IIT ROORKEE

FREE ONLINE EDUCATION swayam शिक्षित भारत, उन्नत भारत

NPTEL ONLINE CERTIFICATION COURSE

VLSI Physical Design with Timing Analysis

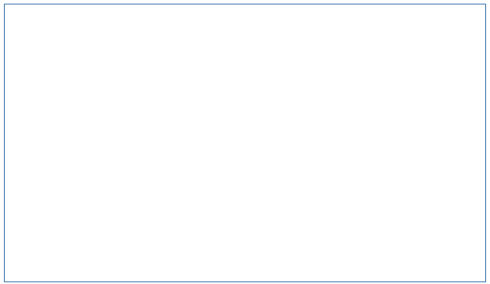
Lecture – 5: Graph Searching Algorithms

Bishnu Prasad Das

Department of Electronics and Communication Engineering

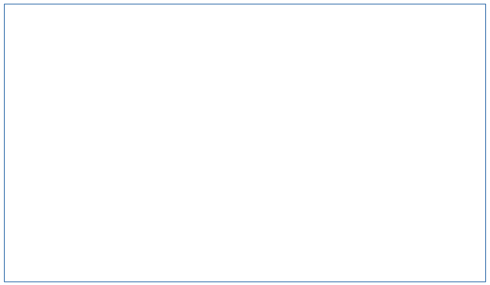
Contents

- Graph Search Algorithms
  - Depth First Search(DFS)
  - Breadth First Search(BFS)



Graph Search Algorithms

- Systematically follow the edges of the graph to visit the vertices of the graph.
- Used to discover the structure of the graph.
- There are two popular algorithms for graph search:
  - Breadth First Search(BFS)
  - Depth First Search(DFS)



Depth First Search

1. Create a stack to keep track of vertices to visit.
2. Choose a starting vertex and push it onto the stack.
3. While the stack is not empty, do the following:
   a. Check if the vertex on the top of the stack has any unvisited adjacent vertices.
   b. If yes, push an unvisited adjacent vertex onto the stack.
   c. If no unvisited adjacent vertices are left for the current vertex, pop it from the stack.
4. Repeat steps 2 and 3 until the stack is empty, which means all vertices have been visited.

Depth First Search

- Algorithm:
  - DFS(G)
    - **for** each vertex u ∈ G.V
      - u.color = WHITE
      - u.π = NIL
    - time = 0
    - **for** each vertex u ∈ G.V
      - **if** u.color == WHITE
        - » DFS-VISIT(G,u)

G(V, E)

Paint all vertices white and initialize their attributes to NIL.

Check each vertex in V, and when a white vertex is found, visit it by calling DFS-VISIT.

Depth First Search

- 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. π = u
  - DFS-VISIT(G,v)
- time = time + 1
- u.f = time
- u.color = BLACK

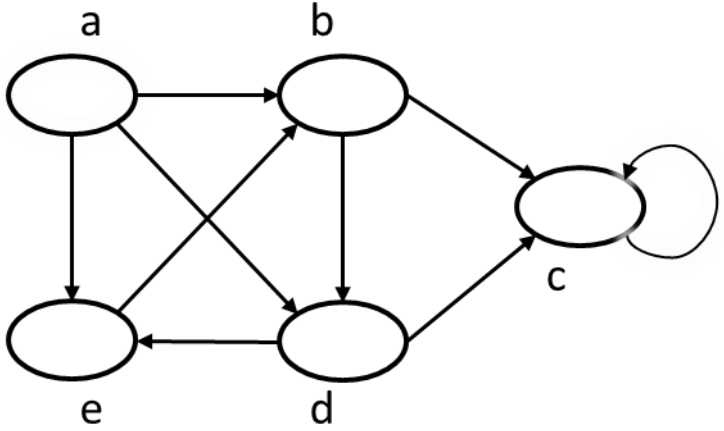
Increment the global variable time, record the new value of time as the discovery time **u.d**, and paint u **GRAY**.

examine each vertex **v** adjacent to **u** and recursively visit **v** if it is white.

Finally, after every edge leaving **u** has been explored, increment time, record the finish time in **u.f** ,and paint **u BLACK**.

