

Graphs: Depth First Search

Bijendra Nath Jain
bnjain@iiitd.ac.in

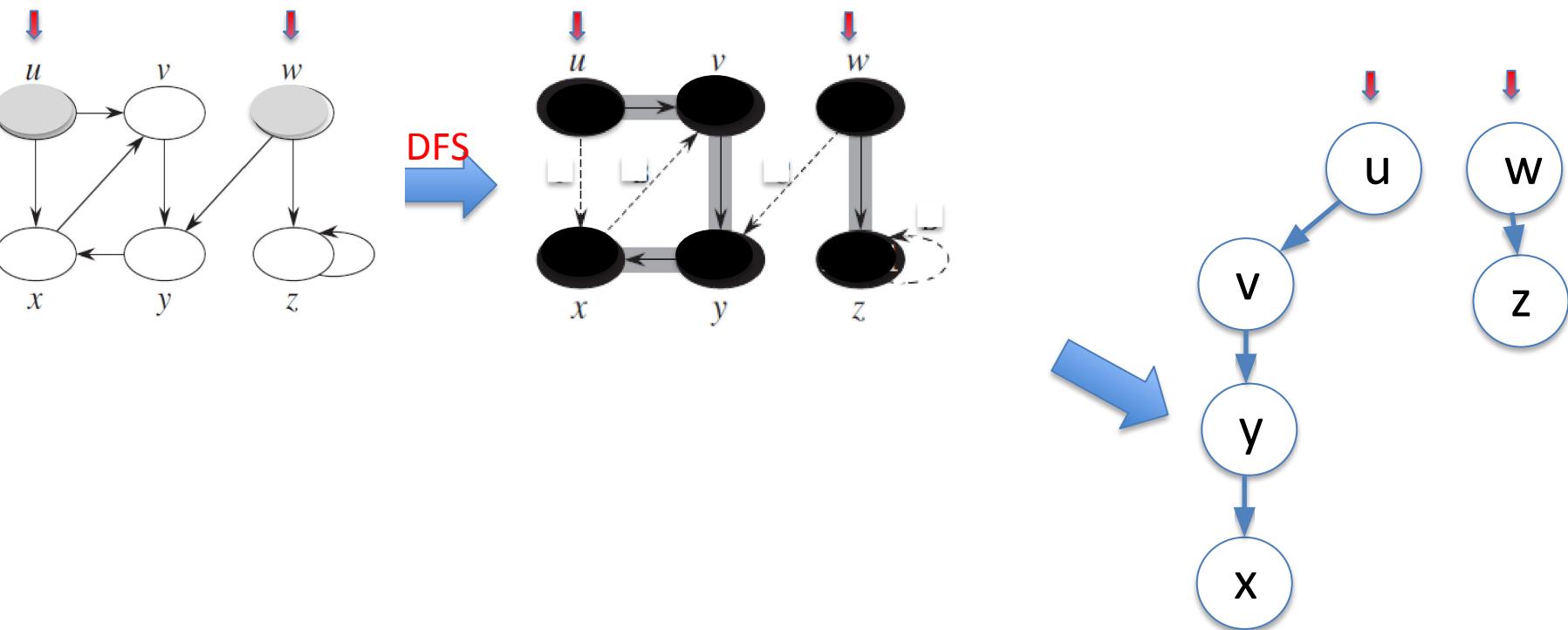
Some of the slides are from <https://courses.cs.washington.edu/courses/cse373/22sp/>

Outline

- Graphs:
 - Undirected graphs
 - Directed graphs
 - (Directed) acyclic graphs (or DAGs)
 - Sparse graphs
 - Weighted graphs
- Graph applications
- Representation of graphs:
 - Adjacency matrix
 - Linked lists
- Algorithms:
 - Traversal algorithms:
 - BFS
 - DFS
 - Topological sort
 - Minimum spanning trees
 - Dijkstra's Shortest path
 - One-to-one
 - One-to-many
 - Many-to-many

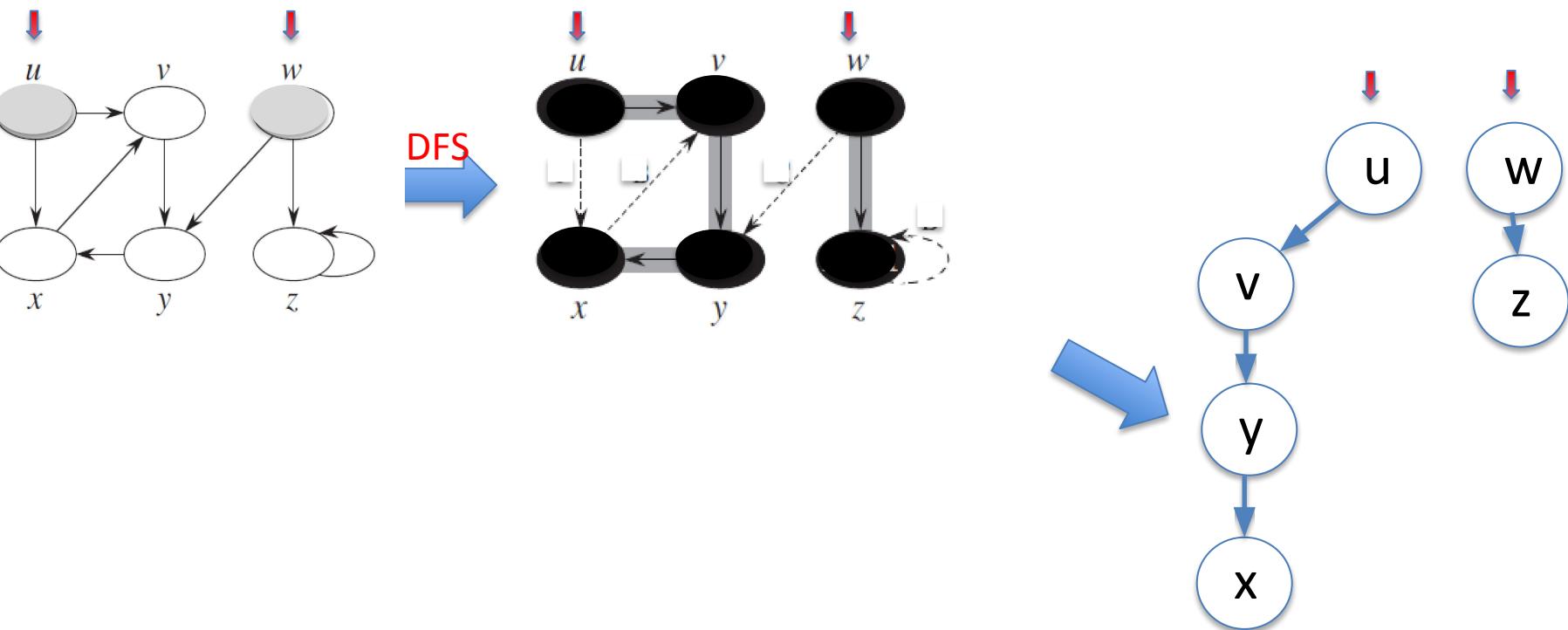
Depth-First Search

- Depth-First Search (BFS)
 - Discovers every vertex by searching down one path as deep as possible
 - When no vertex is discovered, the algorithm simply backtracks
 - When backtracking, it explores vertices that were reachable but not discovered
 - Allows an easy recursive implementation (see next slide)
 - essentially requires a stack (not a queue)
 - Study of DFS helps solve other problems, such as “topological sorting”



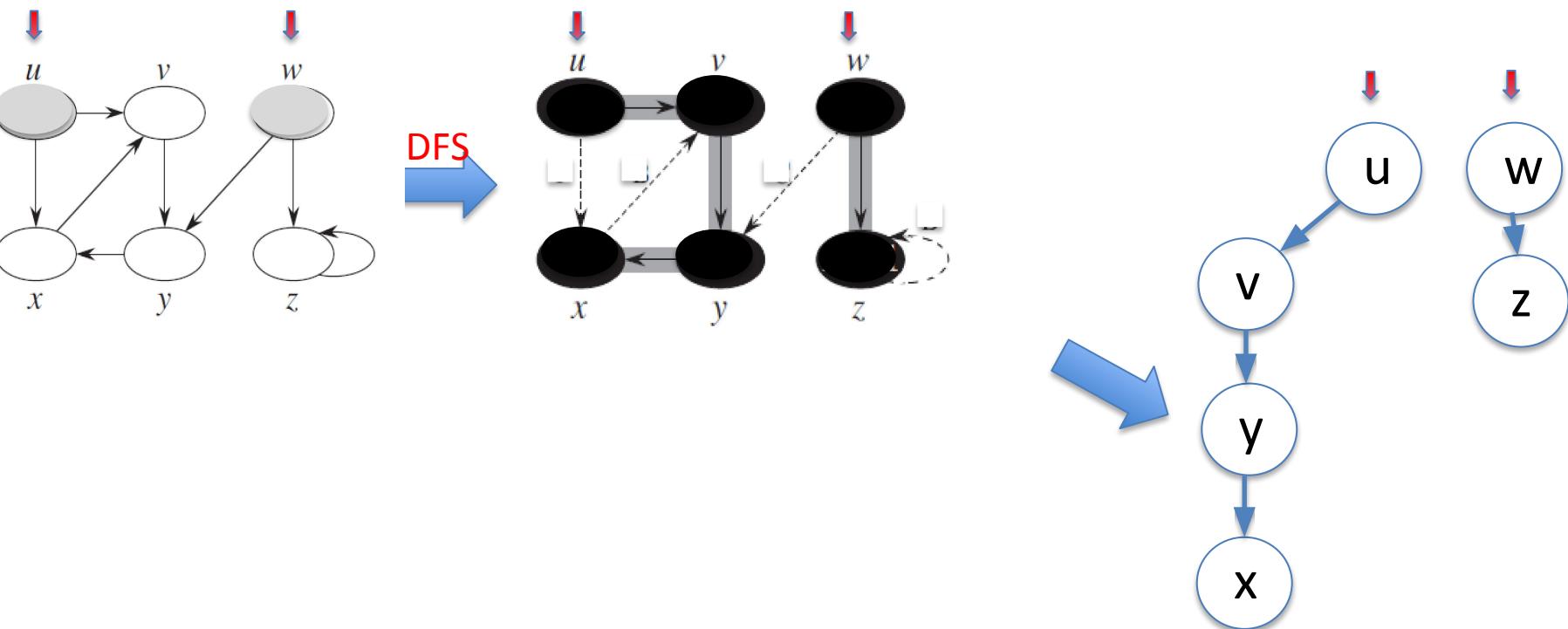
Depth-First Search

- Depth-First Search (BFS)
 - Discovers every vertex by searching down one path as deep as possible
 - When no vertex is discovered, the algorithm simply backtracks
 - When backtracking, it explores vertices that were reachable but not discovered
 - Allows an easy recursive implementation (see next slide)
 - essentially requires a stack (not a queue)
 - Study of DFS helps solve other problems, such as “topological sorting”



Depth-First Search

- Depth-First Search (BFS)
 - Discovers every vertex by searching down one path as deep as possible
 - When no vertex is discovered, the algorithm simply backtracks
 - When backtracking, it explores vertices that were reachable but not discovered
 - Allows an easy recursive implementation (see next slide)
 - essentially requires a stack (not a queue)
 - Study of DFS helps solve other problems, such as “topological sorting”



Depth-First Search

- Depth-First Search (BFS)
 - Algorithm that uses recursion

Initialization step

Recursive algorithm

$\text{DFS}(G)$

```
1 for each vertex  $u \in G.V$ 
2    $u.\text{color} = \text{WHITE}$  ➔
3
4
5 for each vertex  $u \in G.V$ 
6   if  $u.\text{color} == \text{WHITE}$ 
7     DFS-VISIT( $G, u$ )
```

$\text{DFS-VISIT}(G, u)$

```
1
2
3  $u.\text{color} = \text{GRAY}$  // white vertex  $u$  has just
4 for each  $v \in G.\text{Adj}[u]$ 
5   if  $v.\text{color} == \text{WHITE}$ 
6
7     DFS-VISIT( $G, v$ ) // explore edge
8    $u.\text{color} = \text{BLACK}$  // blacken  $u$ ; it is finished
9
10
```

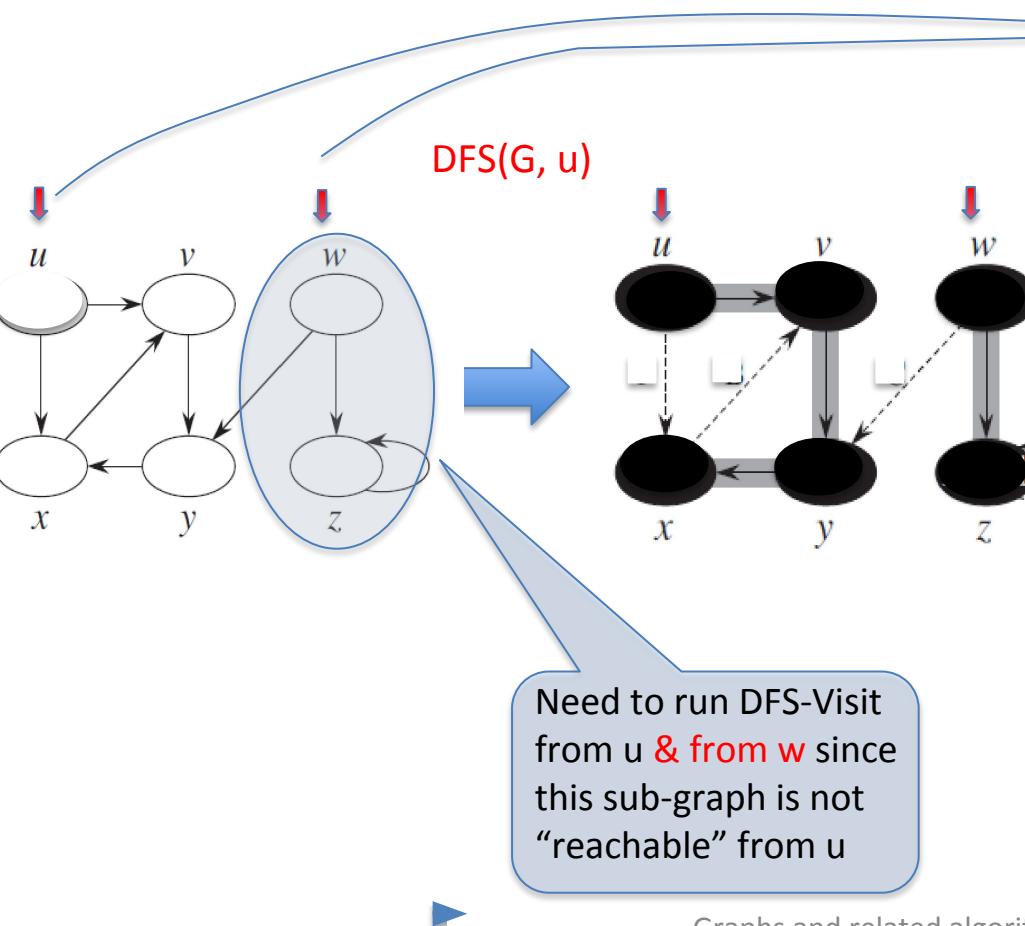
Colour = status
of traversing
vertex u

Colour = Grey
☒ u is discovered, now
DFS will start from u

Colour = Black
☒ DFS from u is
complete

Depth-First Search

- Depth-First Search (BFS)
 - Algorithm that uses recursion



DFS(G)

