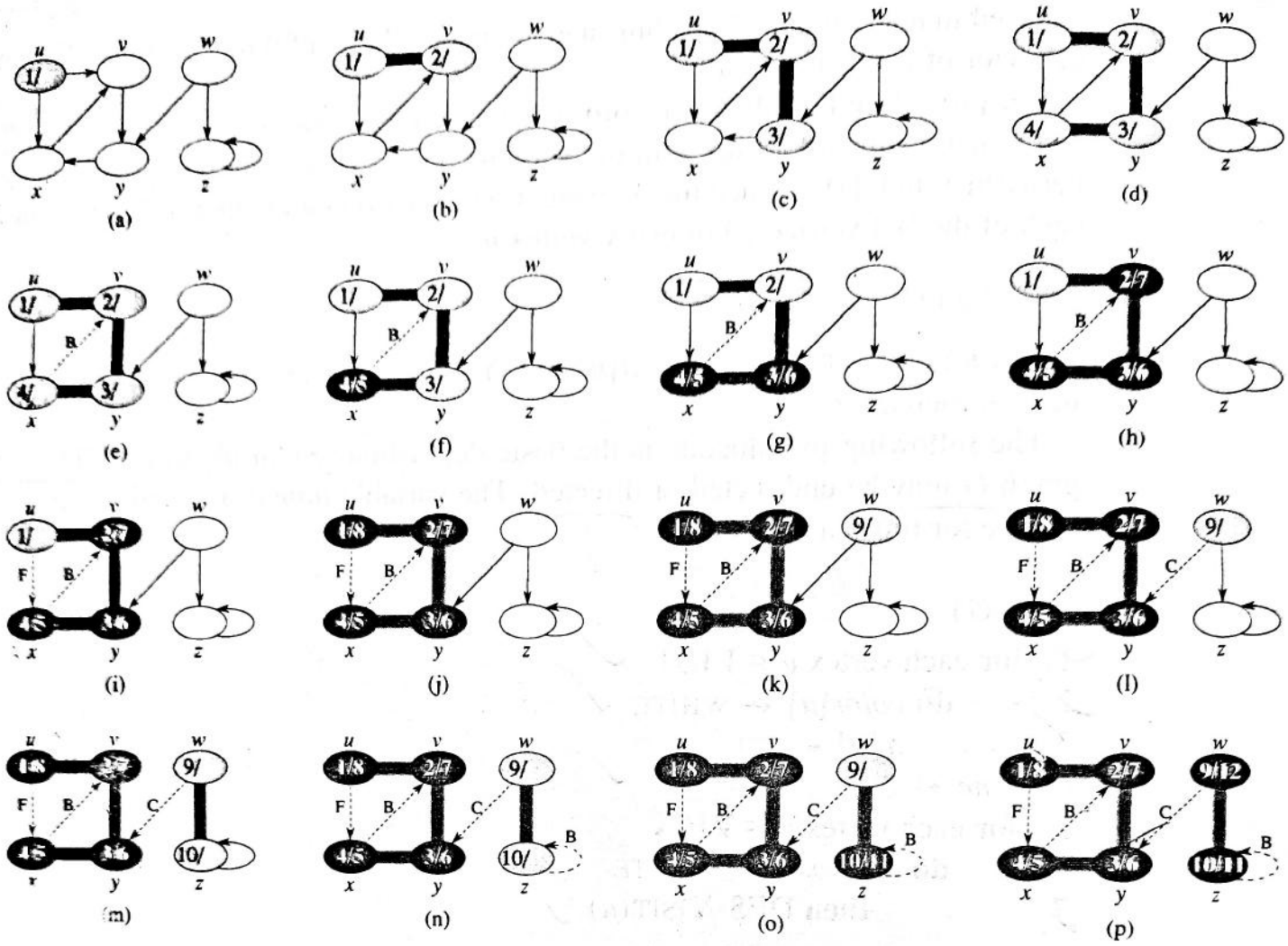


Figure 22.4 The progress of the depth-first-search algorithm DFS on a directed graph. As edges are explored by the algorithm, they are shown as either shaded (if they are tree edges) or dashed (otherwise). Nontree edges are labeled B, C, or F according to whether they are back, cross, or forward edges. Vertices are timestamped by discovery time/finishing time.

In each call DFS-VISIT(u), vertex u is initially white. Line 1 paints u gray, line 2 increments the global variable *time*, and line 3 records the new value of *time* as the discovery time $d[u]$. Lines 4–7 examine each vertex v adjacent to u and recursively visit v if it is white. As each vertex $v \in Adj[u]$ is considered in line 4, we say that edge (u, v) is ***explored*** by the depth-first search. Finally, after every edge leaving u has been explored, lines 8–9 paint u black and record the finishing time in $f[u]$.

Note that the results of depth-first search may depend upon the order in which the vertices are examined in line 5 of DFS, and upon the order in which the neighbors of a vertex are visited in line 4 of DFS-VISIT. These different visitation orders tend not to cause problems in practice, as *any* depth-first search result can usually be used effectively, with essentially equivalent results.