Data Structures & Algorithms

12: Graph Searching

Depth First Search

Dr Ram Prasad Krishnamoorthy

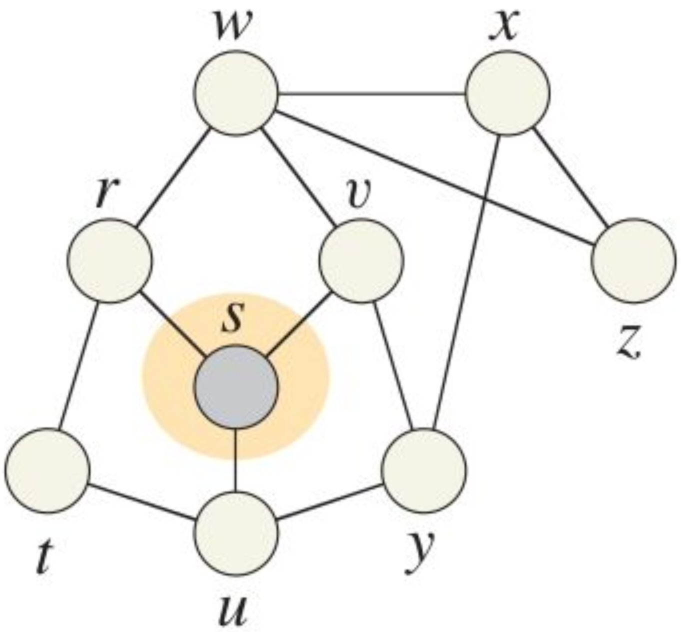
Associate Professor
School of Computing and Data Science

ram.krish@saiuniversity.edu.in

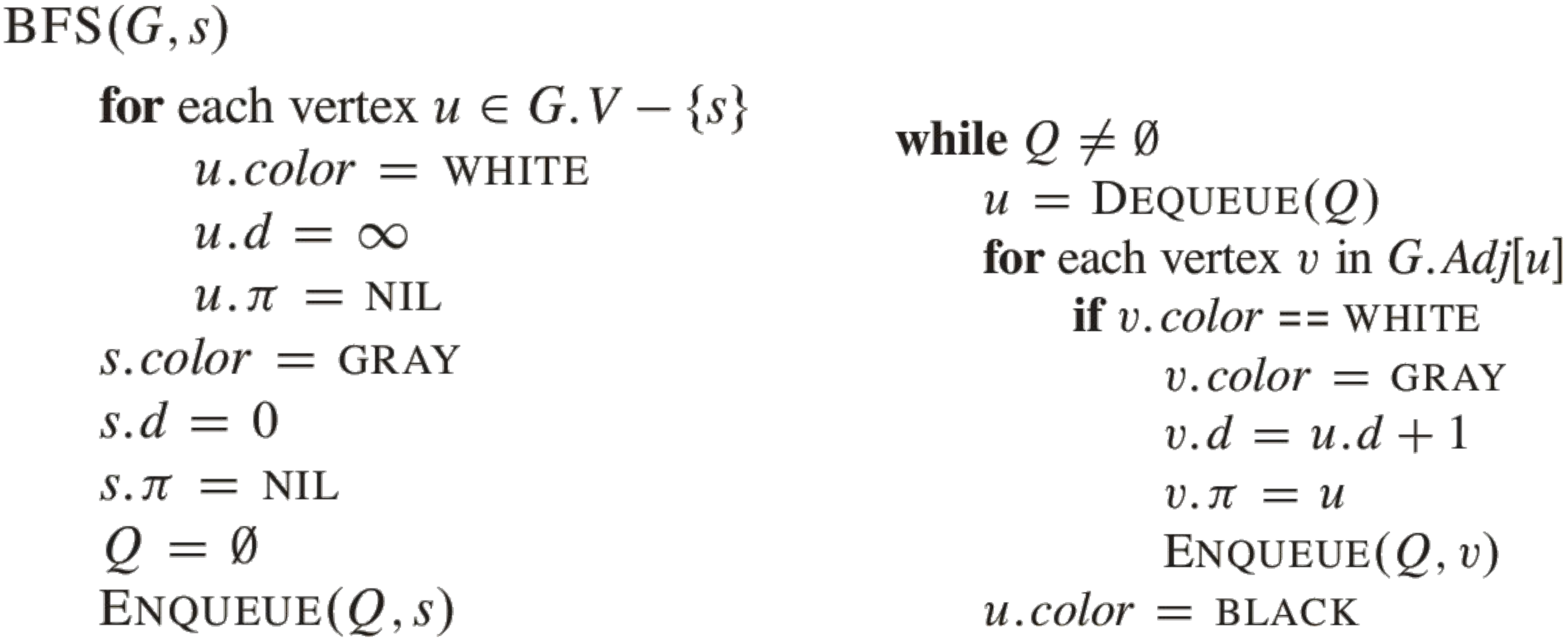
SAI UNIVERSITY

Breadth First Search (BFS)