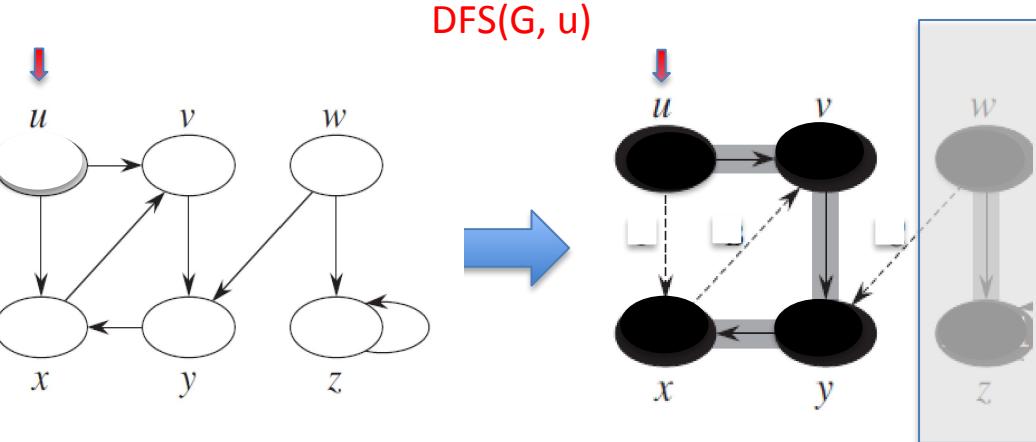
```
1 for each vertex  $u \in G.V$ 
2    $u.color = \text{WHITE}$ 
3
4
5 for each vertex  $u \in G.V$ 
6   if  $u.color == \text{WHITE}$ 
7     DFS-VISIT( $G, u$ )
```

DFS-VISIT(G, u)

```
1
2
3    $u.color = \text{GRAY} //$  white vertex  $u$  has just
4   for each  $v \in G.Adj[u]$ 
5     if  $v.color == \text{WHITE}$ 
6       DFS-VISIT( $G, v$ ) // explore edge
7
8    $u.color = \text{BLACK} //$  blacken  $u$ ; it is finished
9
10
```

Depth-First Search

- Depth-First Search (BFS)
 - Algorithm that uses recursion



DFS(G)

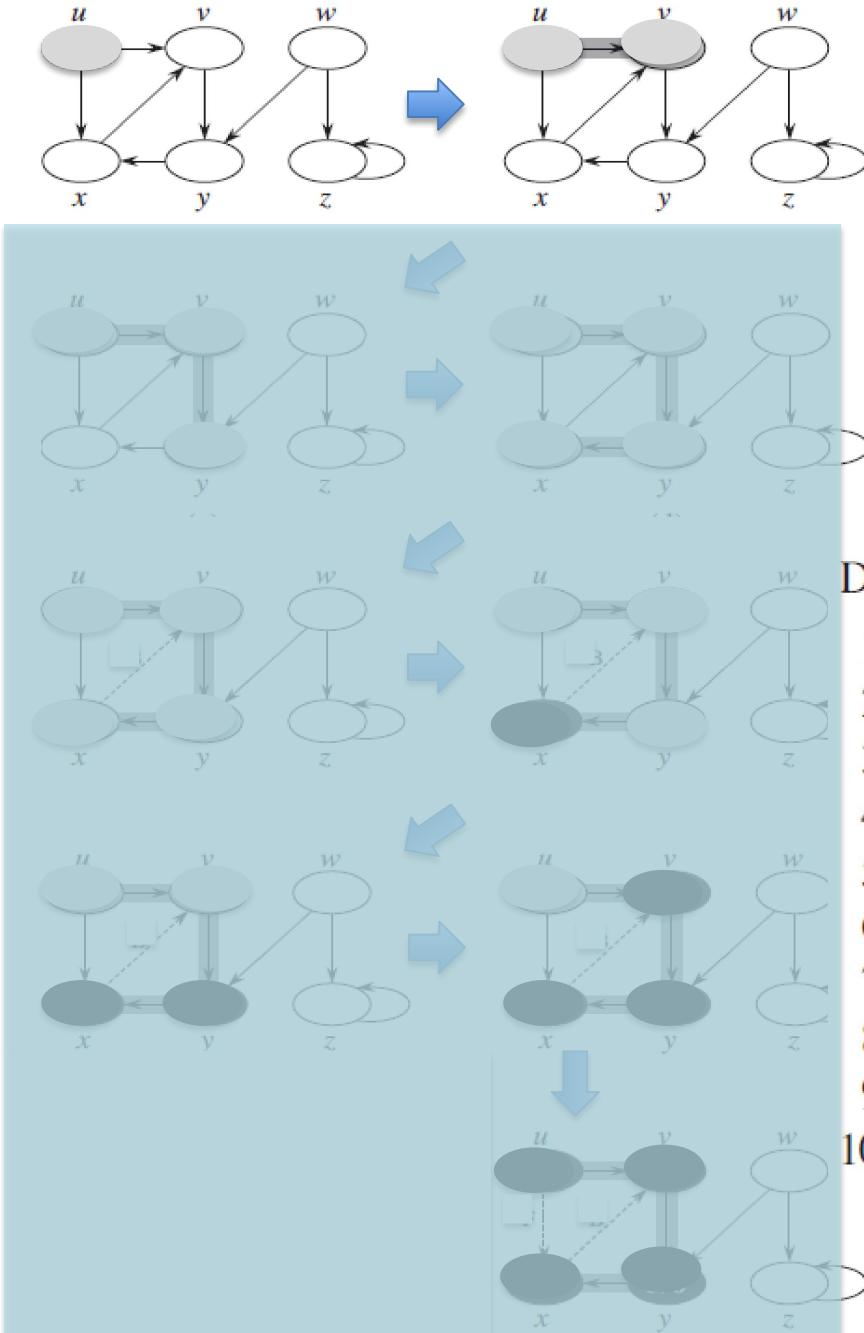
```
1 for each vertex  $u \in G.V$ 
2    $u.color = \text{WHITE}$ 
3
4
5 for each vertex  $u \in G.V$ 
6   if  $u.color == \text{WHITE}$ 
7     DFS-VISIT( $G, u$ )
```

Presently, we discuss
DFS-Visit from vertex u

DFS-VISIT(G, u)

```
1
2
3    $u.color = \text{GRAY} //$  white vertex  $u$  has just
4   for each  $v \in G.Adj[u]$ 
5     if  $v.color == \text{WHITE}$ 
6       DFS-VISIT( $G, v$ ) // explore edge
7
8    $u.color = \text{BLACK} //$  blacken  $u$ ; it is finished
9
10
```

Depth-First Search



DFS-VISIT(G, u)

```

1
2
3  $u.\text{color} = \text{GRAY}$  // white vertex  $u$  has just been discovered
4 for each  $v \in G.\text{Adj}[u]$ 
5   if  $v.\text{color} == \text{WHITE}$ 
6     DFS-VISIT( $G, v$ ) // explore edge  $(u, v)$ 
7
8  $u.\text{color} = \text{BLACK}$  // blacken  $u$ ; it is finished
9
10

```

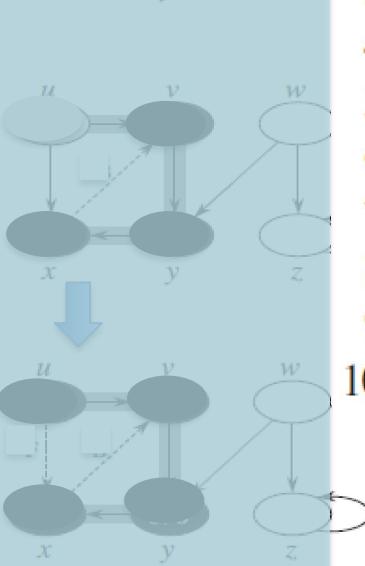
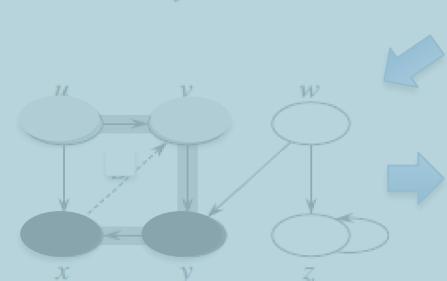
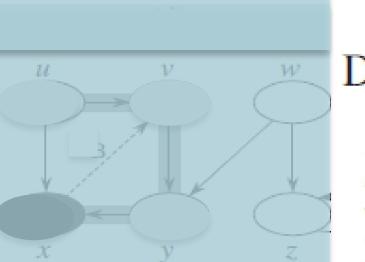
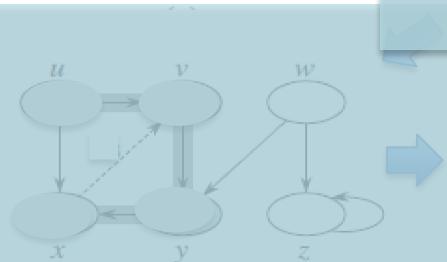
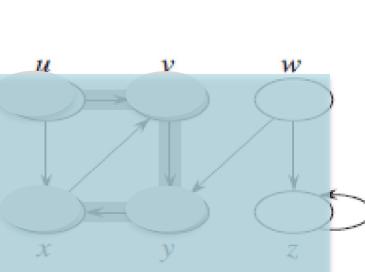
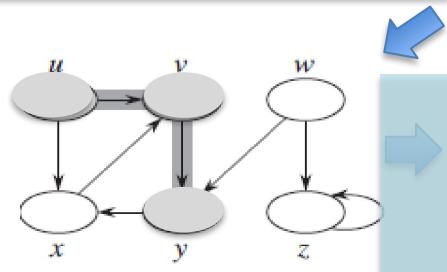
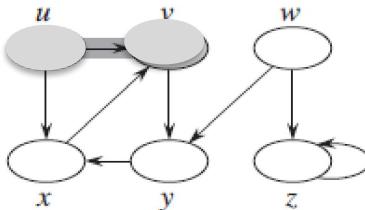
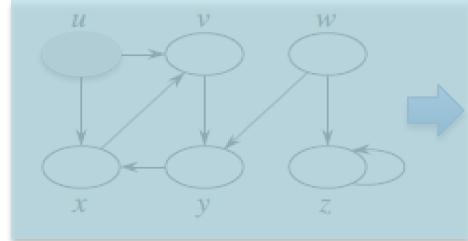
Colour = Grey

❑ u is discovered, now DFS will start from u

Colour = Black

❑ DFS from u is complete

Depth-First Search



DFS-VISIT(G, u)

```

1
2
3  $u.\text{color} = \text{GRAY}$  // white vertex  $u$  has just been discovered
4 for each  $v \in G.\text{Adj}[u]$ 
5   if  $v.\text{color} == \text{WHITE}$ 
6     DFS-VISIT( $G, v$ ) // explore edge  $(u, v)$ 
7
8    $u.\text{color} = \text{BLACK}$  // blacken  $u$ ; it is finished
9
10

```

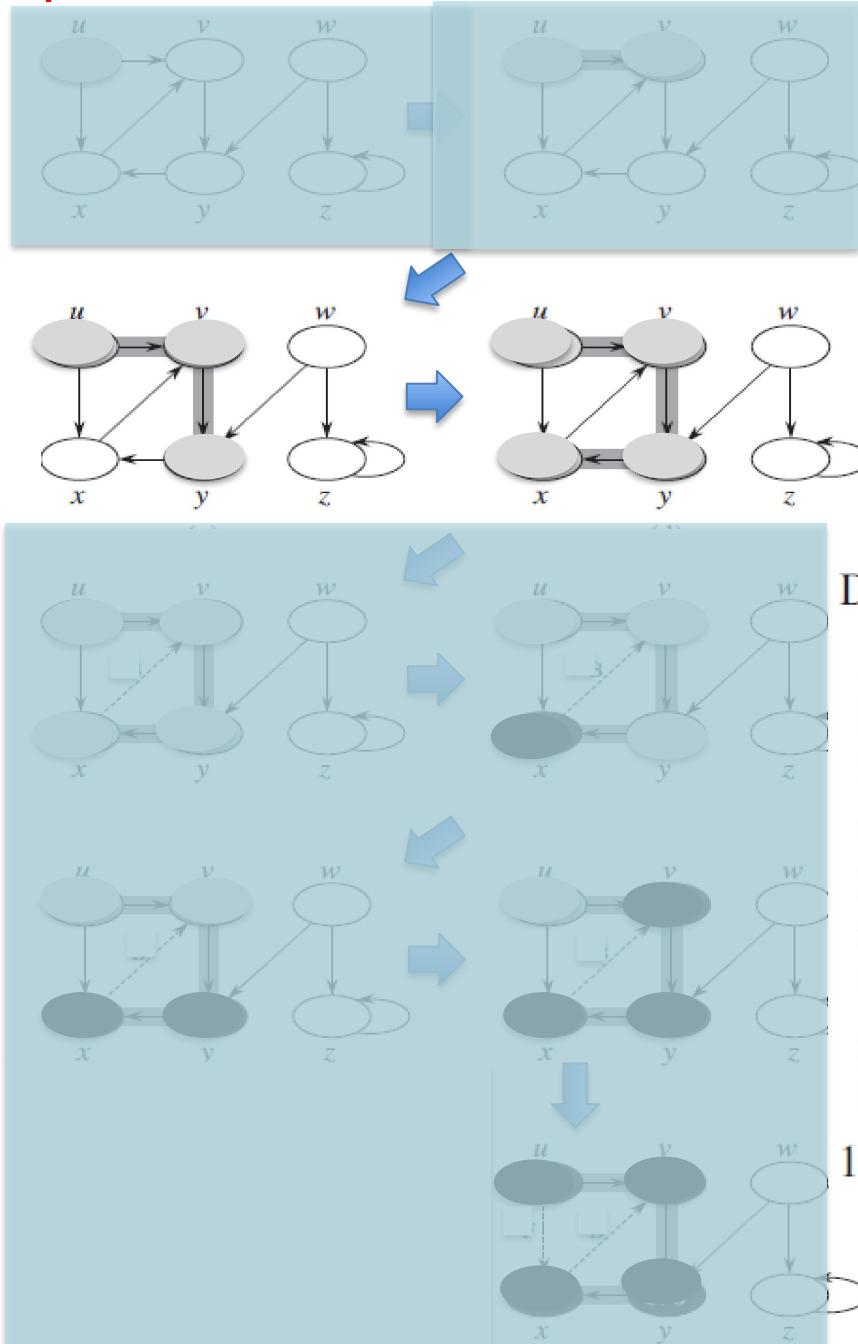
Colour = Grey

❑ u is discovered, now DFS will start from u

Colour = Black

❑ DFS from u is complete

Depth-First Search



DFS-VISIT(G, u)

```
1  _____  
2  _____  
3   $u.color = \text{GRAY}$  // white vertex  $u$  has just been discovered  
4  for each  $v \in G.Adj[u]$   
5    if  $v.color == \text{WHITE}$   
6      _____  
7      DFS-VISIT( $G, v$ ) // explore edge  $(u, v)$   
8     $u.color = \text{BLACK}$  // blacken  $u$ ; it is finished  
9  _____  
10 _____
```