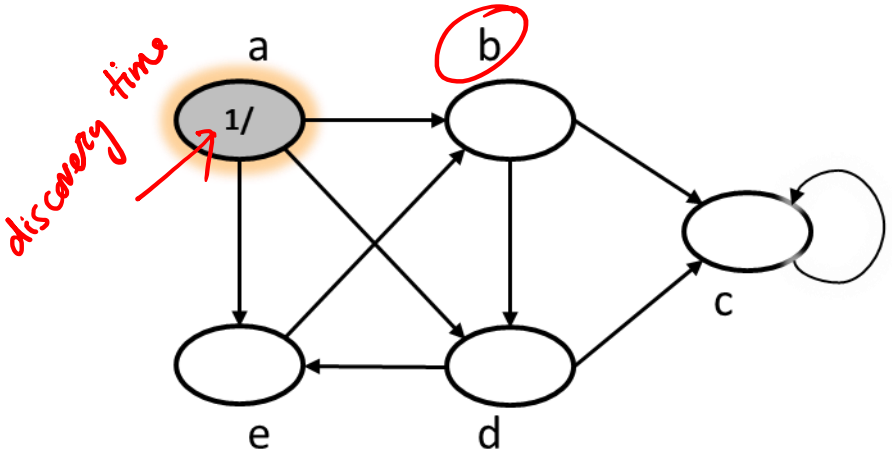
DFS: Example - 1

$G = (V, E)$ $V := \{a, b, c, d, e\}$



Input Graph

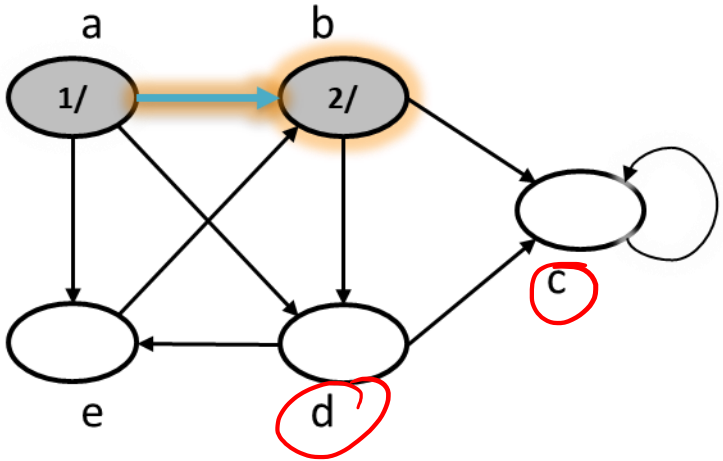
DFS: Example - 1



Stack
a

DFS: Example - 1

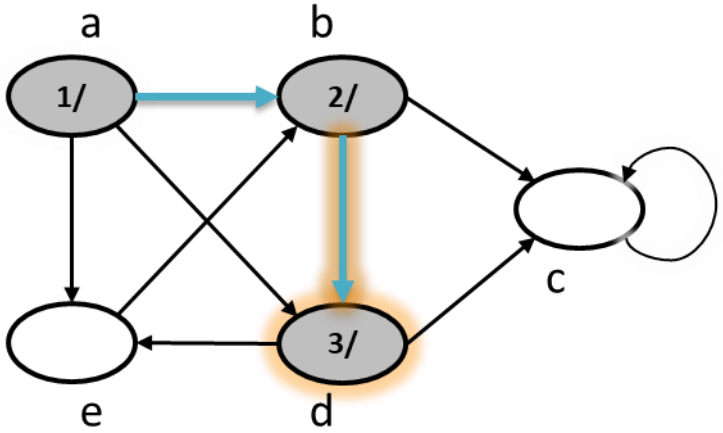
LIFO Data structure



b
a

Stack

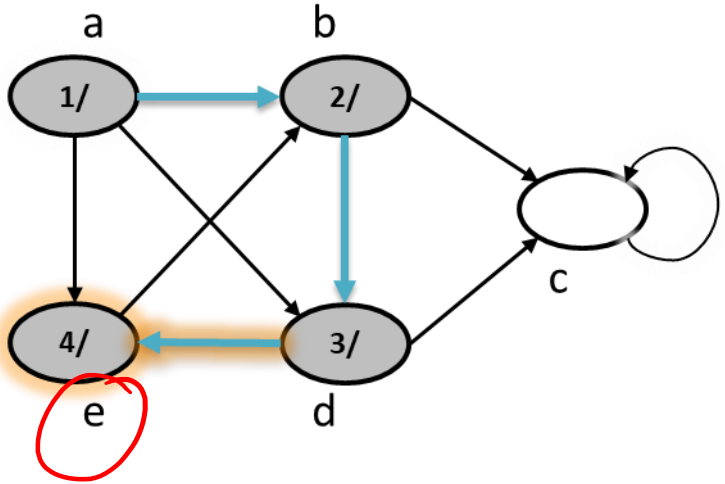
DFS: Example - 1



d
b
a

Stack

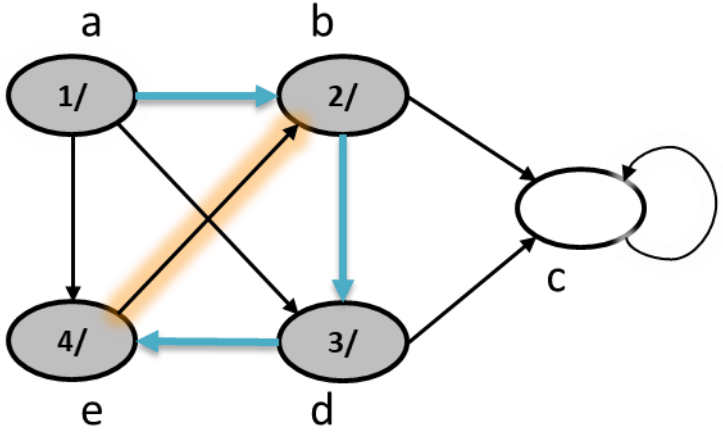
Example - 1



e
d
b
a

Stack