Graph Searching - BFS

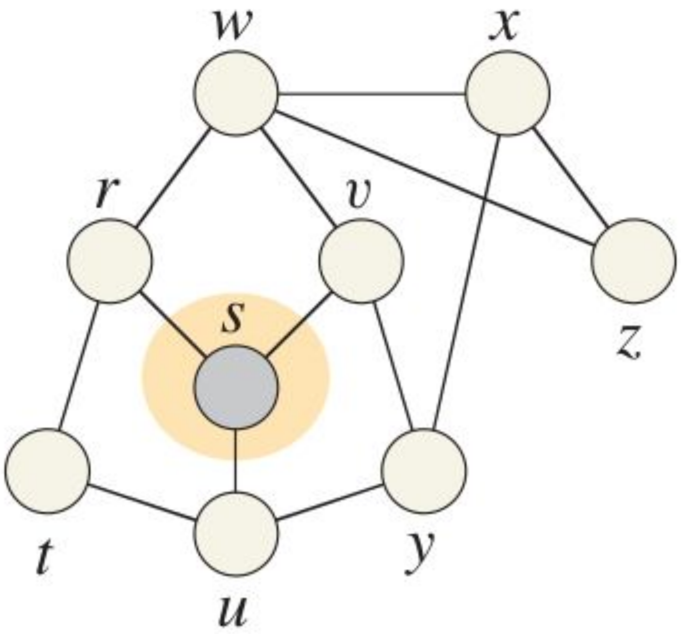


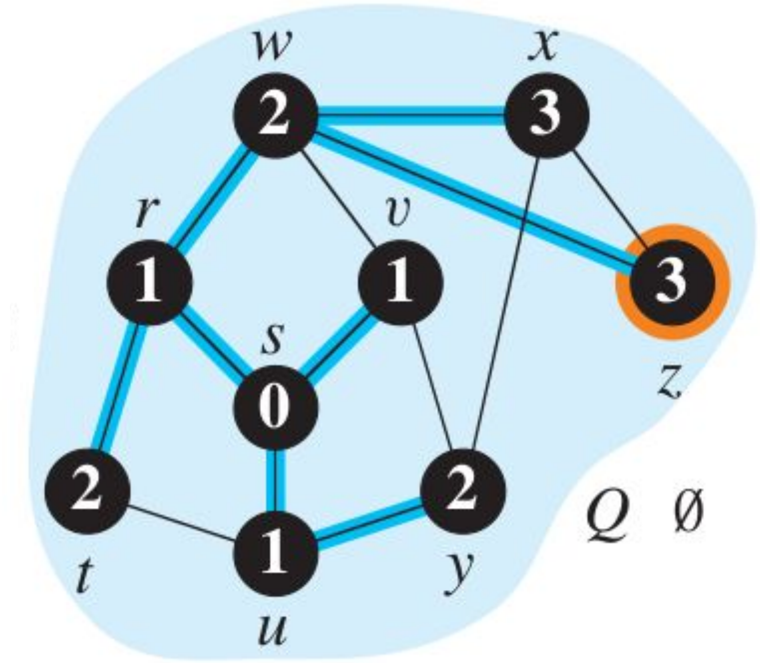
Graph Searching - BFS



```
BFS($G, s$)
    for each vertex $u \in G.V - \{s\}$
        $u.color$ = WHITE
        $u.d = \infty$
        $u.\pi$ = NIL
    $s.color$ = GRAY
    $s.d = 0$
    $s.\pi$ = NIL
    $Q = \emptyset$
    ENQUEUE($Q, s$)
    while $Q \neq \emptyset$
        $u$ = DEQUEUE($Q$)
        for each vertex $v$ in $G.Adj[u]$
            if $v.color$ == WHITE
                $v.color$ = GRAY
                $v.d = u.d + 1$
                $v.\pi = u$
                ENQUEUE($Q, v$)
        $u.color$ = BLACK
```

