TECHNO INDIA UNIVERSITY
WESTBENGAL

Data Structures & Algorithms
(PCC-CS 301)

Dr. Debashis Das
Associate Professor
Department of CSE
Techno India University, Kolkata

Topics Covered

1. Graph traversal algorithm
   1.1. Depth First Search (DFS)
   1.2. Applications

Graph Traversal

- **Graph traversal**

  Traversing all the vertices of a graph starting from an arbitrary vertex

  It produces the set of all vertices that are **reachable from the starting node** of the graph (BFS)

  It produces the set of **all connected vertices** that belongs to the graph (DFS)

- **Traversal algorithm**
  - Breadth First Search (BFS)
  - Depth First Search (DFS)

Graph Traversal

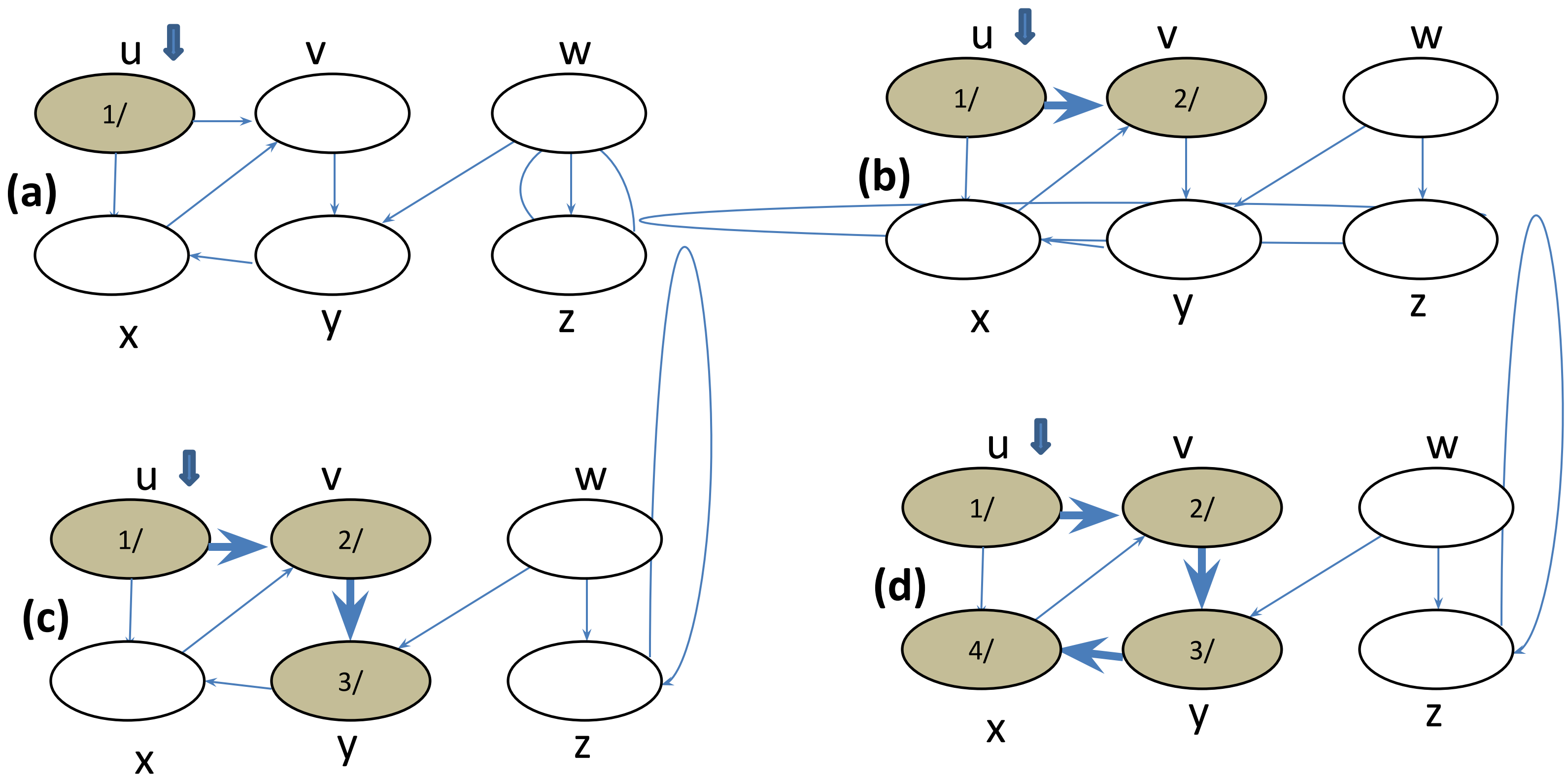
- **Depth First Search**
  - It begins from any vertex ***s*** and start traversing all the edges up to the maximum depth in a particular direction. It stops when reached to an end node i.e. from where no further movement is possible in that particular direction. If it reaches to an end, backtracks to the predecessor of the current node (one step back) and start discovering the other possible adjacent edges
  - Follow the same process until all the nodes are discovered that are reachable from the source node ***s***
  - If any node of the graph remains undiscovered, consider an undiscovered node as the new source and start the algorithm
  - The DFS search stops when all the nodes of the graph are discovered and processed