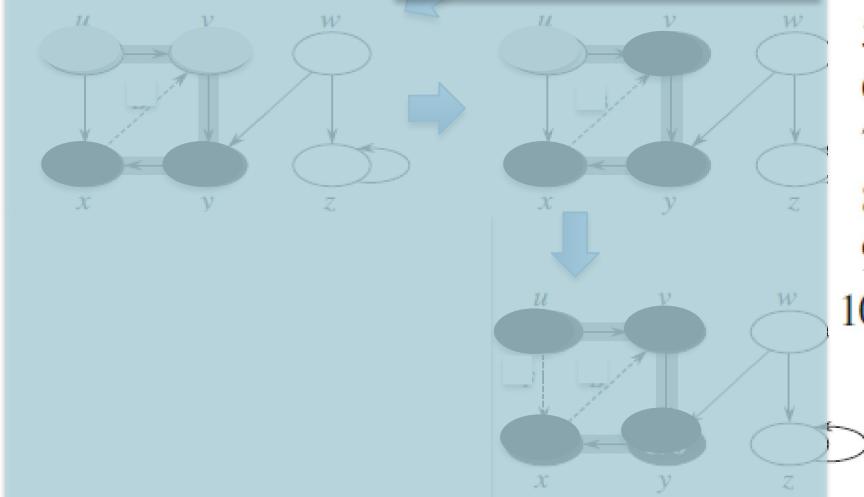
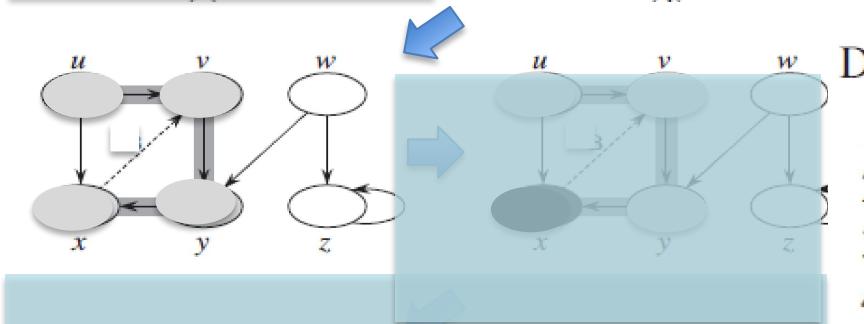
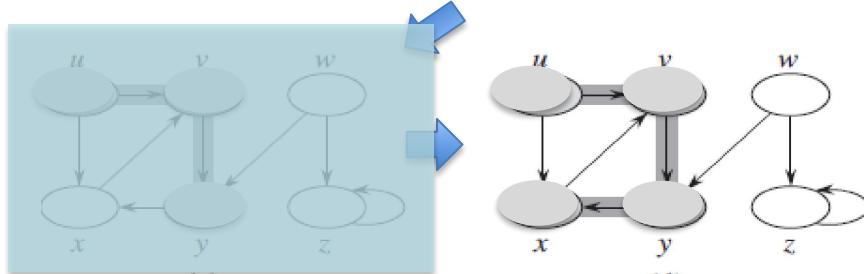
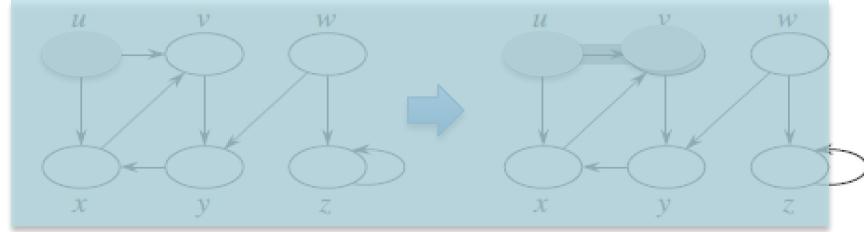
Colour = Grey

❑ u is discovered, now DFS will start from u

Colour = Black

❑ DFS from u is complete

Depth-First Search



DFS-VISIT(G, u)

```

1
2
3  $u.\text{color} = \text{GRAY}$  // white vertex  $u$  has just been discovered
4 for each  $v \in G.\text{Adj}[u]$ 
5   if  $v.\text{color} == \text{WHITE}$ 
6     DFS-VISIT( $G, v$ ) // explore edge  $(u, v)$ 
7
8    $u.\text{color} = \text{BLACK}$  // blacken  $u$ ; it is finished
9
10

```

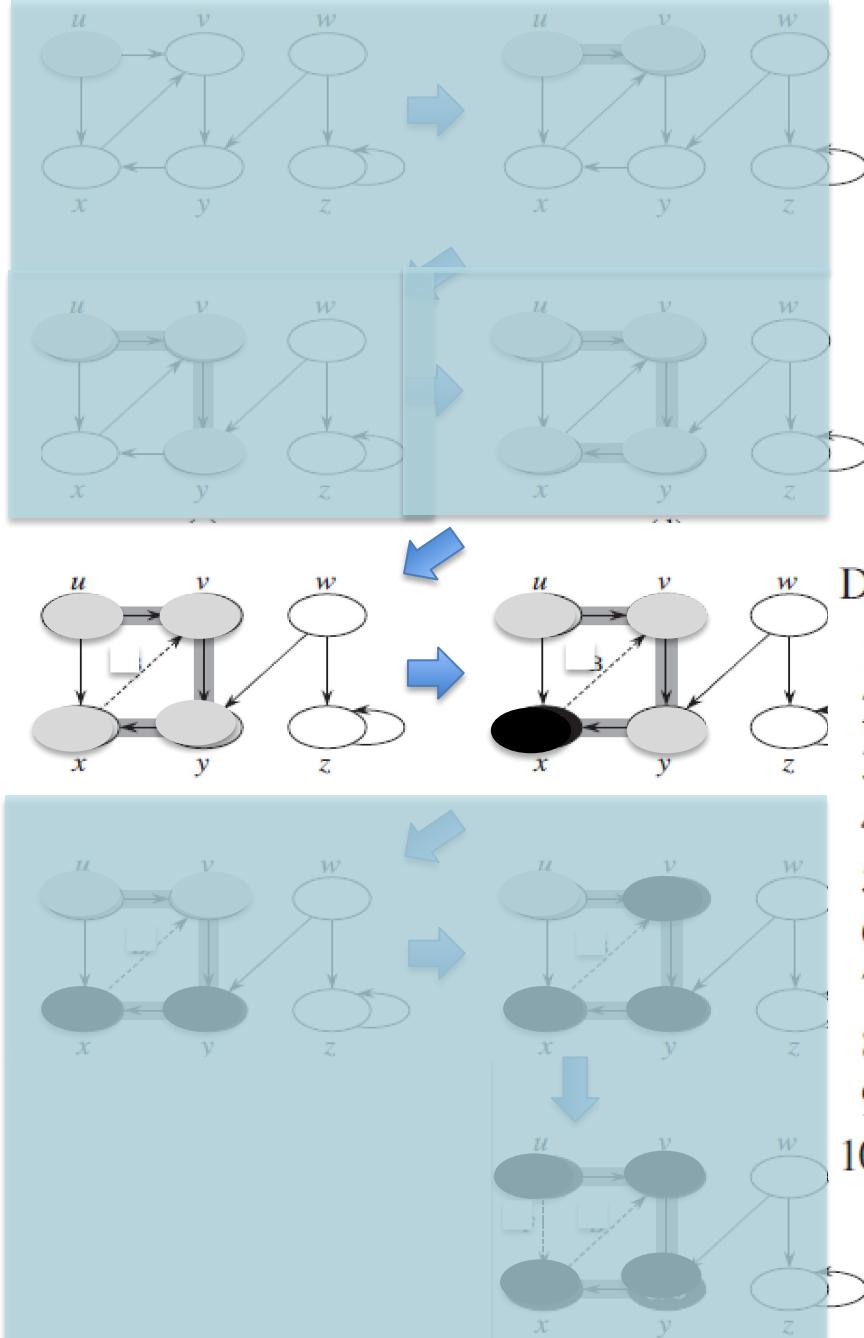
Colour = Grey

❑ u is discovered, now DFS will start from u

Colour = Black

❑ DFS from u is complete

Depth-First Search



DFS-VISIT(G, u)

```

1
2
3  $u.\text{color} = \text{GRAY}$  // white vertex  $u$  has just been discovered
4 for each  $v \in G.\text{Adj}[u]$ 
5   if  $v.\text{color} == \text{WHITE}$ 
6     DFS-VISIT( $G, v$ ) // explore edge  $(u, v)$ 
7
8    $u.\text{color} = \text{BLACK}$  // blacken  $u$ ; it is finished
9
10

```

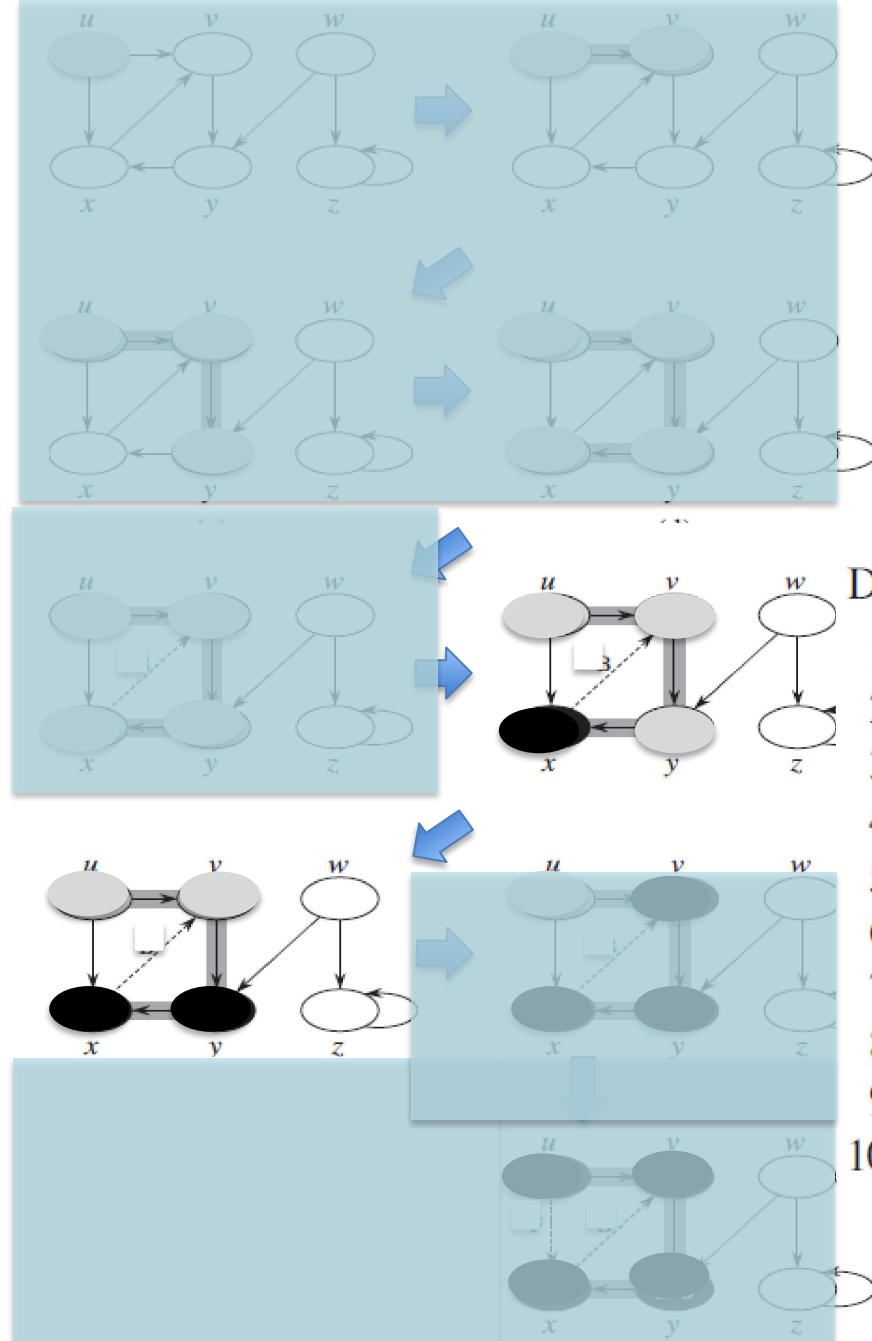
Colour = Grey

❑ u is discovered, now DFS will start from u

Colour = Black

❑ DFS from u is complete

Depth-First Search



DFS-VISIT(G, u)

```

1
2
3  $u.\text{color} = \text{GRAY}$  // white vertex  $u$  has just been discovered
4 for each  $v \in G.\text{Adj}[u]$ 
5   if  $v.\text{color} == \text{WHITE}$ 
6     DFS-VISIT( $G, v$ ) // explore edge  $(u, v)$ 
7
8    $u.\text{color} = \text{BLACK}$  // blacken  $u$ ; it is finished
9
10