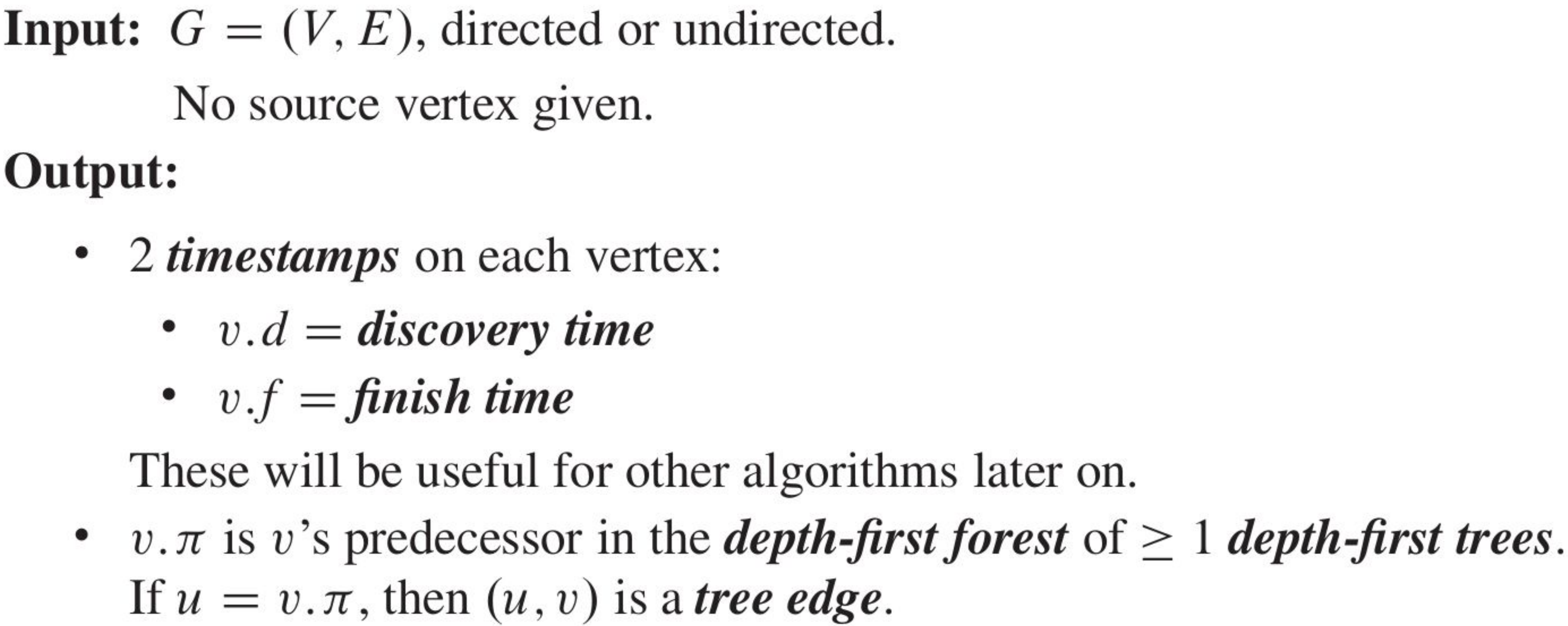
Graph Searching - BFS





Depth First Search (DFS)

Graph Searching - DFS



Input: $G = (V, E)$, directed or undirected.

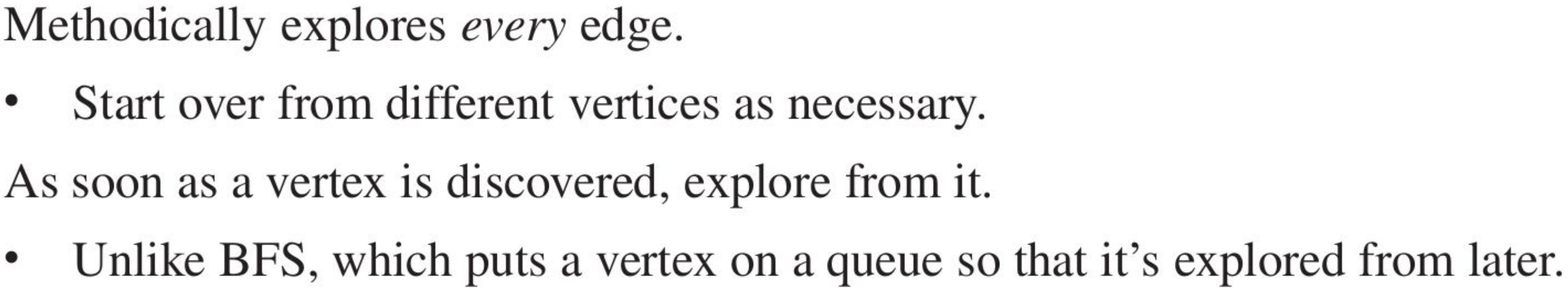
No source vertex given.

Output:

- 2 ***timestamps*** on each vertex:
  - $v.d$ = ***discovery time***
  - $v.f$ = ***finish time***

  These will be useful for other algorithms later on.
- $v.\pi$ is v's predecessor in the ***depth-first forest*** of ≥ 1 ***depth-first trees***. If $u = v.\pi$, then (u, v) is a ***tree edge***.