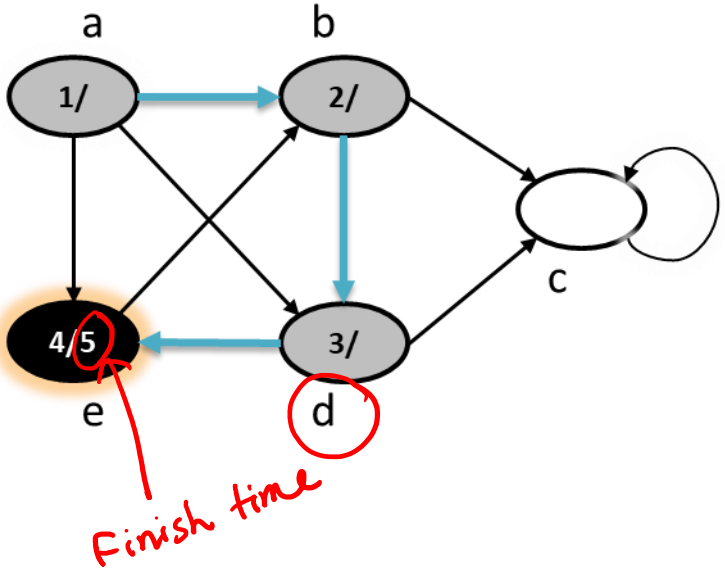
DFS: Example - 1



e
d
b
a

Stack

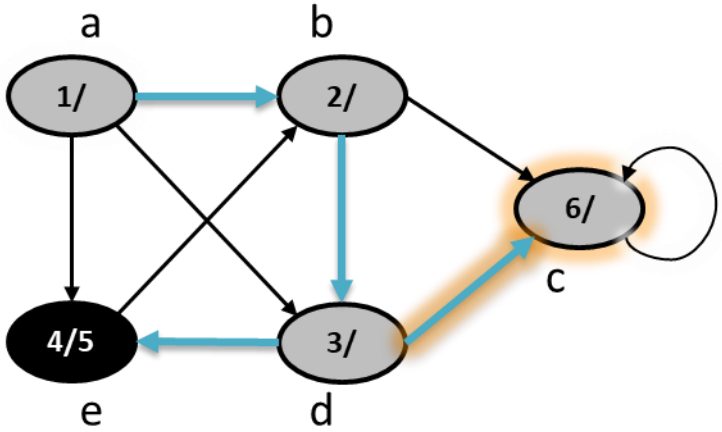
DFS: Example - 1



d
b
a

Stack

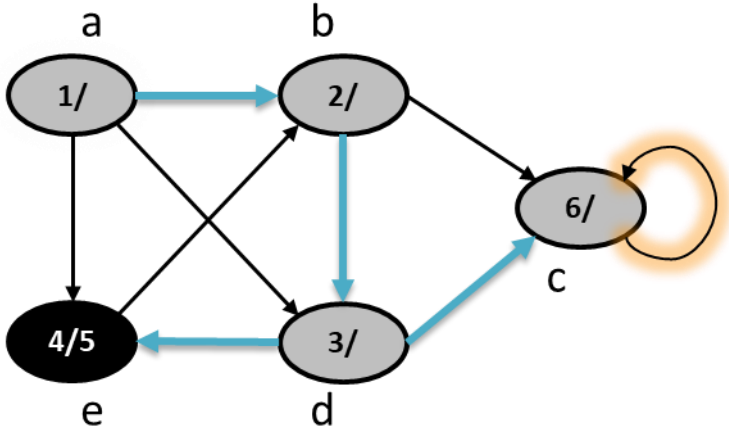
DFS: Example - 1



c
d
b
a

Stack

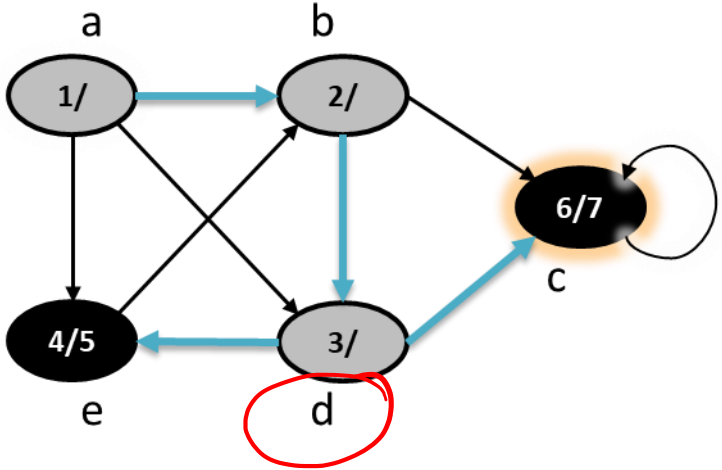
DFS: Example - 1

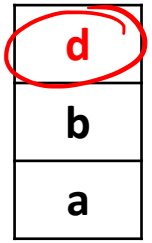


c
d
b
a

Stack

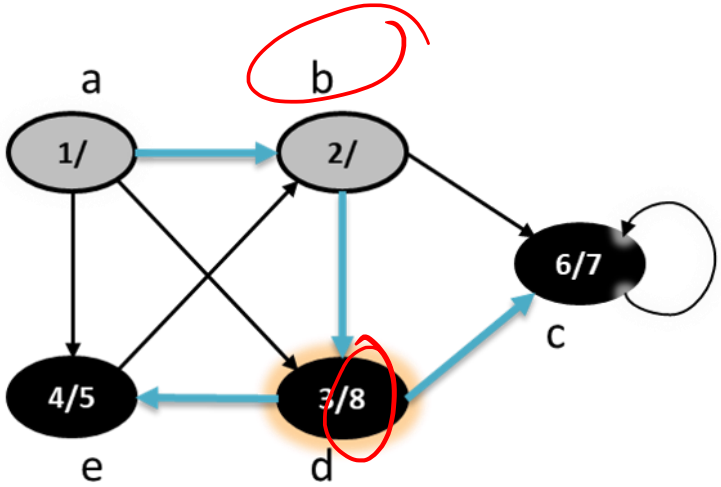
DFS: Example - 1

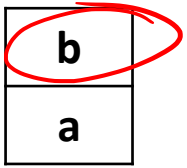