```

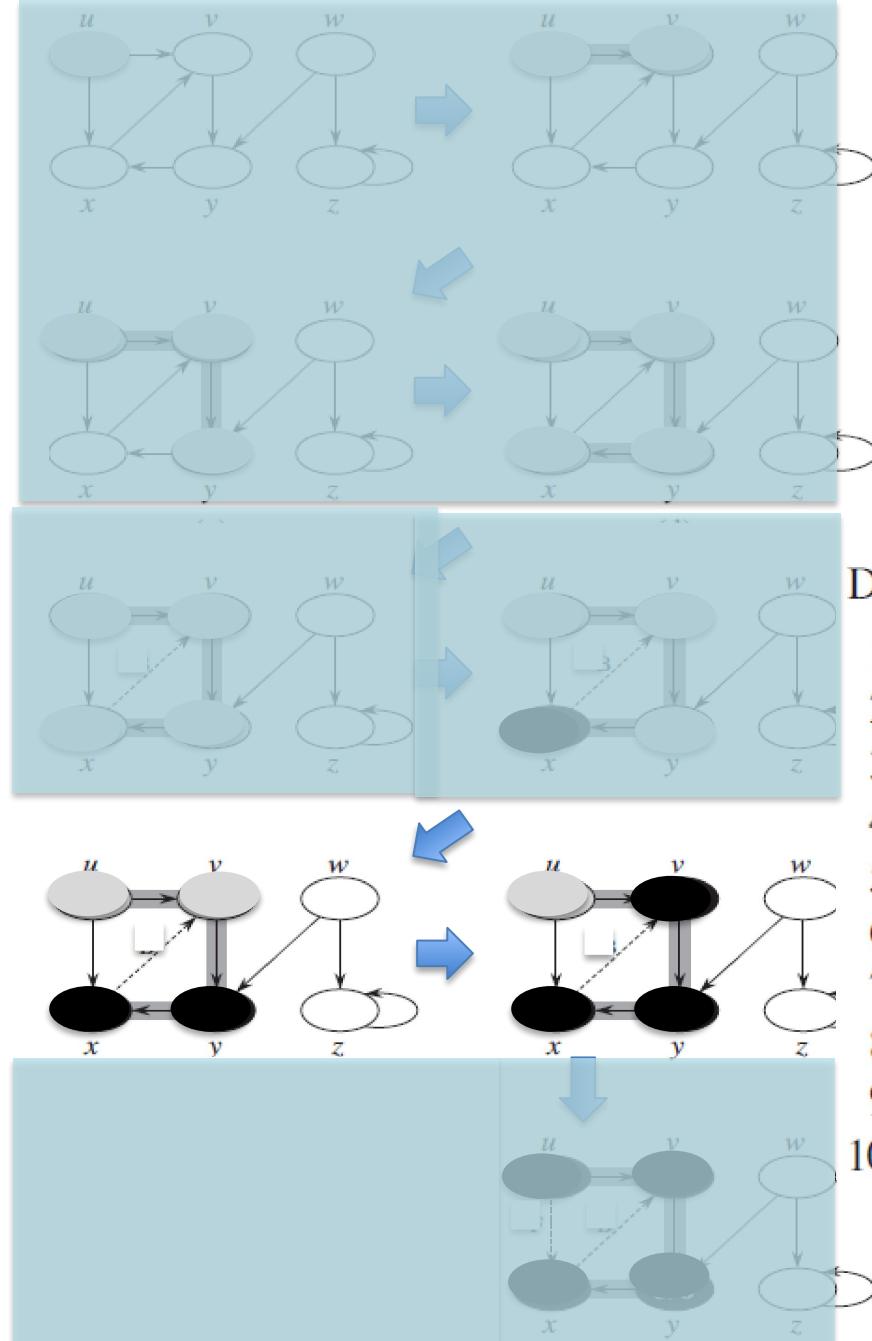
Colour = Grey

❑ u is discovered, now DFS will start from u

Colour = Black

❑ DFS from u is complete

Depth-First Search



DFS-VISIT(G, u)

```

1
2
3  $u.\text{color} = \text{GRAY}$  // white vertex  $u$  has just been discovered
4 for each  $v \in G.\text{Adj}[u]$ 
5   if  $v.\text{color} == \text{WHITE}$ 
6     DFS-VISIT( $G, v$ ) // explore edge  $(u, v)$ 
7
8    $u.\text{color} = \text{BLACK}$  // blacken  $u$ ; it is finished
9
10

```

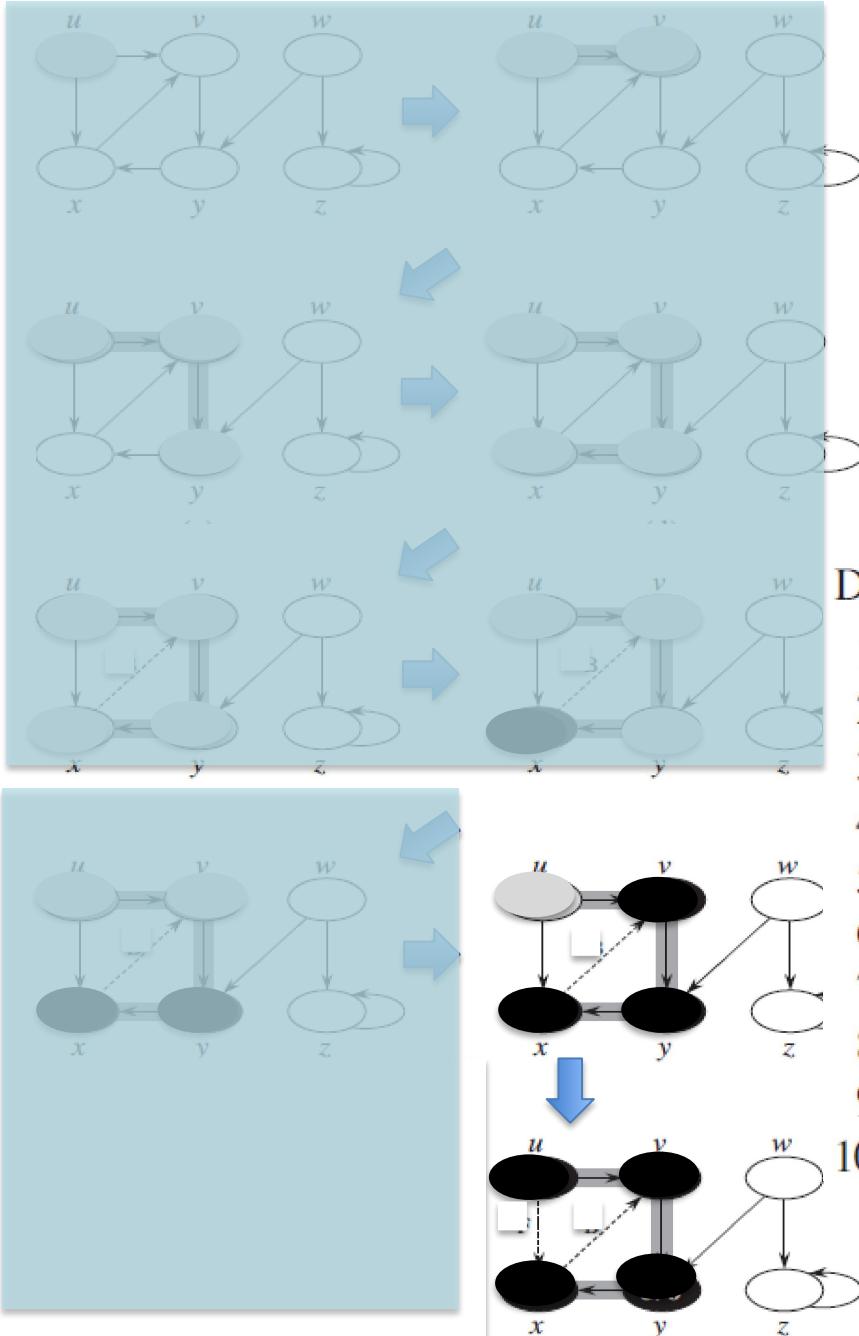
Colour = Grey

❑ u is discovered, now DFS will start from u

Colour = Black

❑ DFS from u is complete

Depth-First Search



DFS-VISIT(G, u)

```

1
2
3  $u.\text{color} = \text{GRAY}$  // white vertex  $u$  has just been discovered
4 for each  $v \in G.\text{Adj}[u]$ 
5   if  $v.\text{color} == \text{WHITE}$ 
6     DFS-VISIT( $G, v$ ) // explore edge  $(u, v)$ 
7
8  $u.\text{color} = \text{BLACK}$  // blacken  $u$ ; it is finished
9
10

```

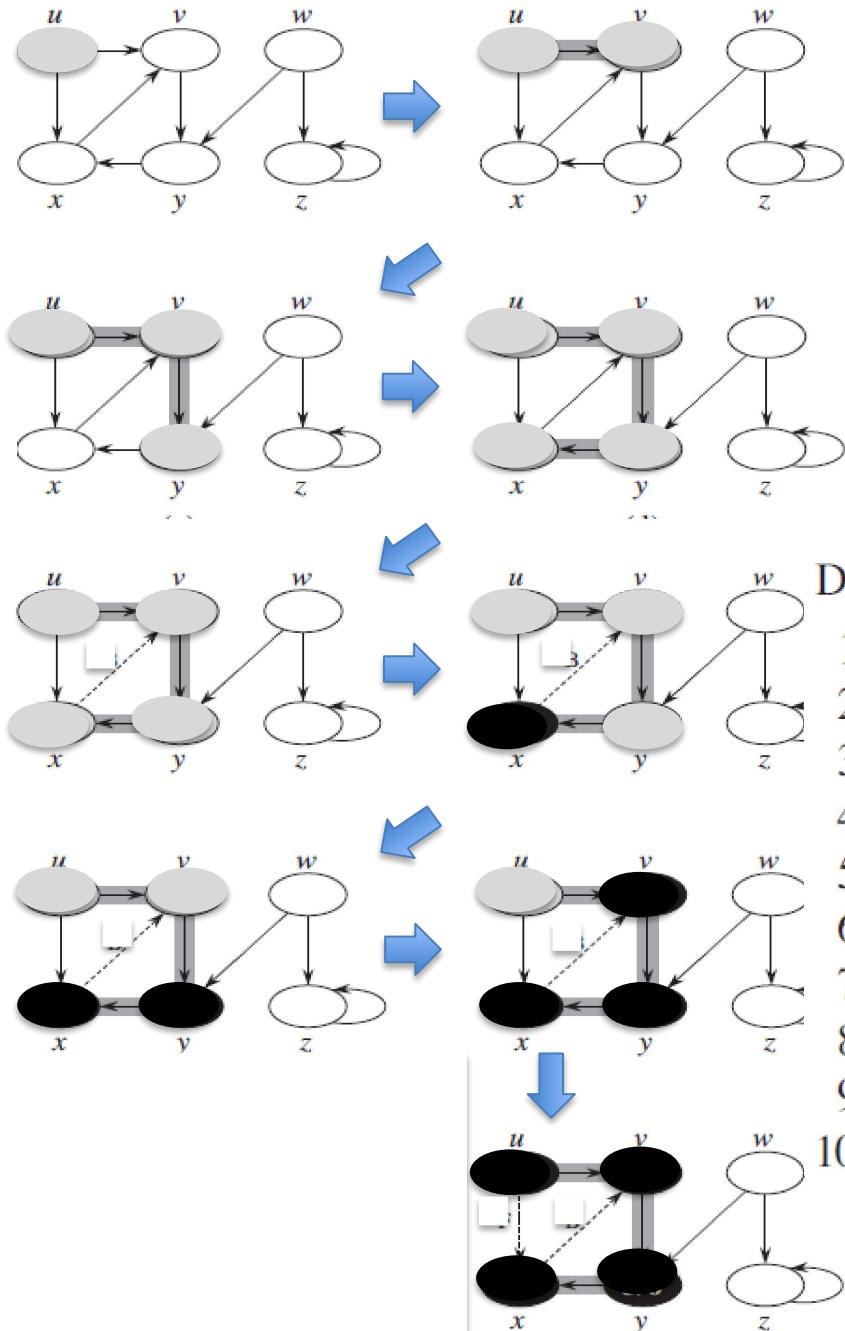
Colour = Grey

❑ u is discovered, now DFS will start from u

Colour = Black

❑ DFS from u is complete

Depth-First Search



DFS-VISIT(G, u)

1 []

2 []

3 $u.\text{color} = \text{GRAY}$ // white vertex u has just been discovered

4 for each $v \in G.\text{Adj}[u]$

5 if $v.\text{color} == \text{WHITE}$

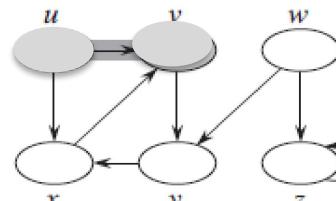
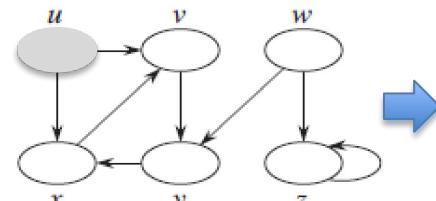
6 $v.\pi = u$

7 DFS-VISIT(G, v) // explore edge (u, v)

8 $u.\text{color} = \text{BLACK}$ // blacken u ; it is finished

$v.\pi = \text{parent of } v$

Depth-First Search

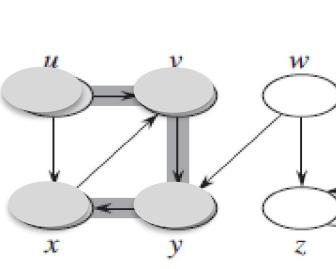
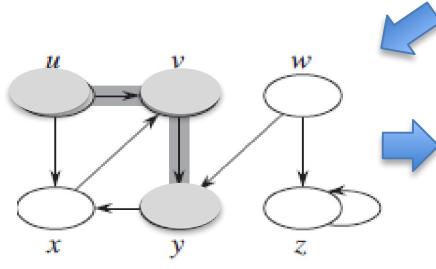


$\text{DFS}(G)$

```

1 for each vertex  $u \in G.V$ 
2    $u.\text{color} = \text{WHITE}$ 
3    $u.\pi = \text{NIL}$ 

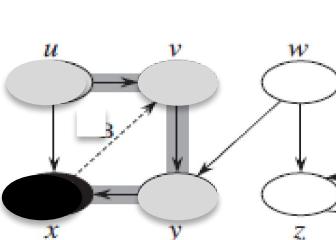
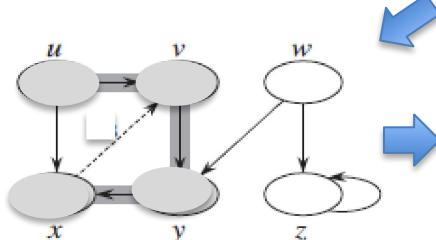
```



```

5 for each vertex  $u \in G.V$ 
6   if  $u.\text{color} == \text{WHITE}$ 
7      $\text{DFS-VISIT}(G, u)$ 

```



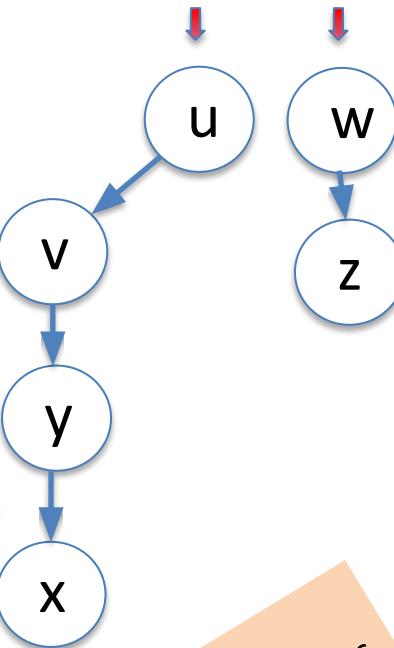
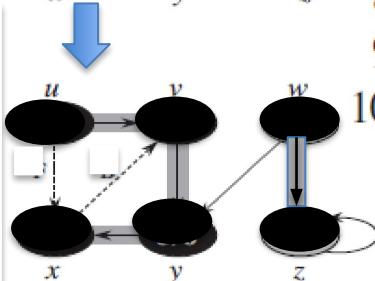
$\text{DFS-VISIT}(G, u)$

```

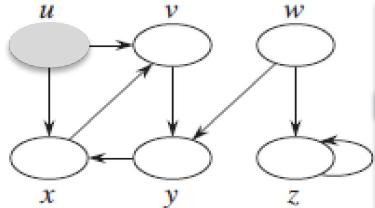
1
2
3  $u.\text{color} = \text{GRAY} // \text{white vertex } u$ 
4 for each  $v \in G.\text{Adj}[u]$ 
5   if  $v.\text{color} == \text{WHITE}$ 
6      $v.\pi = u$ 
7      $\text{DFS-VISIT}(G, v) // \text{explore edge } (u, v)$ 
8
9
10  $u.\text{color} = \text{BLACK} // \text{blacken } u; \text{ it is finished}$ 