Graph Searching - DFS



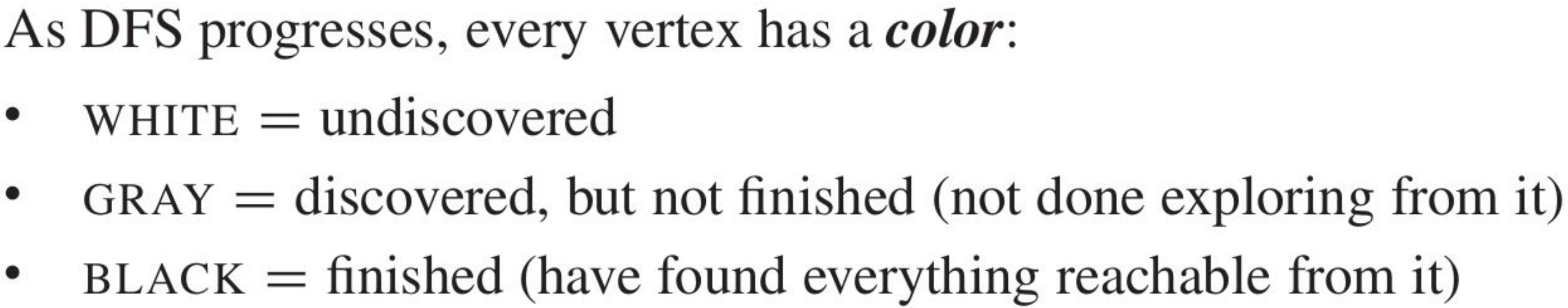
Methodically explores *every* edge.

- Start over from different vertices as necessary.

As soon as a vertex is discovered, explore from it.

- Unlike BFS, which puts a vertex on a queue so that it's explored from later.

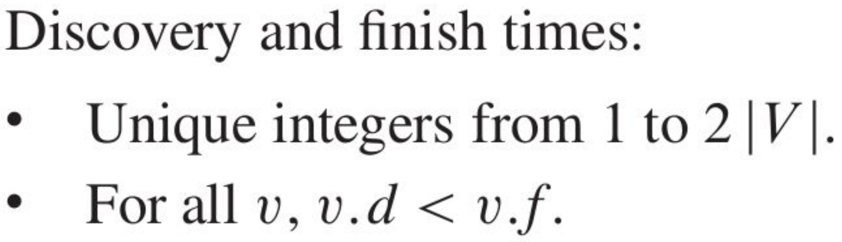
Graph Searching - DFS



As DFS progresses, every vertex has a ***color***:

- WHITE = undiscovered
- GRAY = discovered, but not finished (not done exploring from it)
- BLACK = finished (have found everything reachable from it)