Stack

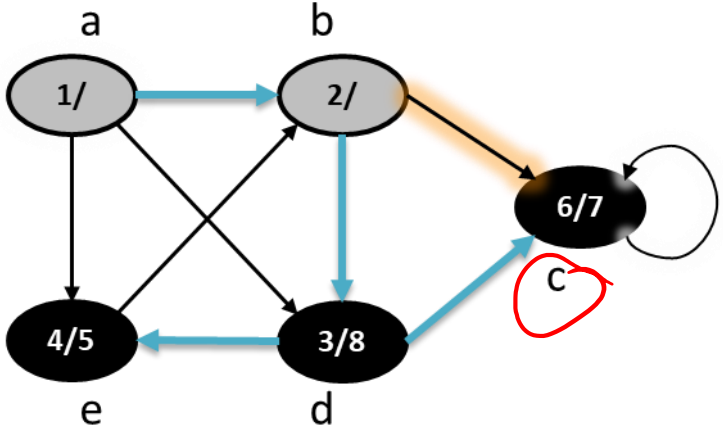
DFS: Example - 1

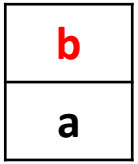




Stack

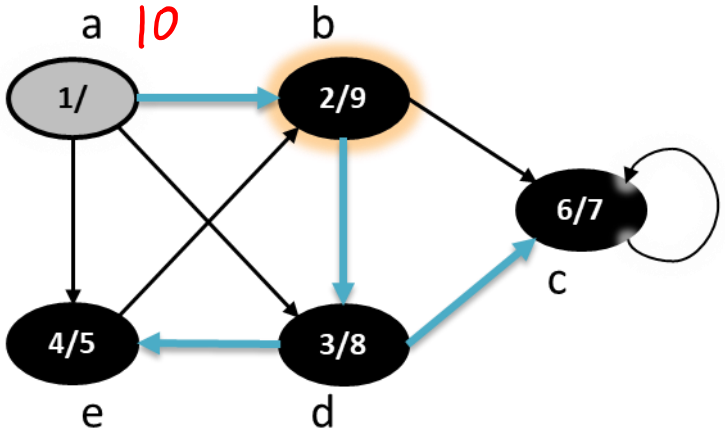
DFS: Example - 1

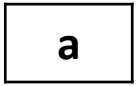




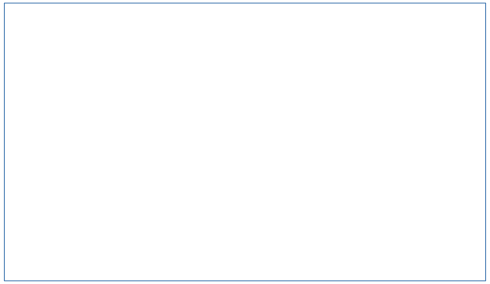
Stack

DFS: Example - 1

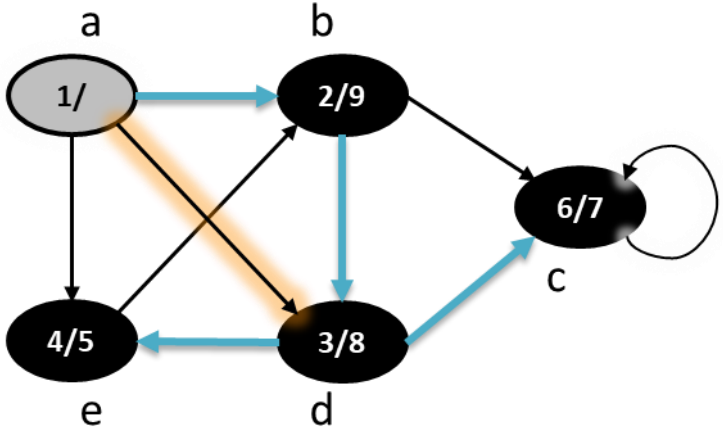


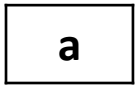


Stack

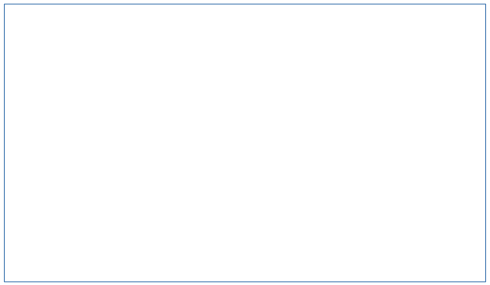