```

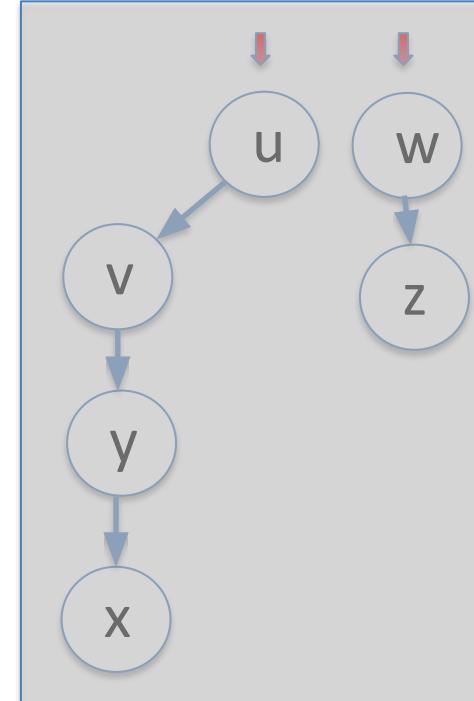
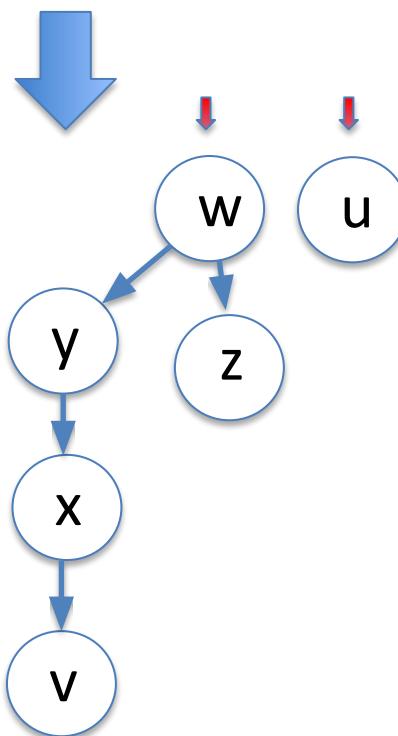
Once $\text{DFS-VISIT}(G, u)$ is done, $\text{DFS-VISIT}(G, w)$ starts to yield the rest of search



Depth-First Search



Resulting DFS Tree, if the search had instead first started from w, and then from u

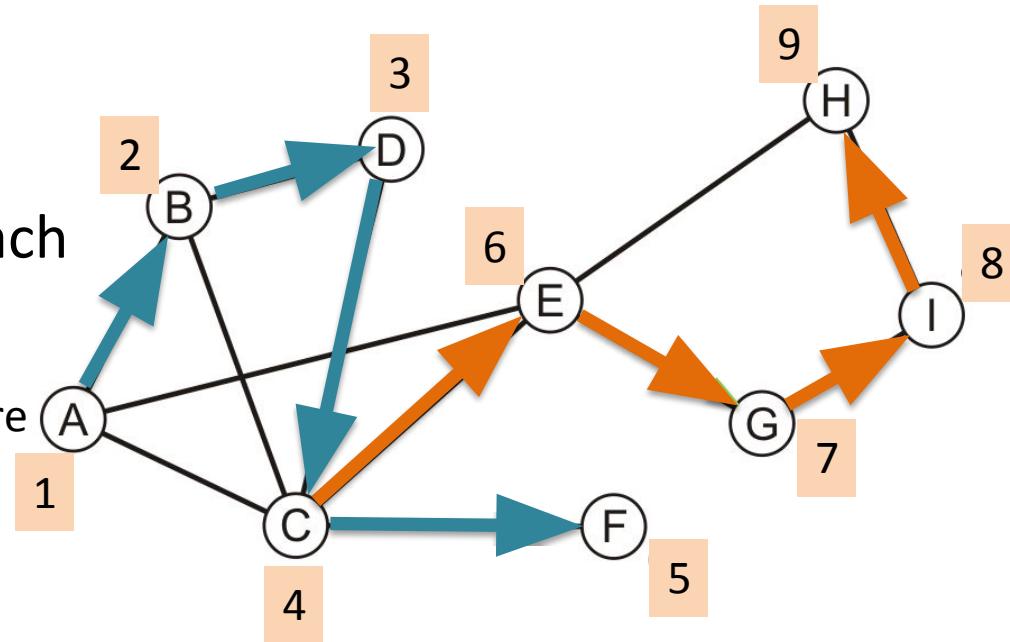


DFS: another example

- What is the resulting DFS tree if search was to start at vertex A

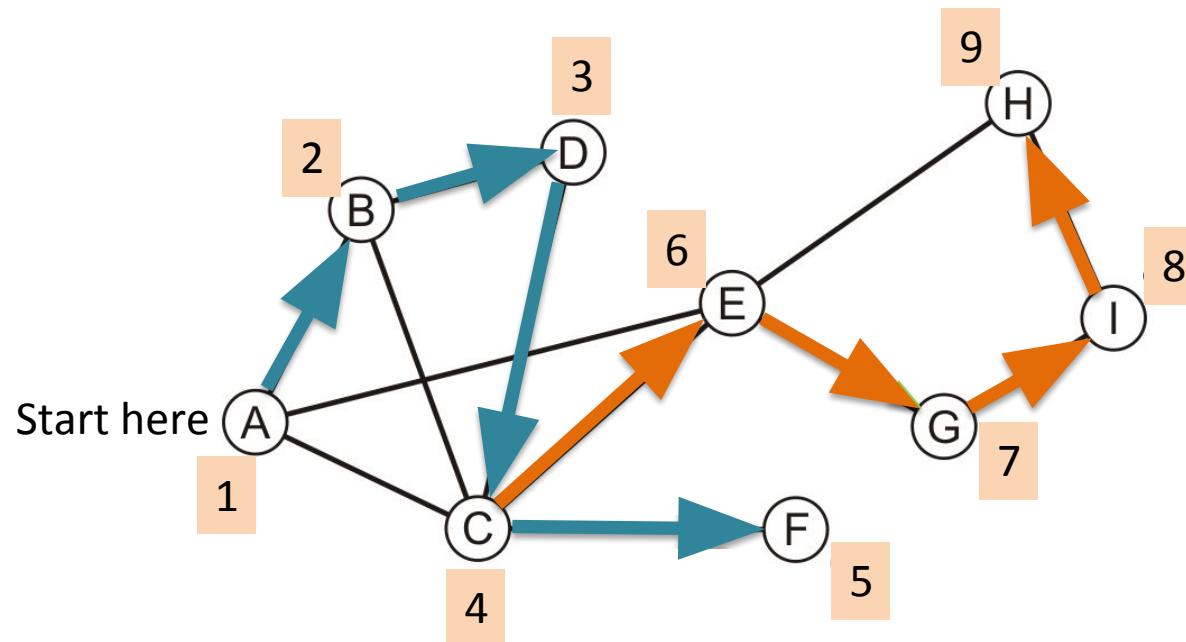
DFS search

Start here

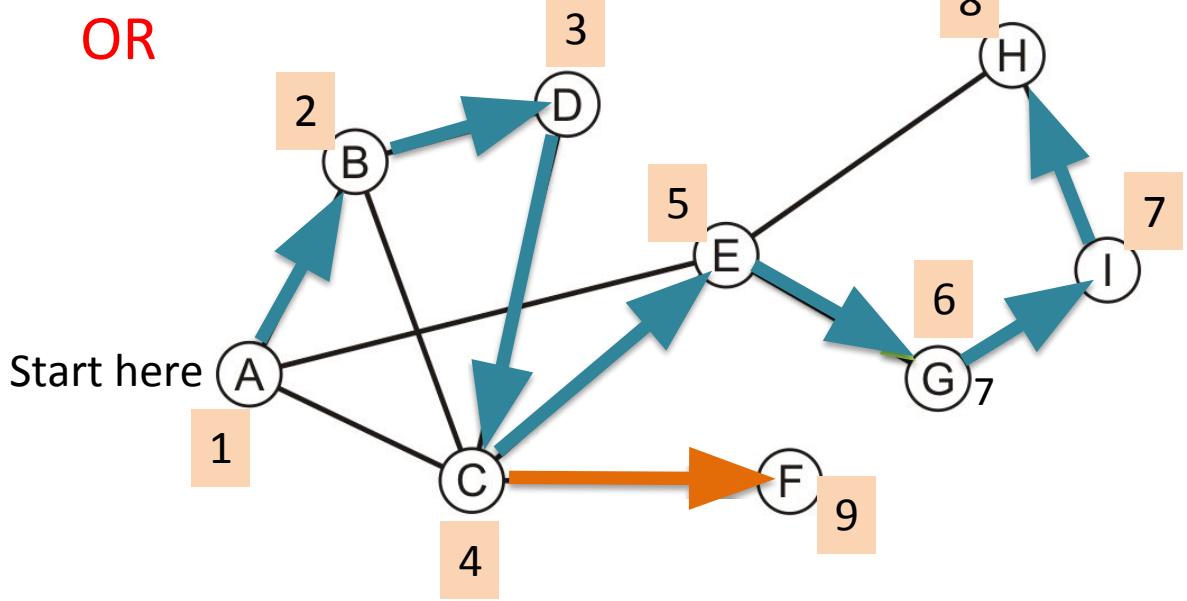


DFS: another example

- What is the resulting DFS tree if search was to start at vertex A



Under what conditions on the adjacency lists would the resulting DFS search (or tree) be as indicated in the two outcomes

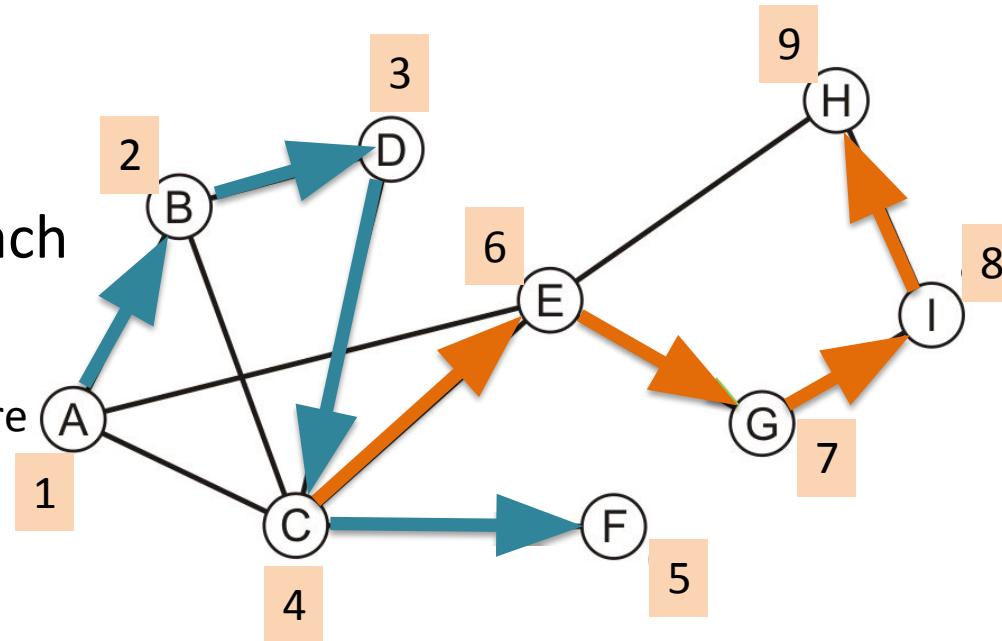


DFS vs. BFS

- What is the resulting DFS tree or BFS tree if search was to start at vertex A

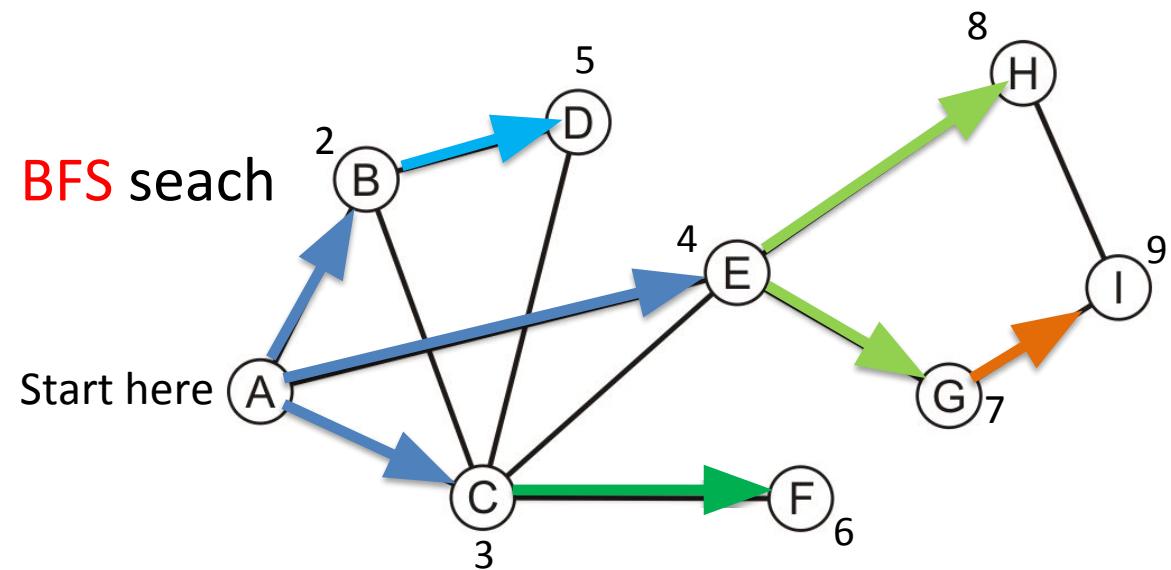
DFS search

Start here



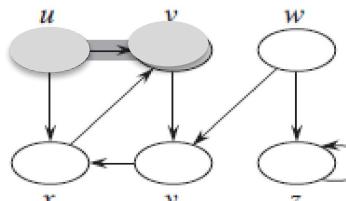
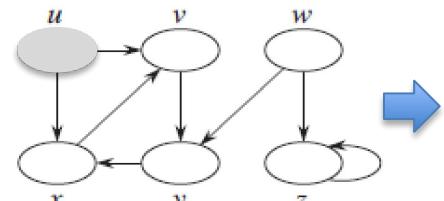
BFS search

Start here



Back to Depth-First Search

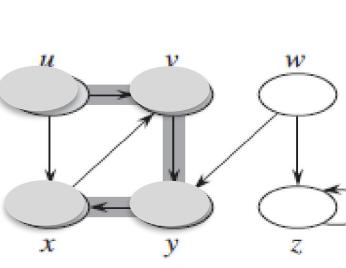
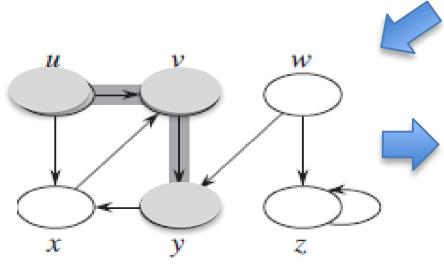
Depth-First Search