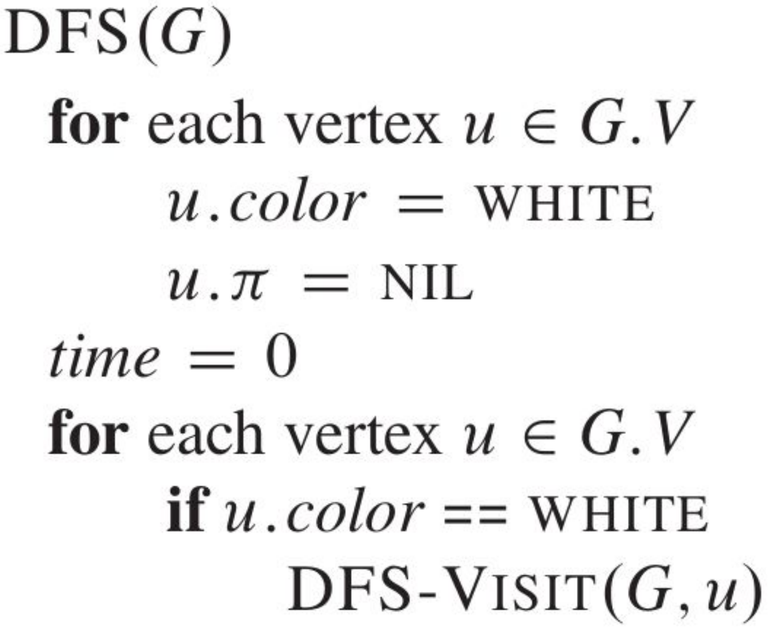
Graph Searching - DFS



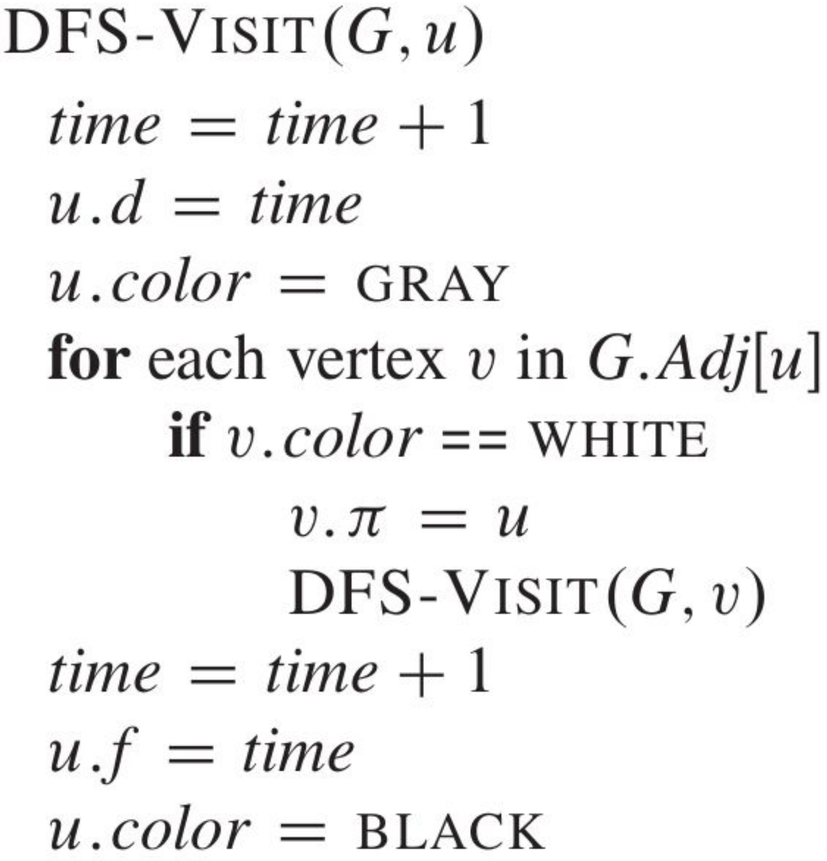
Discovery and finish times:

- Unique integers from 1 to $2|V|$.
- For all v, $v.d < v.f$.

Graph Searching - DFS

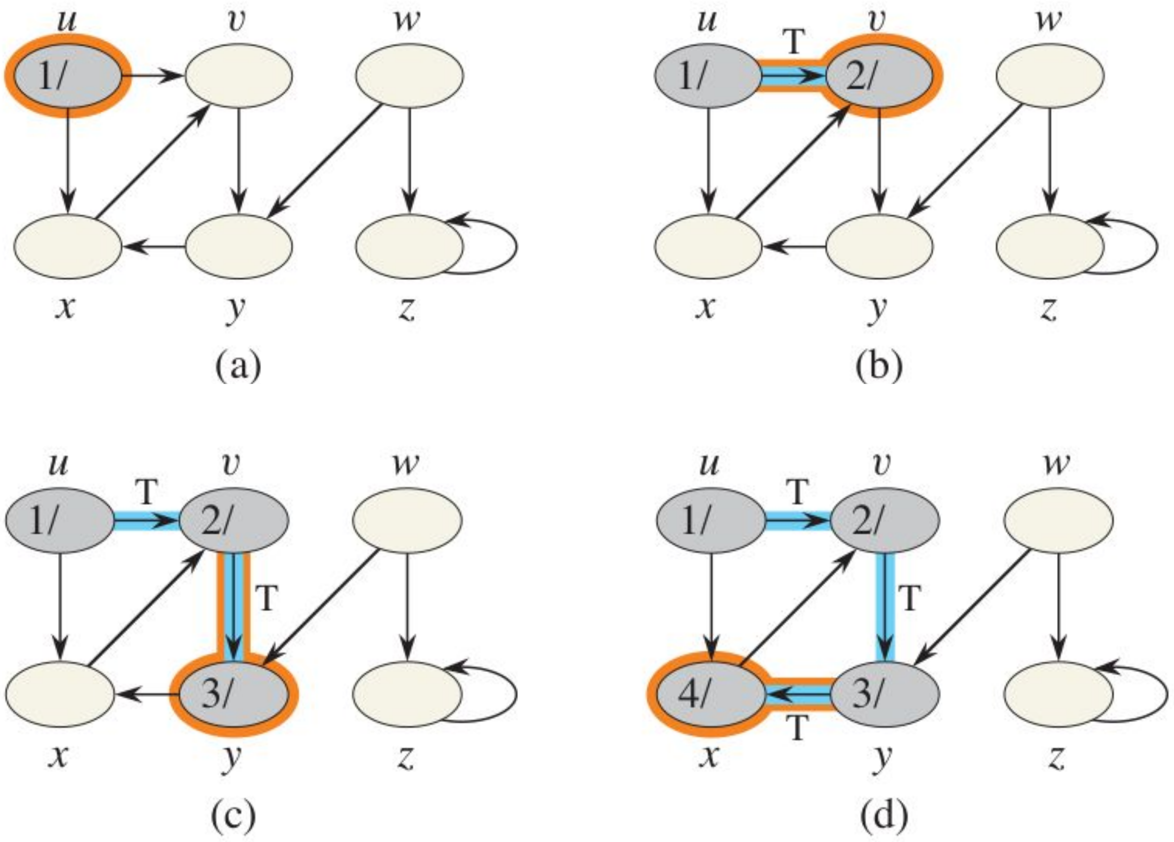


```
DFS($G$)
  for each vertex $u \in G.V$
      $u.color = $ WHITE
      $u.\pi = $ NIL
  $time = 0$
  for each vertex $u \in G.V$
      if $u.color ==$ WHITE
          DFS-VISIT($G, u$)
```