Graph Traversal

- **Depth First Search**
  - As DFS traversal moves at maximum depth along a specific path and backtrack, it needs STACK data structure to maintain the discovered nodes
  - DFS uses node coloring concept to keep track of the discovered, undiscovered and processed nodes
    - White color node: still undiscovered
    - Gray node: discovered for the first time but still not processed
    - Black node: processed and written in O/P
  - DFS also uses timestamp to every vertex on first discovering and on finish processing
    - Attribute **v.d** records the time of node ***v*** when first discovered
    - Attribute **v.f** records the time of ***v*** when processed finally

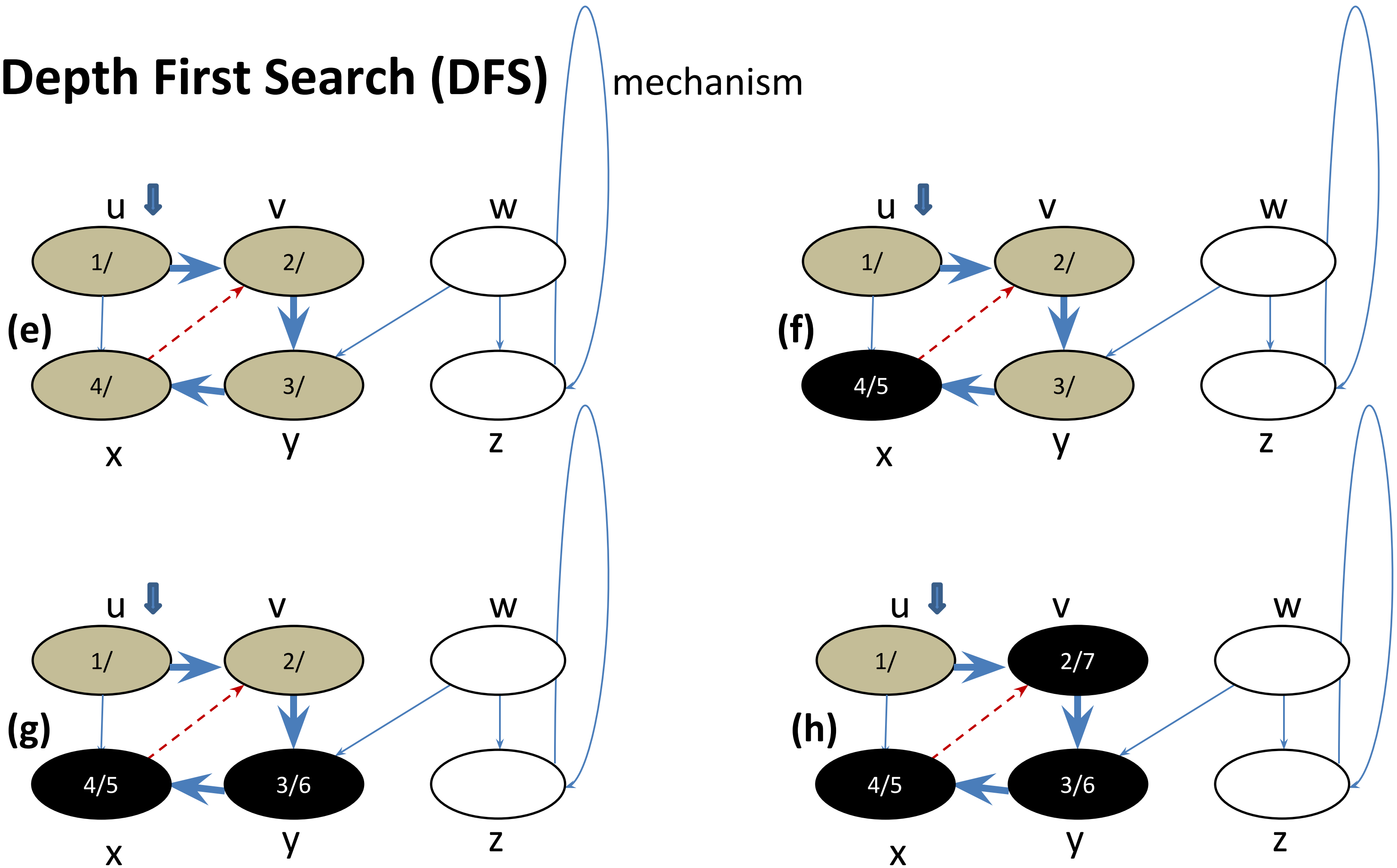
Graph Traversal

- **Depth First Search (DFS)** mechanism



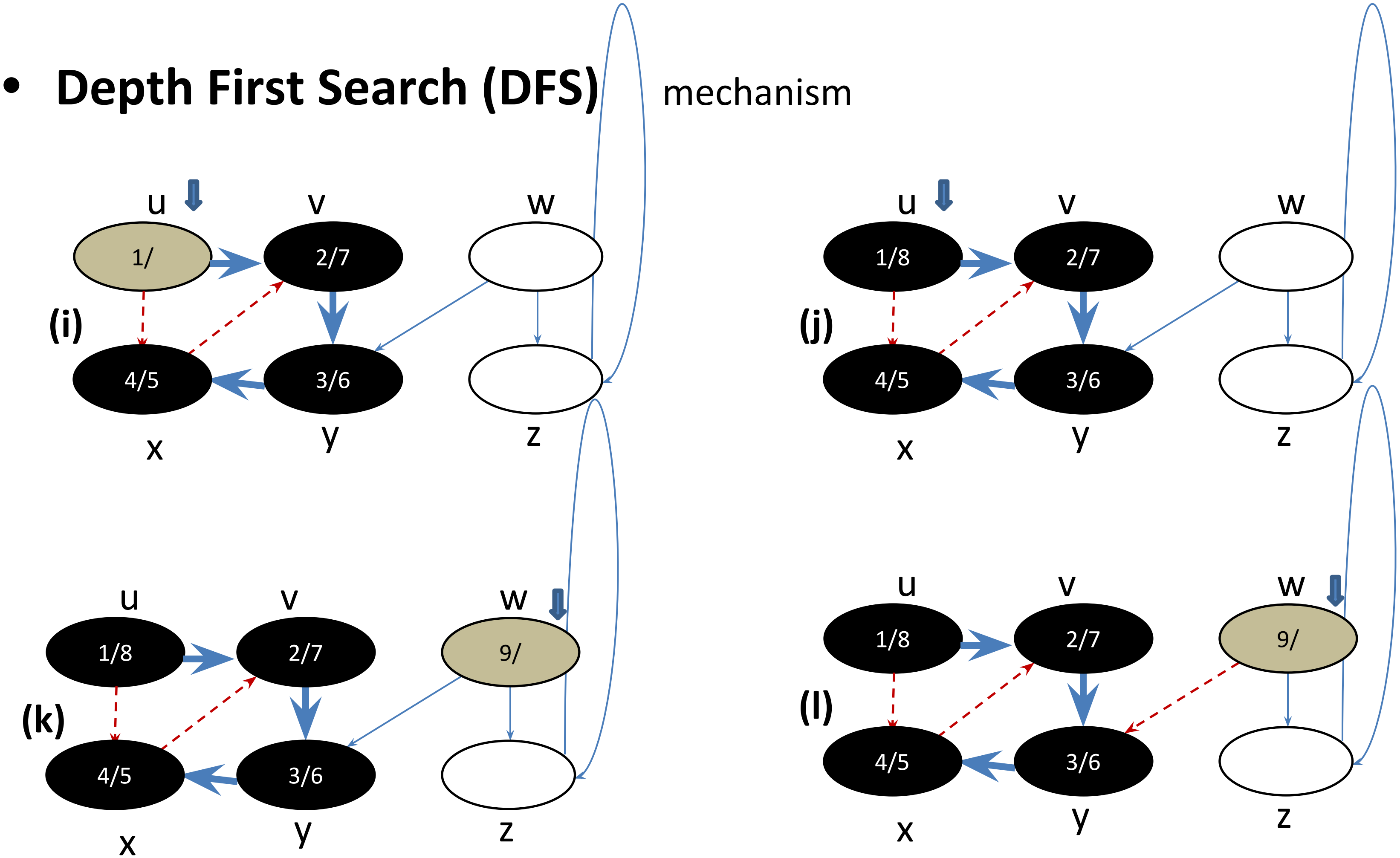
Graph Traversal

- **Depth First Search (DFS)** mechanism



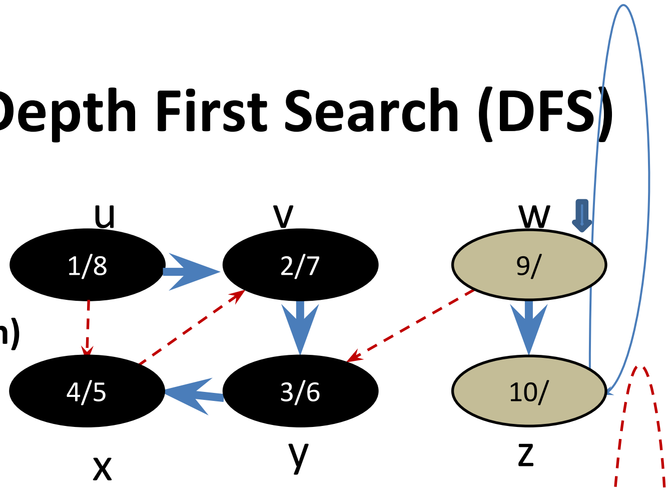
Graph Traversal

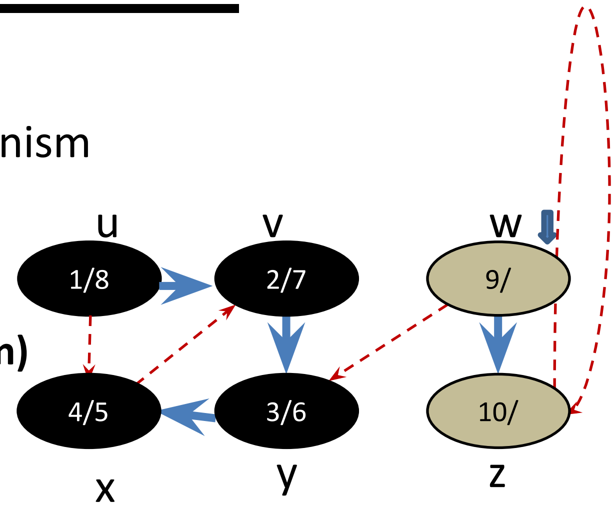
- **Depth First Search (DFS)** mechanism