$\text{DFS}(G)$

```

1 for each vertex  $u \in G.V$ 
2    $u.\text{color} = \text{WHITE}$ 
3    $u.\pi = \text{NIL}$ 
4    $time = 0$ 
5 for each vertex  $u \in G.V$ 
6   if  $u.\text{color} == \text{WHITE}$ 
7     DFS-VISIT( $G, u$ )
  
```

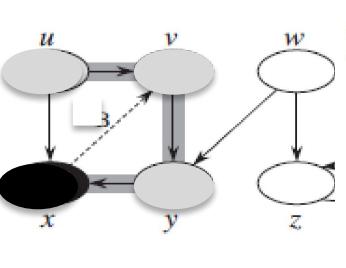
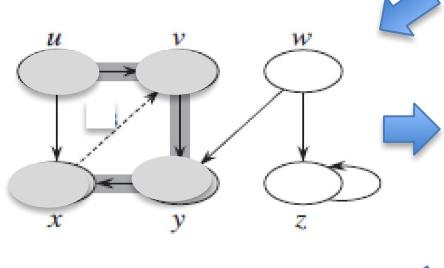


$\text{DFS-VISIT}(G, u)$

```

1  $time = time + 1$ 
2  $u.d = time$ 
3  $u.\text{color} = \text{GRAY} //$  white vertex  $u$  has just been discovered
4 for each  $v \in G.\text{Adj}[u]$ 
5   if  $v.\text{color} == \text{WHITE}$ 
6      $v.\pi = u$ 
7     DFS-VISIT( $G, v$ ) // explore edge  $(u, v)$ 
8    $u.\text{color} = \text{BLACK} //$  blacken  $u$ ; it is finished
9    $time = time + 1$ 
10   $u.f = time$ 
  
```

10

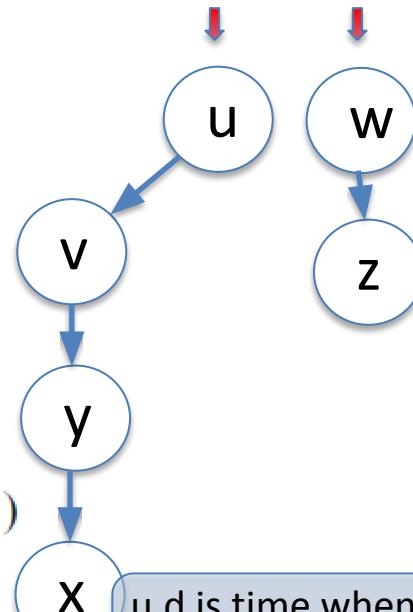


7

8

9

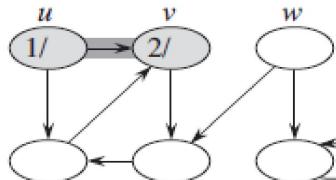
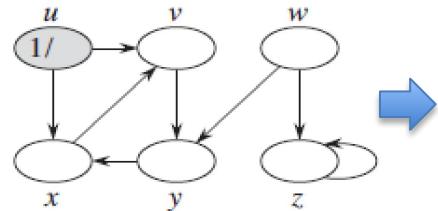
10



$u.d$ is time when u is discovered

$u.f$ is time when DFS from u is finished

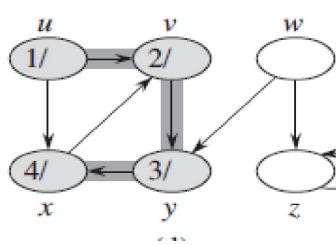
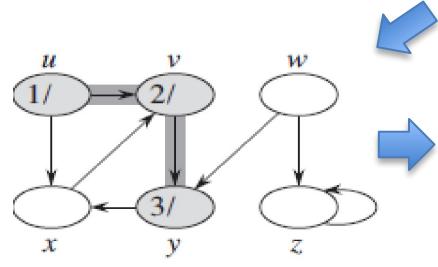
Depth-First Search



$\text{DFS}(G)$

```

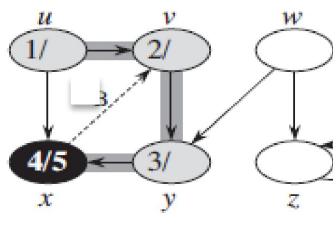
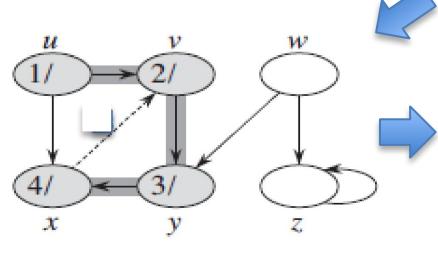
1 for each vertex  $u \in G.V$ 
2    $u.\text{color} = \text{WHITE}$ 
3    $u.\pi = \text{NIL}$ 
4    $time = 0$ 
5 for each vertex  $u \in G.V$ 
6   if  $u.\text{color} == \text{WHITE}$ 
7      $\text{DFS-VISIT}(G, u)$ 
  
```



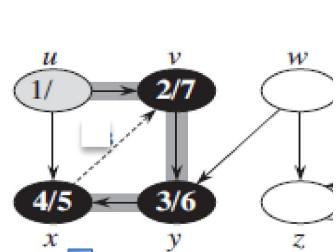
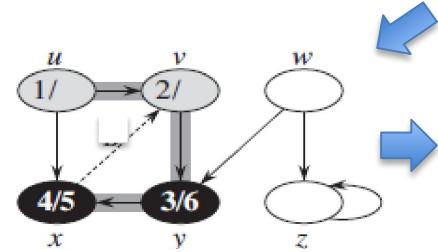
$\text{DFS-VISIT}(G, u)$

```

1    $time = time + 1$ 
2    $u.d = time$ 
3    $u.\text{color} = \text{GRAY} //$  white vertex  $u$  has just been discovered
4   for each  $v \in G.\text{Adj}[u]$ 
5     if  $v.\text{color} == \text{WHITE}$ 
6        $v.\pi = u$ 
7        $\text{DFS-VISIT}(G, v) //$  explore edge  $(u, v)$ 
8      $u.\text{color} = \text{BLACK} //$  blacken  $u$ ; it is finished
9    $time = time + 1$ 
10   $u.f = time$ 
  
```

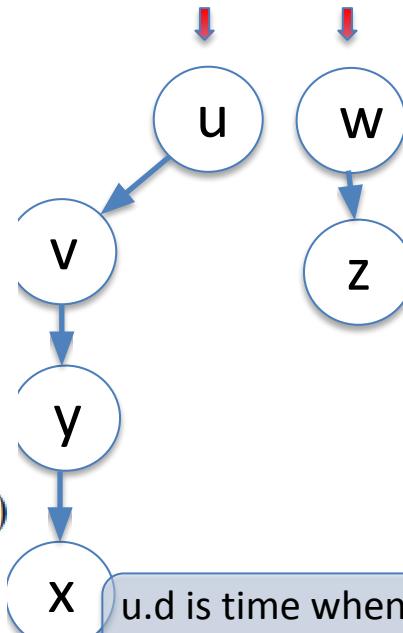


$\text{DFS-VISIT}(G, u)$

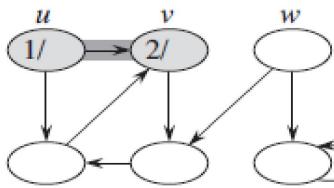
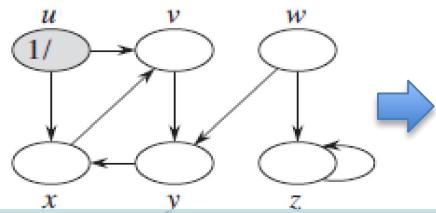


10

$u.f$ is time when DFS from u is finished



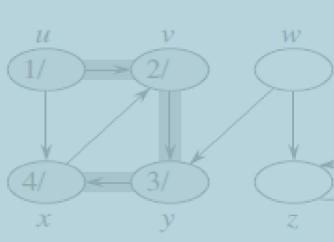
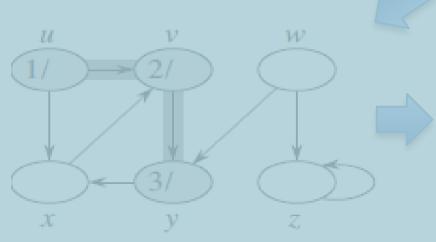
Depth-First Search



$\text{DFS}(G)$

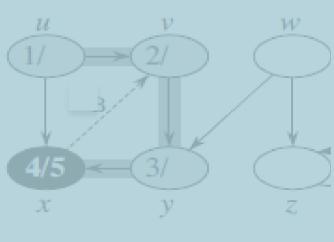
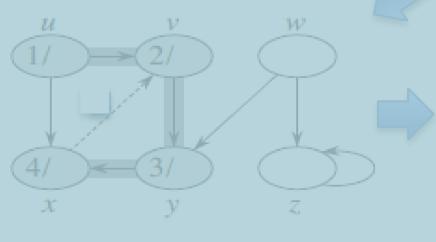
```

1 for each vertex  $u \in G.V$ 
2    $u.\text{color} = \text{WHITE}$ 
3    $u.\pi = \text{NIL}$ 
4    $time = 0$ 
5 for each vertex  $u \in G.V$ 
6   if  $u.\text{color} == \text{WHITE}$ 
7     DFS-VISIT( $G, u$ )
  
```



```

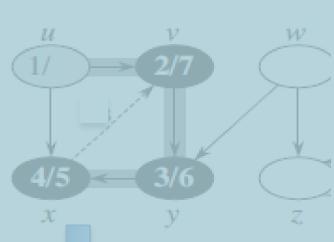
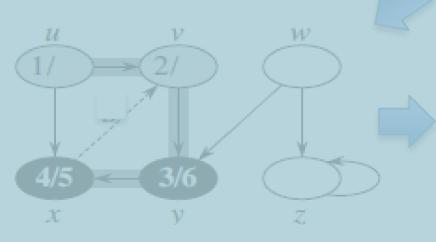
5 for each vertex  $u \in G.V$ 
6   if  $u.\text{color} == \text{WHITE}$ 
7     DFS-VISIT( $G, u$ )
  
```



$\text{DFS-VISIT}(G, u)$

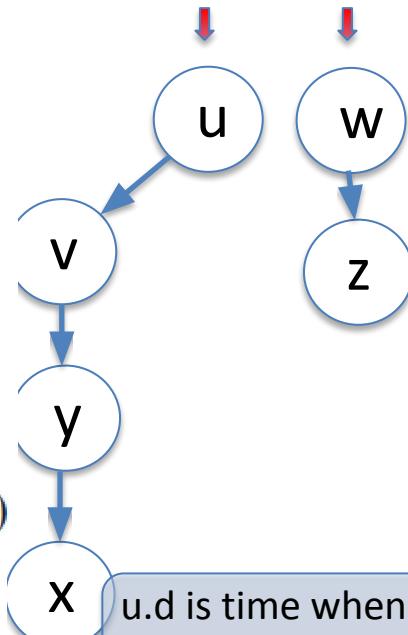
```

1  $time = time + 1$ 
2  $u.d = time$ 
3  $u.\text{color} = \text{GRAY}$  // white vertex  $u$  has just been discovered
4 for each  $v \in G.\text{Adj}[u]$ 
5   if  $v.\text{color} == \text{WHITE}$ 
6      $v.\pi = u$ 
7     DFS-VISIT( $G, v$ ) // explore edge  $(u, v)$ 
8    $u.\text{color} = \text{BLACK}$  // blacken  $u$ ; it is finished
9    $time = time + 1$ 
10   $u.f = time$ 
  
```



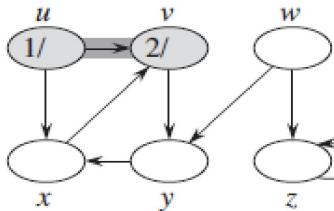
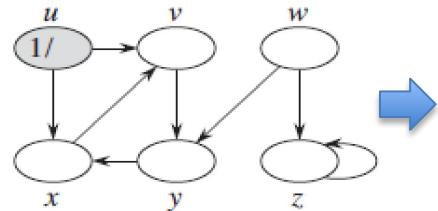
10 $u.f = time$

$u.f$ is time when DFS from u is finished



$u.d$ is time when u is discovered

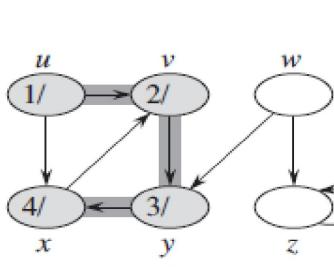
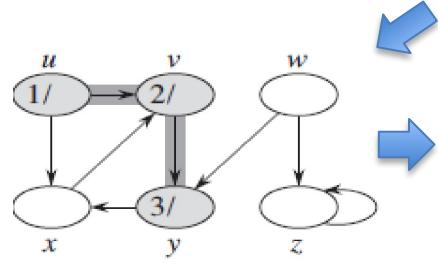
Depth-First Search



$\text{DFS}(G)$

```

1 for each vertex  $u \in G.V$ 
2    $u.\text{color} = \text{WHITE}$ 
3    $u.\pi = \text{NIL}$ 
4    $time = 0$ 
5 for each vertex  $u \in G.V$ 
6   if  $u.\text{color} == \text{WHITE}$ 
7      $\text{DFS-VISIT}(G, u)$ 
  
```



$\text{DFS-VISIT}(G, u)$

```

1  $time = time + 1$ 
2  $u.d = time$ 
3  $u.\text{color} = \text{GRAY}$  // white vertex  $u$  has just been discovered
4 for each  $v \in G.\text{Adj}[u]$ 
5   if  $v.\text{color} == \text{WHITE}$ 
6      $v.\pi = u$ 
7      $\text{DFS-VISIT}(G, v)$  // explore edge  $(u, v)$ 
8    $u.\text{color} = \text{BLACK}$  // blacken  $u$ ; it is finished
9    $time = time + 1$ 
10   $u.f = time$ 
  
```



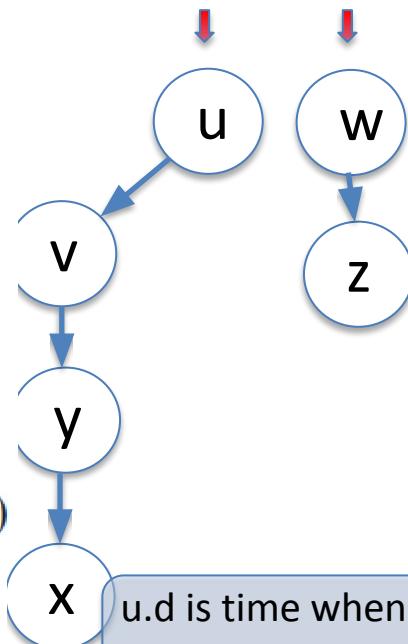
$\text{DFS-VISIT}(G, u)$

```

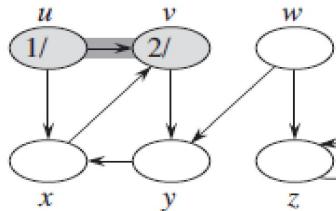
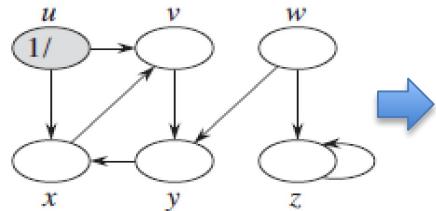
1  $time = time + 1$ 
2  $u.d = time$ 
3  $u.\text{color} = \text{GRAY}$  // white vertex  $u$  has just been discovered
4 for each  $v \in G.\text{Adj}[u]$ 
5   if  $v.\text{color} == \text{WHITE}$ 
6      $v.\pi = u$ 
7      $\text{DFS-VISIT}(G, v)$  // explore edge  $(u, v)$ 
8    $u.\text{color} = \text{BLACK}$  // blacken  $u$ ; it is finished
9    $time = time + 1$ 
10   $u.f = time$ 
  