DFS: Example - 1

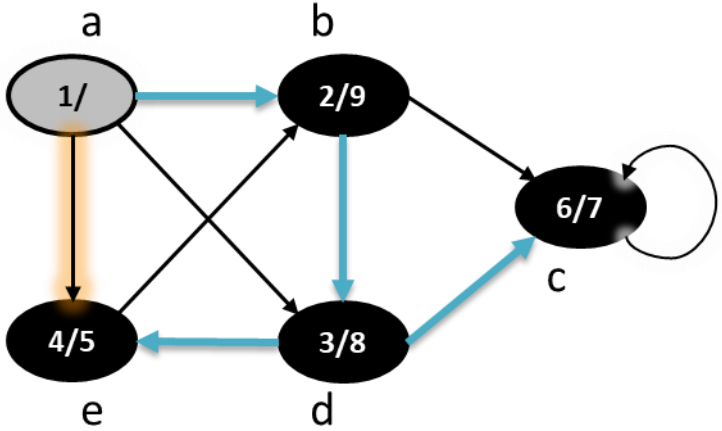




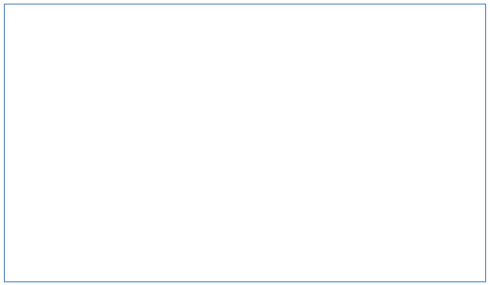
Stack



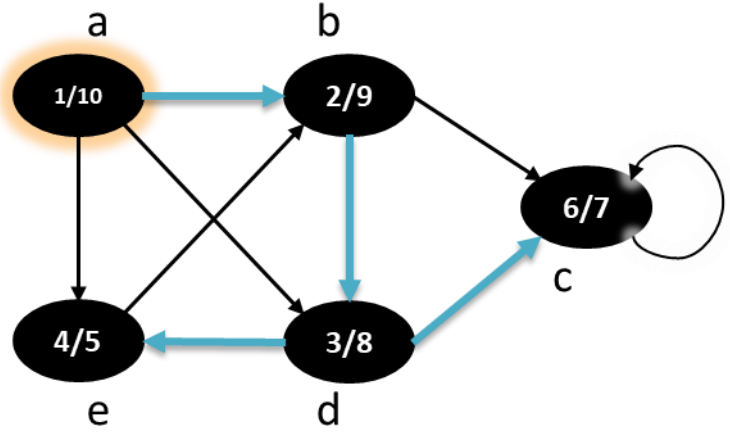
DFS: Example - 1



Stack

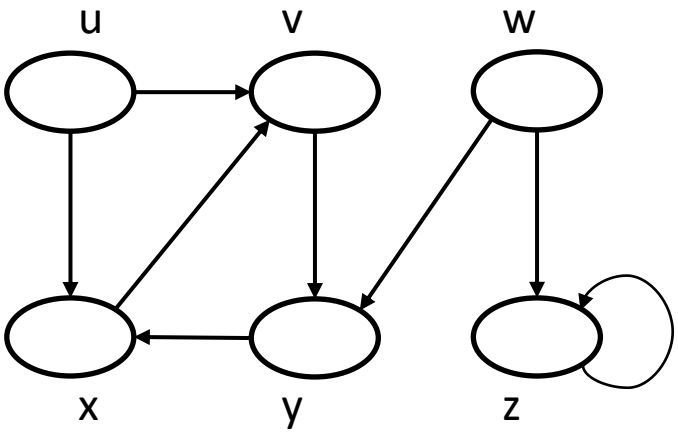


DFS: Example - 1



Stack

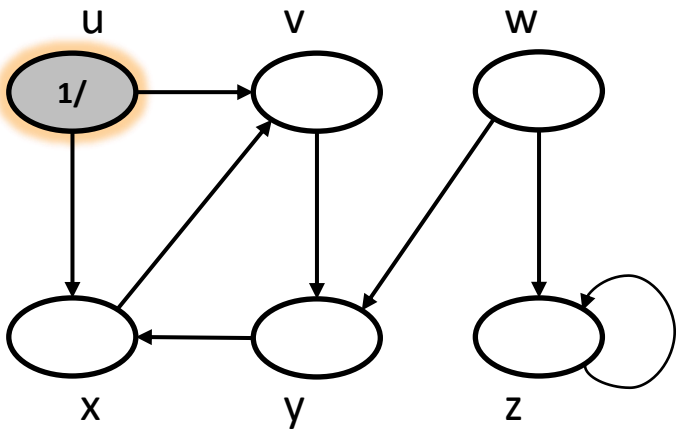
DFS: Example - 2

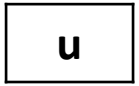


Input Graph

Source :Cormen, Leiserson, Rivest, Stein “Introduction to Algorithms”