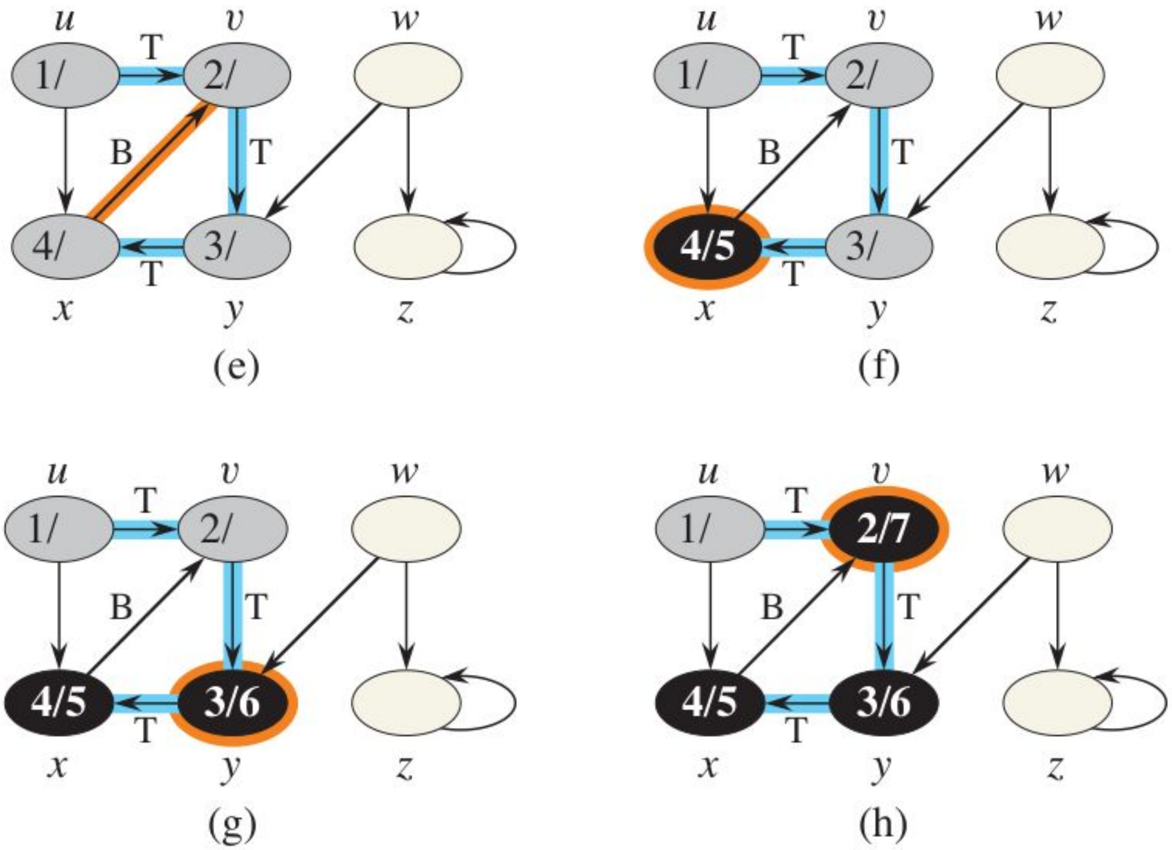


```
DFS-VISIT($G, u$)
  $time = time + 1$
  $u.d = time$
  $u.color = $ GRAY
  for each vertex $v$ in $G.Adj[u]$
      if $v.color ==$ WHITE
          $v.\pi = u$
          DFS-VISIT($G, v$)
  $time = time + 1$
  $u.f = time$
  $u.color = $ BLACK
```