```



$u.f$ is time when DFS from u is finished



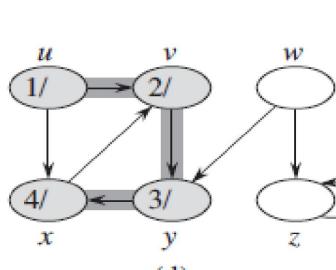
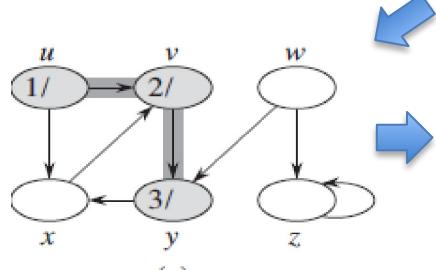
Depth-First Search



$\text{DFS}(G)$

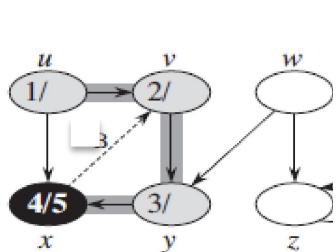
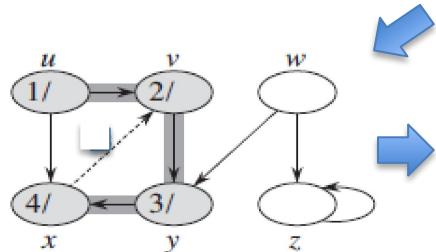
```

1 for each vertex  $u \in G.V$ 
2    $u.\text{color} = \text{WHITE}$ 
3    $u.\pi = \text{NIL}$ 
4    $time = 0$ 
5 for each vertex  $u \in G.V$ 
6   if  $u.\text{color} == \text{WHITE}$ 
7     DFS-VISIT( $G, u$ )
  
```



```

1 for each vertex  $u \in G.V$ 
2   if  $u.\text{color} == \text{WHITE}$ 
3     DFS-VISIT( $G, u$ )
  
```

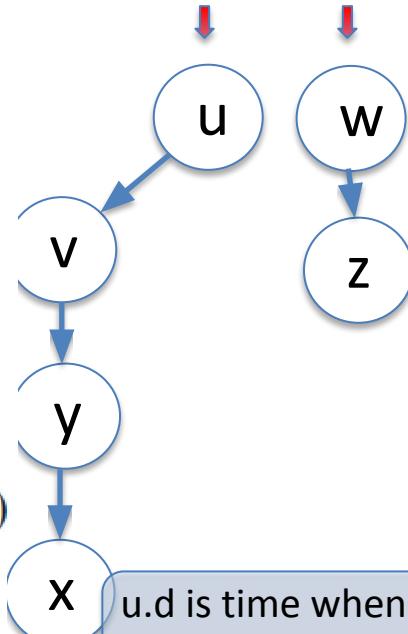


$\text{DFS-VISIT}(G, u)$

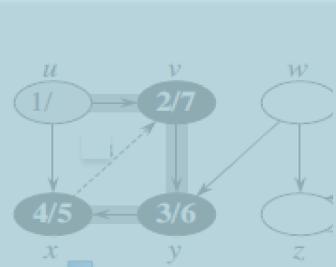
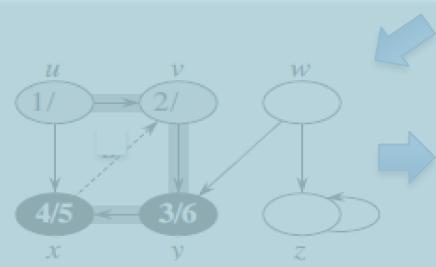
```

1 time = time + 1
2 u.d = time
3 u.color = GRAY // white vertex  $u$  has just been discovered
4 for each  $v \in G.\text{Adj}[u]$ 
5   if  $v.\text{color} == \text{WHITE}$ 
6      $v.\pi = u$ 
7     DFS-VISIT( $G, v$ ) // explore edge  $(u, v)$ 
8   u.color = BLACK // blacken  $u$ ; it is finished
9   time = time + 1
10  u.f = time
  
```

u.f is time when DFS from u is finished

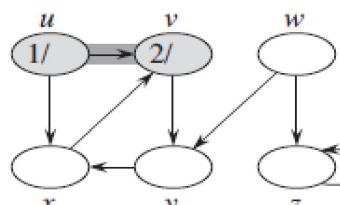
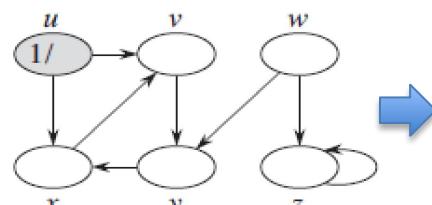


u.d is time when u is discovered



10

Depth-First Search

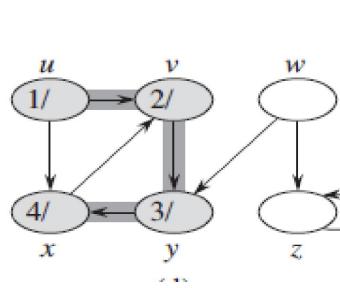
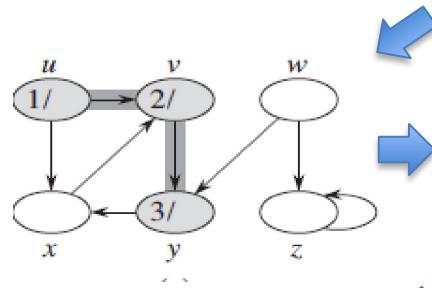


$\text{DFS}(G)$

```

1   for each vertex  $u \in G.V$ 
2        $u.color = \text{WHITE}$ 
3        $u.\pi = \text{NIL}$ 

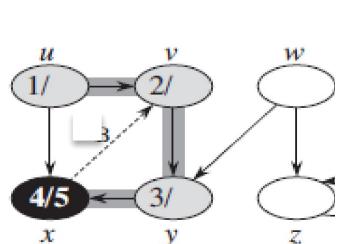
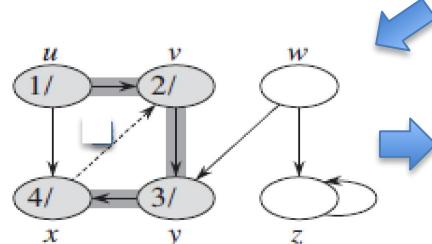
```



```

5   for each vertex  $u \in G.V$ 
6     if  $u.color == \text{WHITE}$ 
7       DFS-VISIT( $G, u$ )

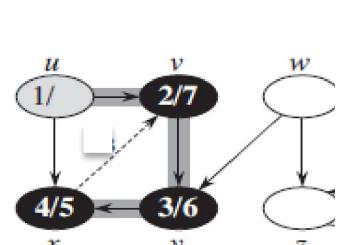
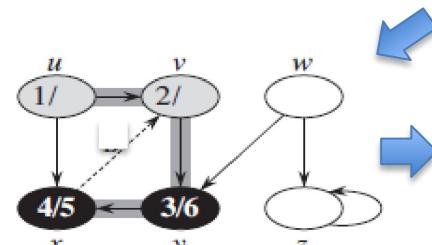
```



DFS-VISIT(G, u)

1 *time* = *time* + 1

- 2 $u.d = \text{time}$
- 3 $u.\text{color} = \text{GRAY} //$ white vertex u has just been discovered
- 4 **for** each $v \in G.\text{Adj}[u]$



5 if $v.color == \text{WHITE}$

$$6 \qquad \qquad v.\pi = u$$

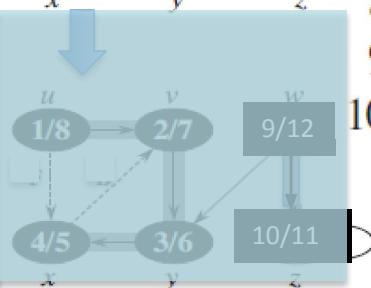
7 DFS-VISIT(G, v) // explore edge (u, v)

8 *u.color* = BLACK

9 *time* \equiv *time* + 1 // slacken it; it is finished

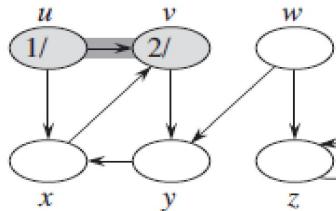
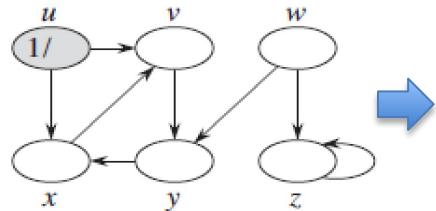
10 $u.f \equiv \text{time}$

DFS fom u is



u.f is time when
DFS from u is
finished

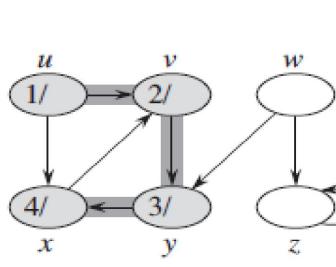
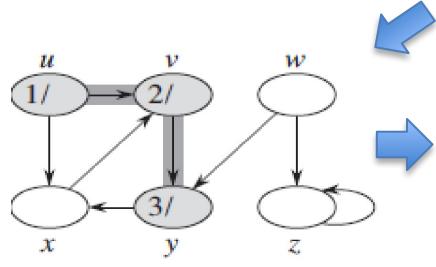
Depth-First Search



$\text{DFS}(G)$

```

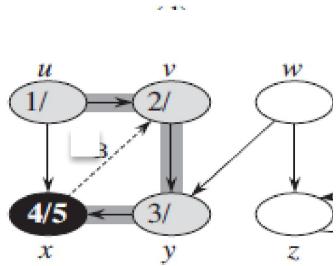
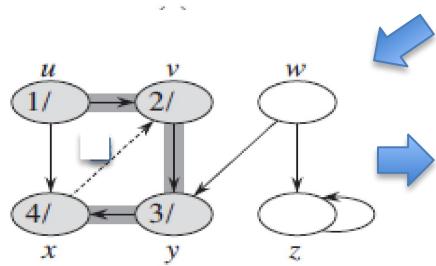
1 for each vertex  $u \in G.V$ 
2    $u.\text{color} = \text{WHITE}$ 
3    $u.\pi = \text{NIL}$ 
4    $time = 0$ 
5 for each vertex  $u \in G.V$ 
6   if  $u.\text{color} == \text{WHITE}$ 
7      $\text{DFS-VISIT}(G, u)$ 
  
```



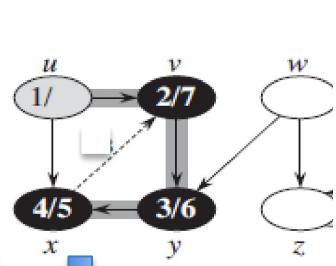
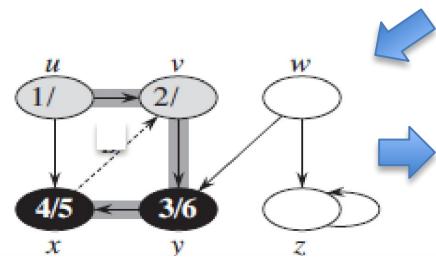
$\text{DFS-VISIT}(G, u)$

```

1    $time = time + 1$ 
2    $u.d = time$ 
3    $u.\text{color} = \text{GRAY} //$  white vertex  $u$  has just been discovered
4   for each  $v \in G.\text{Adj}[u]$ 
5     if  $v.\text{color} == \text{WHITE}$ 
6        $v.\pi = u$ 
7        $\text{DFS-VISIT}(G, v) //$  explore edge  $(u, v)$ 
8      $u.\text{color} = \text{BLACK} //$  blacken  $u$ ; it is finished
9    $time = time + 1$ 
10   $u.f = time$ 
  
```



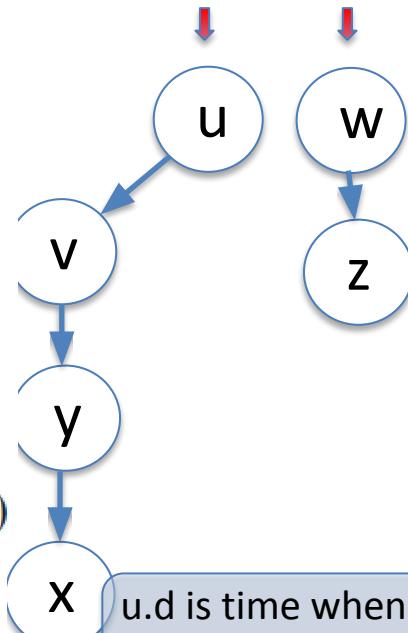
$\text{DFS-VISIT}(G, u)$



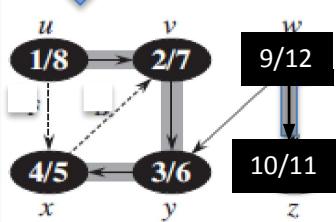
$\text{DFS-VISIT}(G, u)$

```

1    $time = time + 1$ 
2    $u.d = time$ 
3    $u.\text{color} = \text{GRAY} //$  white vertex  $u$  has just been discovered
4   for each  $v \in G.\text{Adj}[u]$ 
5     if  $v.\text{color} == \text{WHITE}$ 
6        $v.\pi = u$ 
7        $\text{DFS-VISIT}(G, v) //$  explore edge  $(u, v)$ 
8      $u.\text{color} = \text{BLACK} //$  blacken  $u$ ; it is finished
9    $time = time + 1$ 
10   $u.f = time$ 
  