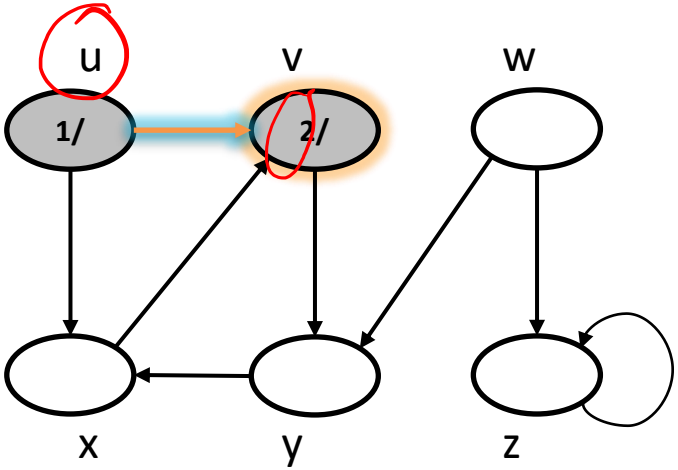
DFS: Example - 2

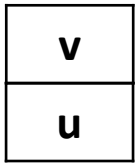




Stack

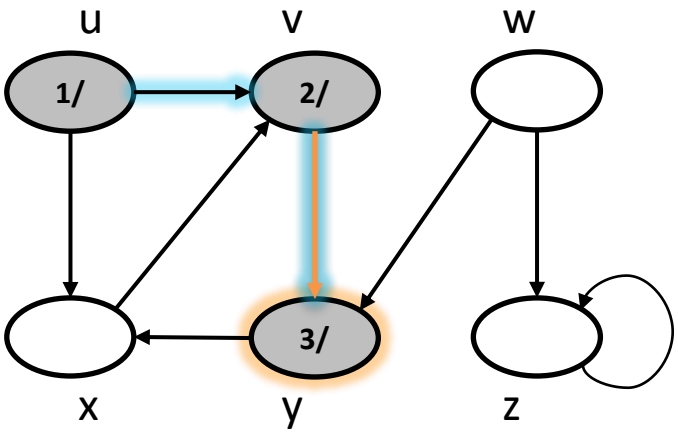
DFS: Example - 2

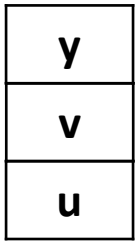




Stack

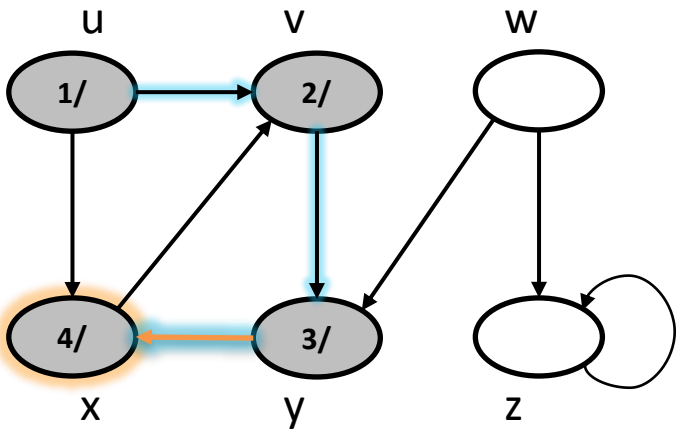
DFS: Example - 2

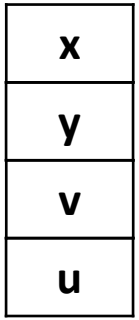




Stack

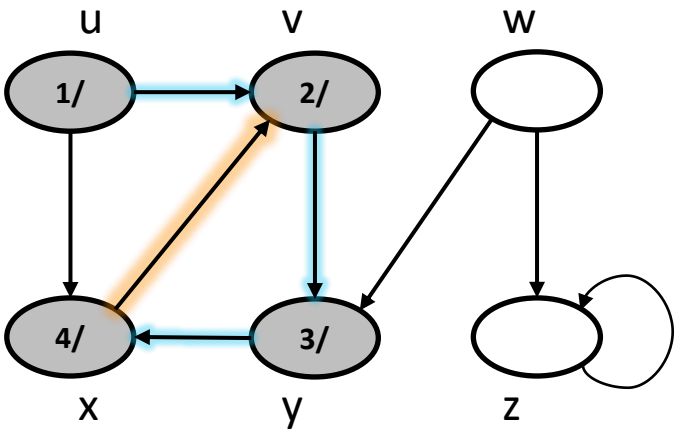
DFS: Example - 2

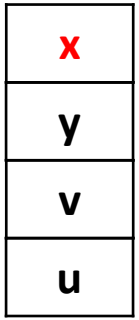




Stack

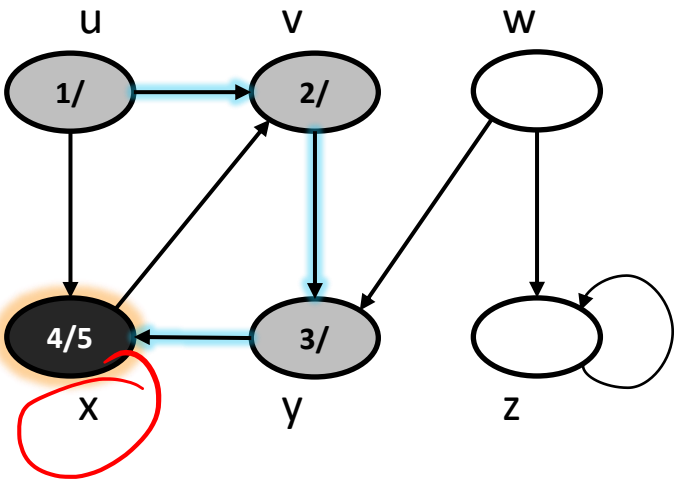
DFS: Example - 2

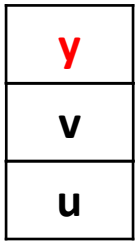




Stack

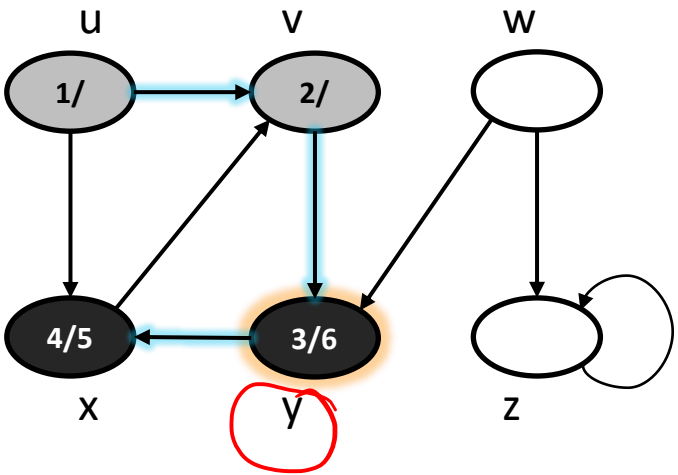
DFS: Example - 2

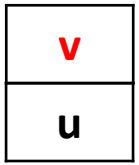




Stack

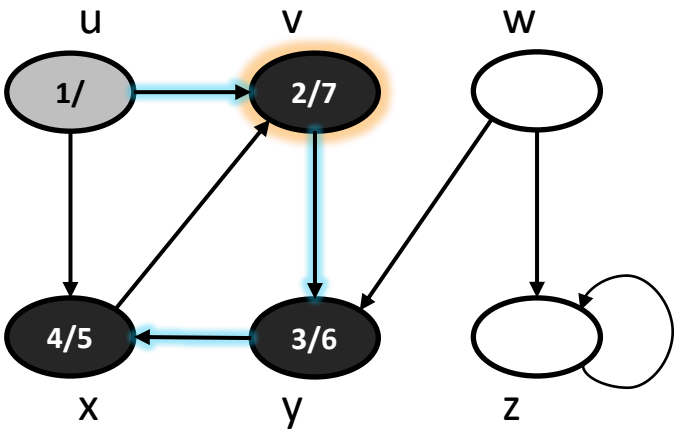
DFS: Example - 2

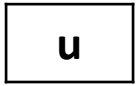




Stack