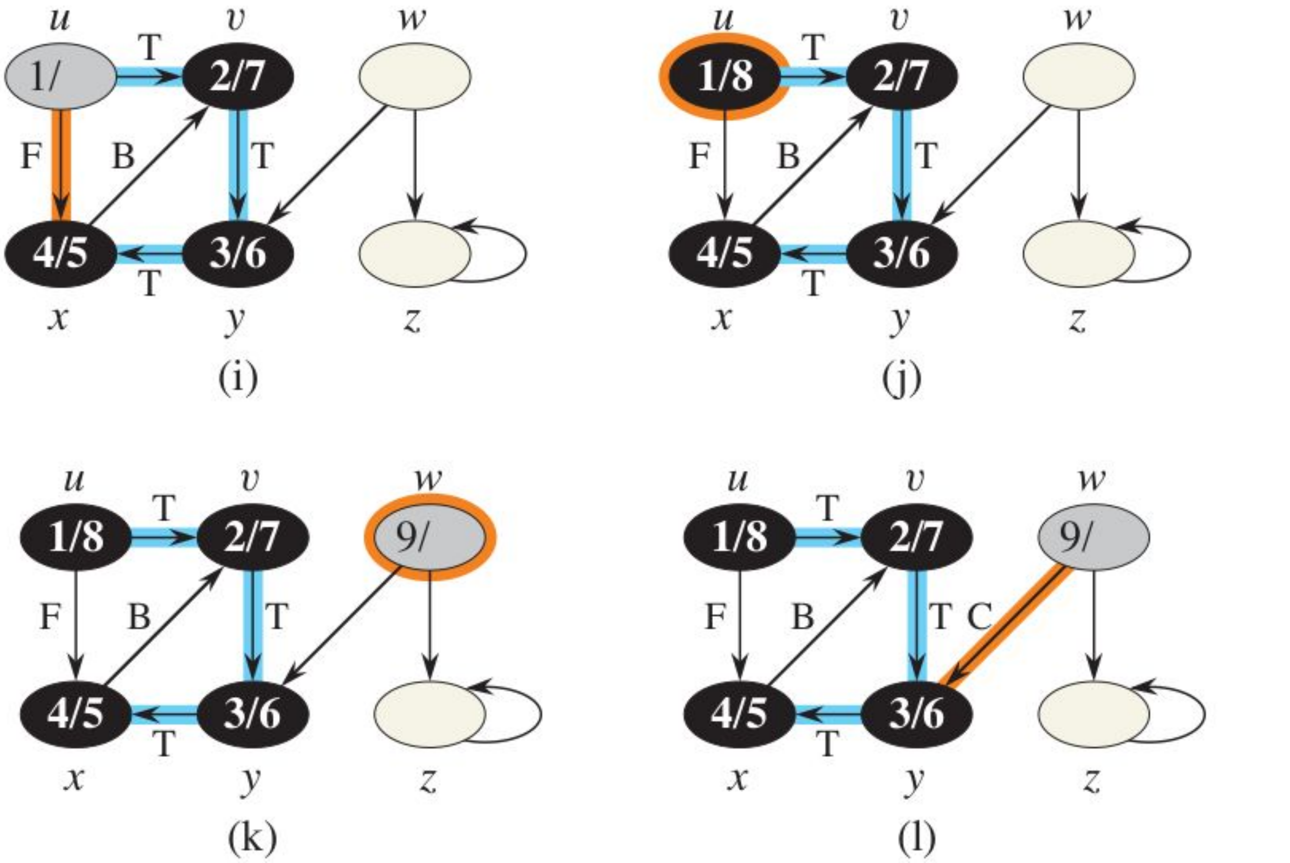
Graph Searching - DFS



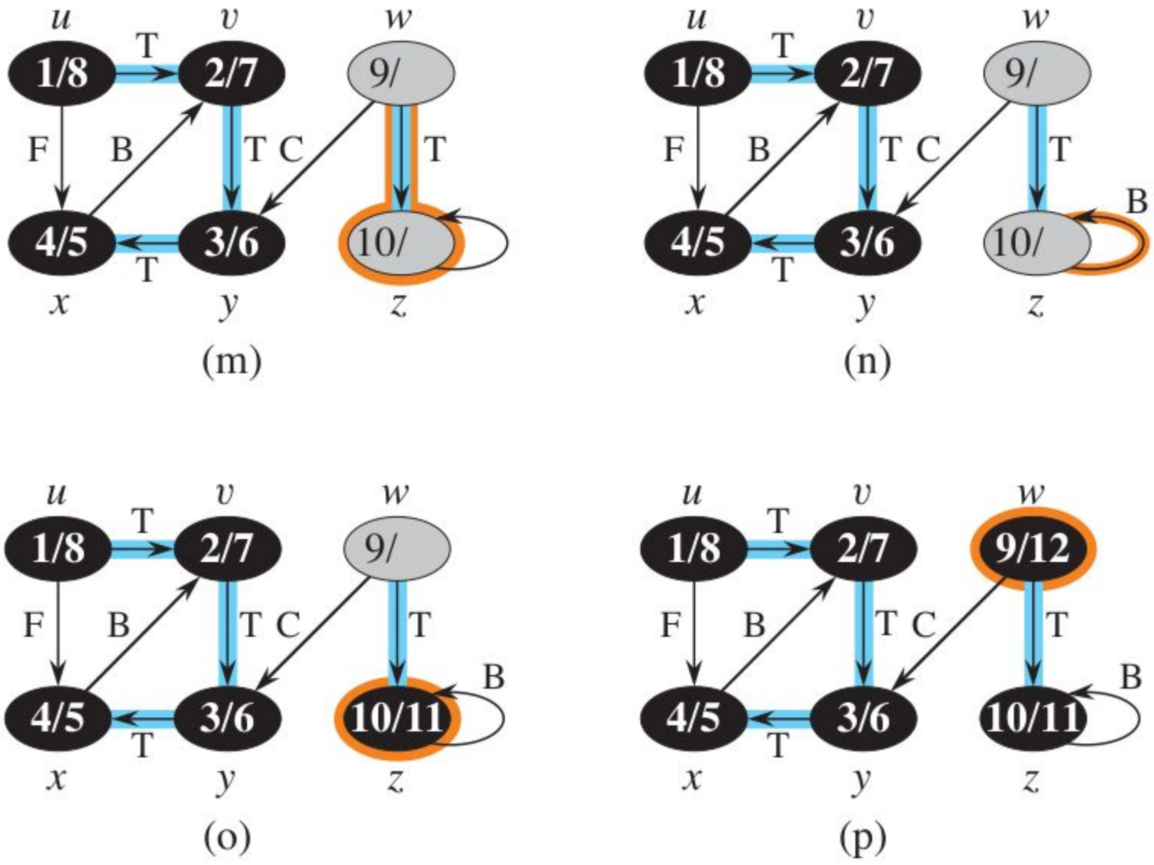
Graph Searching - DFS



Graph Searching - DFS



Graph Searching - DFS



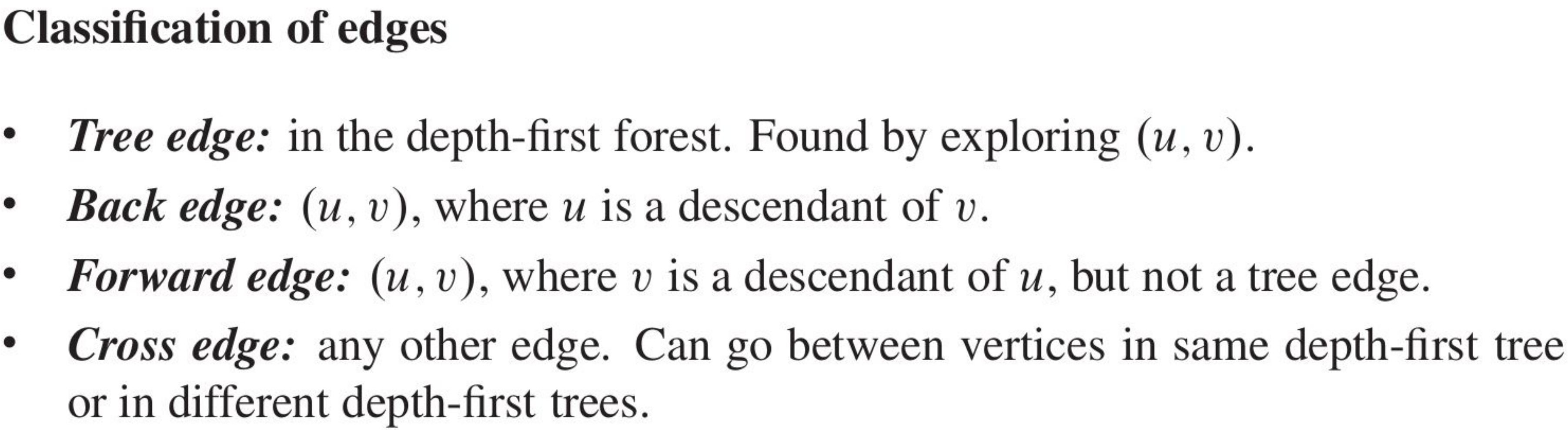
(m)

(n)

(o)

(p)

Graph Searching - DFS



Classification of edges

- ***Tree edge:*** in the depth-first forest. Found by exploring (u, v).
- ***Back edge:*** (u, v), where u is a descendant of v.
- ***Forward edge:*** (u, v), where v is a descendant of u, but not a tree edge.
- ***Cross edge:*** any other edge. Can go between vertices in same depth-first tree or in different depth-first trees.