```

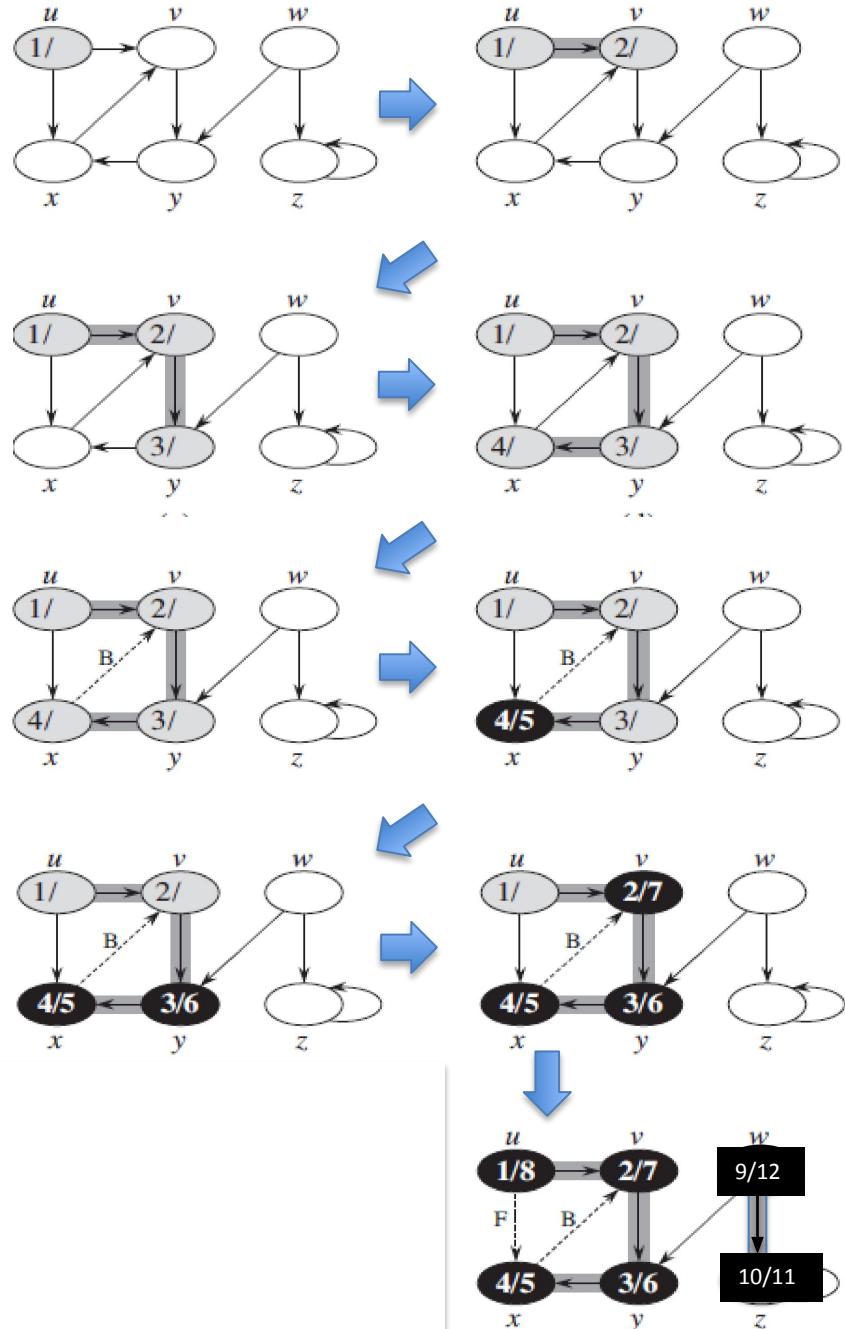


$u.d$ is time when u is discovered



$u.f$ is time when DFS from u is finished

Complexity of DFS: $O(|V| + |E|)$



Max no. of OPS

$O(|V|)$

$O(1)$

$O(|V|)$

$O(1)$

DFS(G)

```

1 for each vertex  $u \in G.V$ 
2    $u.color = \text{WHITE}$ 
3    $u.\pi = \text{NIL}$ 
4    $time = 0$ 
5 for each vertex  $u \in G.V$ 
6   if  $u.color == \text{WHITE}$ 
7     DFS-VISIT( $G, u$ )
  
```

DFS-VISIT(G, u)

```

O(1) →
1  $time = time + 1$ 
2  $u.d = time$ 
3  $u.color = \text{GRAY}$ 
  
```

$\sum \text{out-deg}(u)$

```

O(1) →
4 for each  $v \in G.Adj[u]$ 
5   if  $v.color == \text{WHITE}$ 
6      $v.\pi = u$ 
7     DFS-VISIT( $G, v$ )
  
```

$\sum \text{out-deg}(u)$

```

O(1) →
8  $u.color = \text{BLACK}$ 
9  $time = time + 1$ 
10  $u.f = time$ 
  
```

Complexity of DFS: $O(|V| + |E|)$

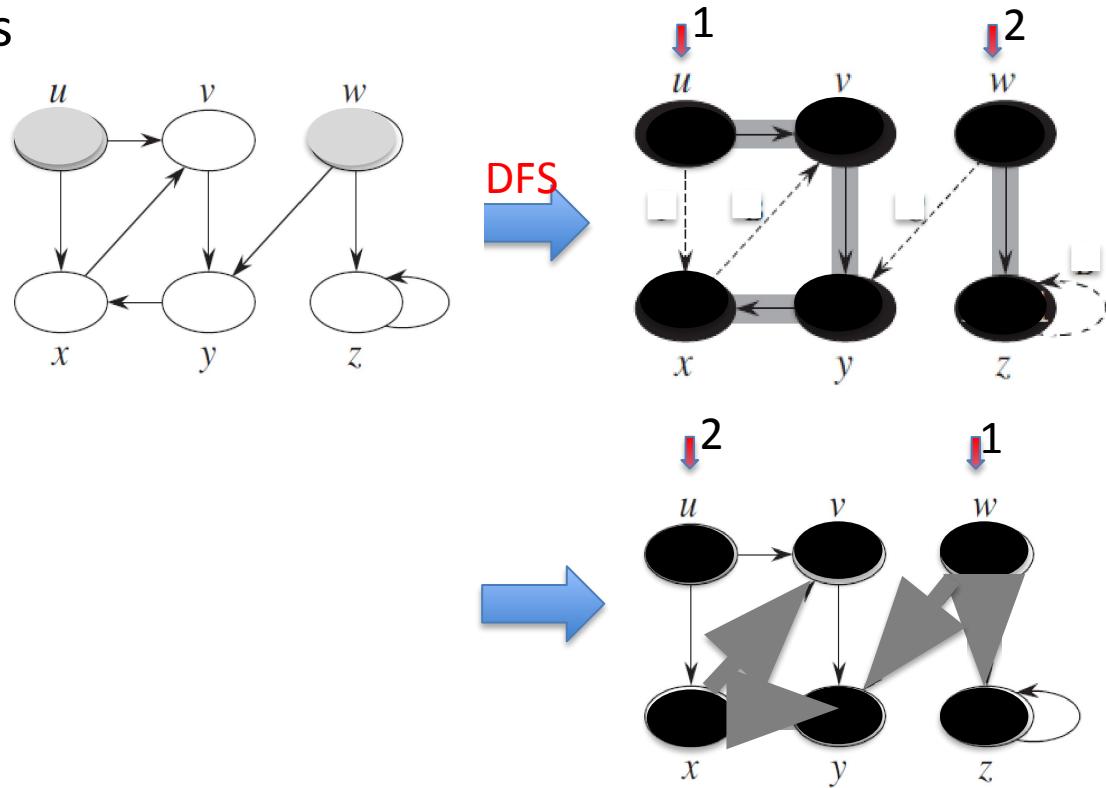
Time complexity of DFS

- Let $G = (V, E)$ be a directed or an undirected graph , and $|V| = n$, $|E| = m$.
- Then DFS of G takes time $O(n + m)$
- How about $\Omega(f(n, m))$ or $\Theta(g(n, m))$?

Properties of DFS

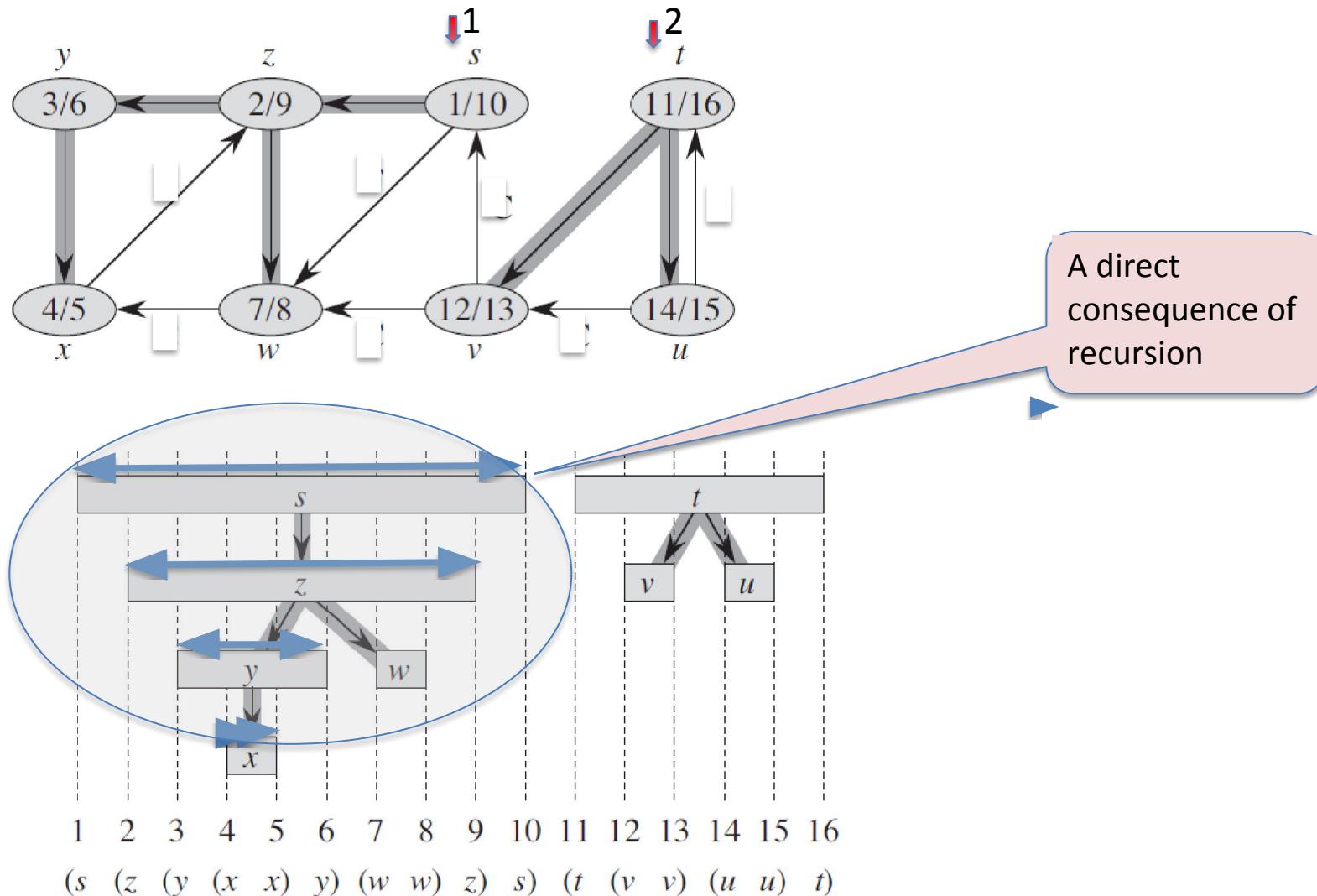
Let G be an directed graph on which **DFS** has been performed. Then:

- DFS results in a forest of trees



Properties of DFS

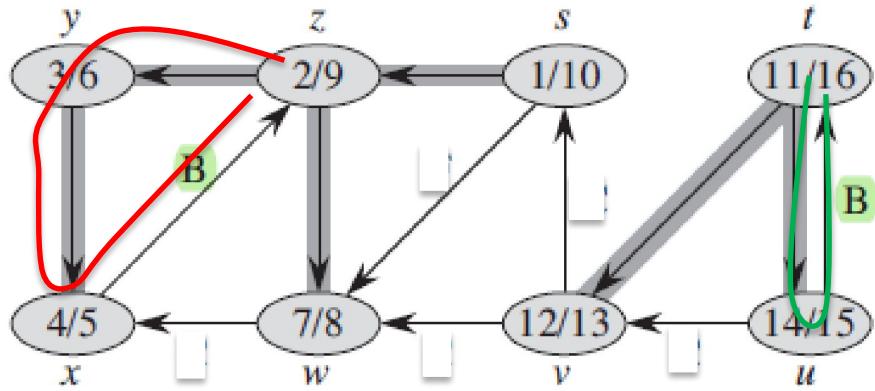
- DFS traversal yields the following forest of trees, together with the start times & finish times:



Properties of DFS

Let G be a directed graph on which **DFS** has been performed. Then:

- A directed cycle can be detected if it exists



$\text{DFS}(G)$

```
1 for each vertex  $u \in G.V$ 
2    $u.\text{color} = \text{WHITE}$ 
3    $u.\pi = \text{NIL}$ 
4    $\text{time} = 0$ 
5 for each vertex  $u \in G.V$ 
6   if  $u.\text{color} == \text{WHITE}$ 
7      $\text{DFS-VISIT}(G, u)$ 
```

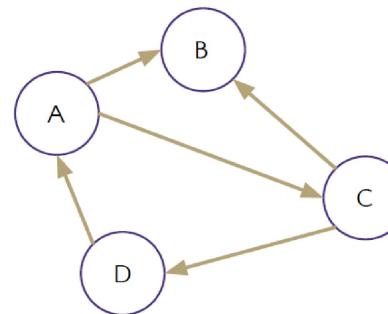
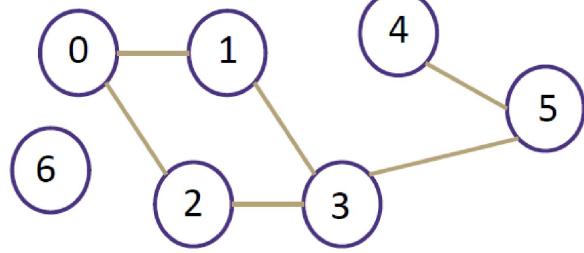
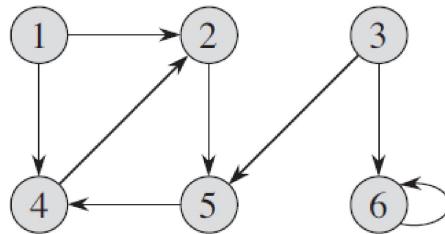
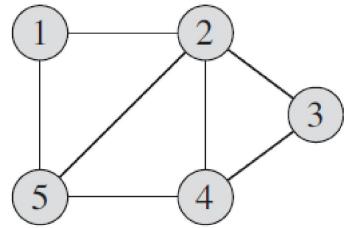
$\text{DFS-VISIT}(G, u)$

```
1  $\text{time} = \text{time} + 1$ 
2  $u.d = \text{time}$ 
3  $u.\text{color} = \text{GRAY} //$  white vertex  $u$  has just been d
4 for each  $v \in G.\text{Adj}[u]$ 
5   if  $v.\text{color} == \text{WHITE}$ 
6      $v.\pi = u$ 
7      $\text{DFS-VISIT}(G, v) //$  explore edge  $(u, v)$ 
8    $u.\text{color} = \text{BLACK} //$  blacken  $u$ ; it is finished
9    $\text{time} = \text{time} + 1$ 
10   $u.f = \text{time}$ 
```

What conclusion can be drawn when colour = Gray?
◻ There is a cycle ◻ the graph is not a DAG

Try these examples

DFS traversal of the following graphs



Q&A