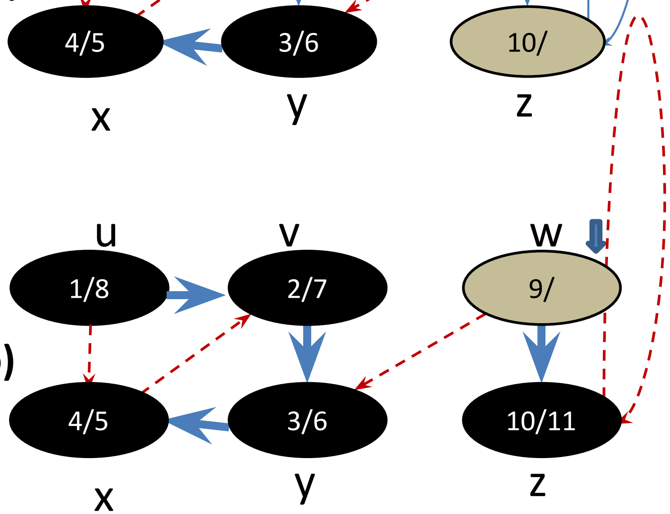


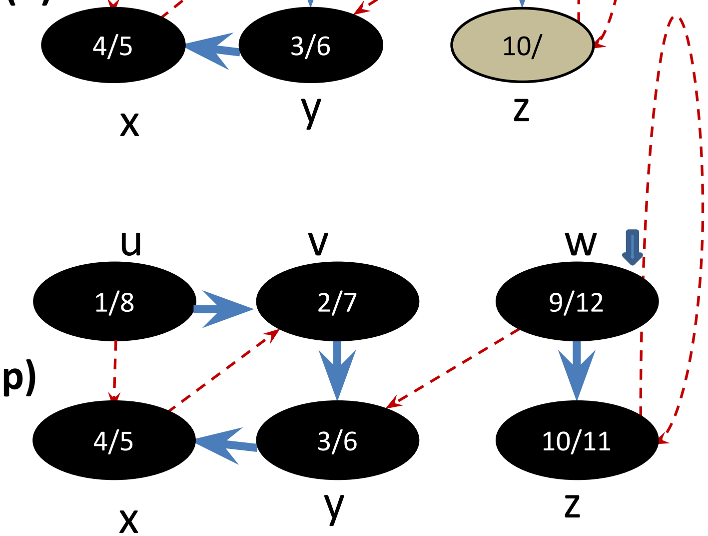
Graph Traversal

- **Depth First Search (DFS)** mechanism

(m)


(n)


(o)


(p)


DFS traversal:

x->y->v->u->z->w

O/P sequence may be different, it depends on the sequence in which I am considering the nodes of a particular depth

Graph Traversal

- **Depth First Search (DFS)** Algorithm

```
DFS(G)
 for each vertex u ϵ G.V
    u.color = WHITE
    u.π = NIL   // predecessor of node u
 time=0
 for each vertex u ϵ 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 v ϵ G.Adj[u]   // explore edge (u,v)
      if v.color == WHITE
         v.π = u
         DFS_Visit(G,v)
   u.color = BLACK
   time=time+1
   u.f=time
```

Graph Traversal

- **Depth First Search (DFS)** Complexity analysis

1st for loop of DFS() : The initialization process requires **Θ(V)** time as all the vertices are initialized each of which takes **Θ(1)** time
2nd for loop of DFS() can execute maximum **V** times.
Hence, DFS() requires Θ(V) +Θ(V) = Θ(V) times

Now, in DFS_Visit(), each WHITE vertex discovers some edges of graph G. DFS_Visit() can be called each of the vertices at most once.
In this process, all edges should be discovered. Hence, it requires Θ(E) time. Since,

Thus, the total time complexity of DFS algorithm = **Θ(V+E)**

Graph Traversal

- **Graph Traversal** Applications

1. Finding out shortest path and minimum spanning tree

   With BFS, we always reach a vertex from given source using the minimum number of edges
2. In Peer to Peer Networks like BitTorrent, BFS is used to find all neighbors
3. In social networks, we can find people within a given distance 'k' from a person using BFS till 'k' levels
4. BFS is used to find all neighboring locations in GPS navigation system
5. In networks, a broadcasted packet follows BFS to reach all nodes
6. Solving puzzle games as it needs to track the previous steps and backtracking, DFS is used
7. DFS is used for solving various other graph based problems like – shortest path, spanning tree, connected component, operating system deadlock, topological sorting etc.

Queries?

Practice Problem

1. Find the DFS traversal for the graph shown below.