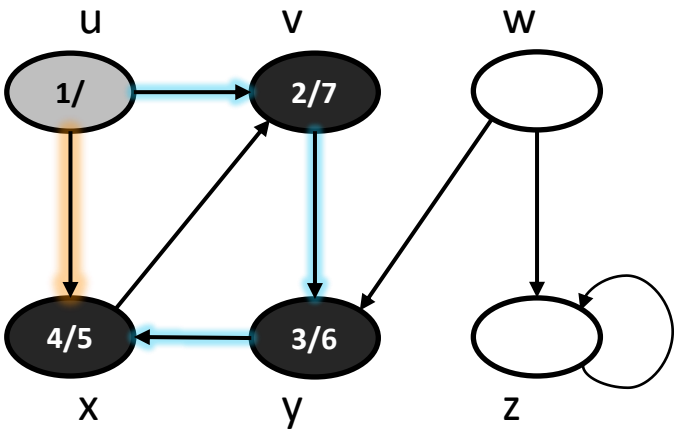
DFS: Example - 2

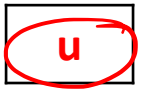




Stack

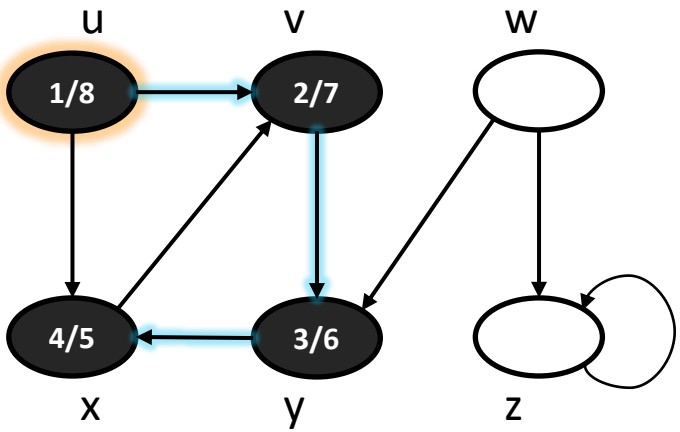
DFS: Example - 2





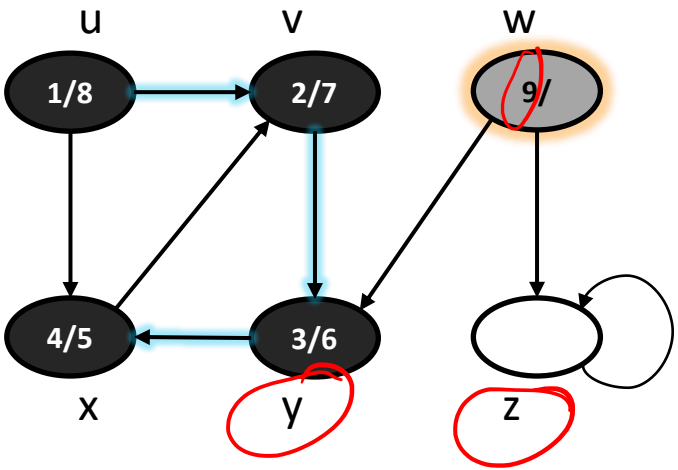
Stack

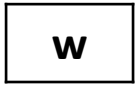
DFS: Example - 2



Stack

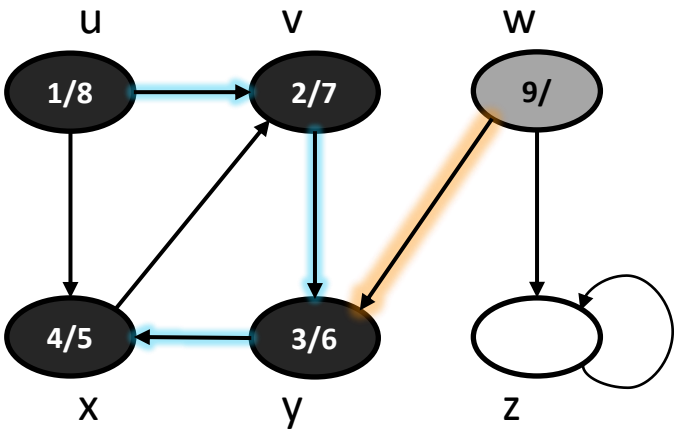
DFS: Example - 2

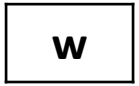




Stack

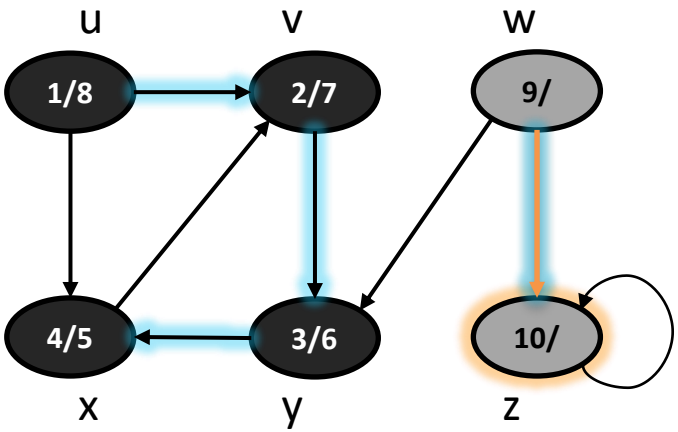
DFS: Example - 2

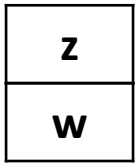




Stack

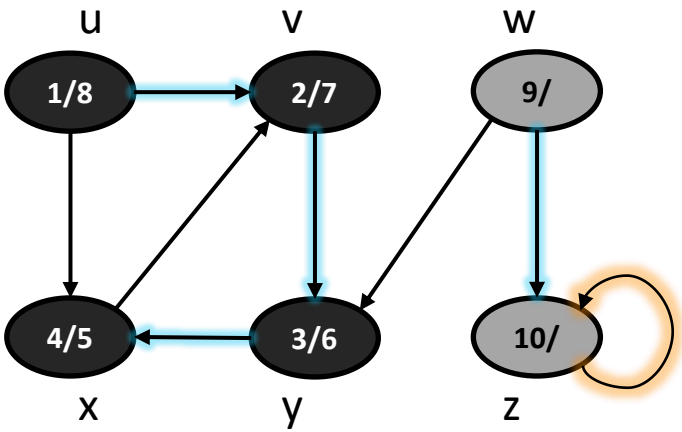
DFS: Example - 2

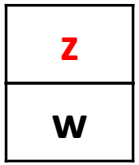




Stack

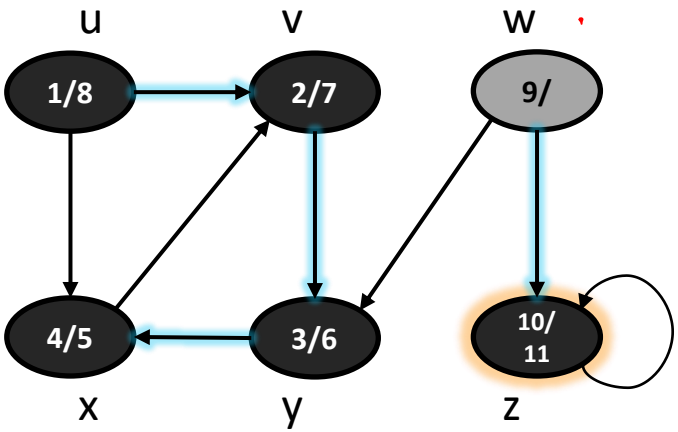
DFS: Example - 2





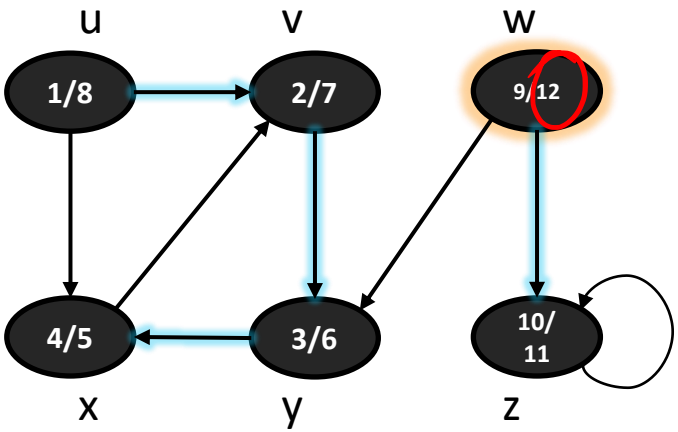
Stack

DFS: Example - 2



Stack

DFS: Example - 2



Stack

Breadth First Search(BFS)

- BFS(G, s)
  - for each vertex u ϵ G.V – {s}
    - u.color = WHITE
    - u.d = ∞
    - u.π = NIL
  - s.color = GRAY
  - s.d = 0
  - s. π = NIL
  - Q = Ø
  - ENQUEUE(Q, s)

With the exception of the source vertex s, paint every vertex white, set u.d = 1 for each vertex u, and set the parent of every vertex to be NIL.

Because the source vertex s is always the first vertex discovered, paint s gray, set s.d to 0, and set the predecessor of s to NIL.

Create the queue Q, initially containing just the source vertex.

Breadth First Search(BFS)

- while Q ≠ Ø
  - U = DEQUEUE(Q)
  - for each vertex v in G.Adj[u]
    - if v.color = WHITE
      - » v.color = GRAY
      - » v.d = u.d + 1
      - » v. π = u
      - » ENQUEUE(Q, v)
  - u.color = BLACK

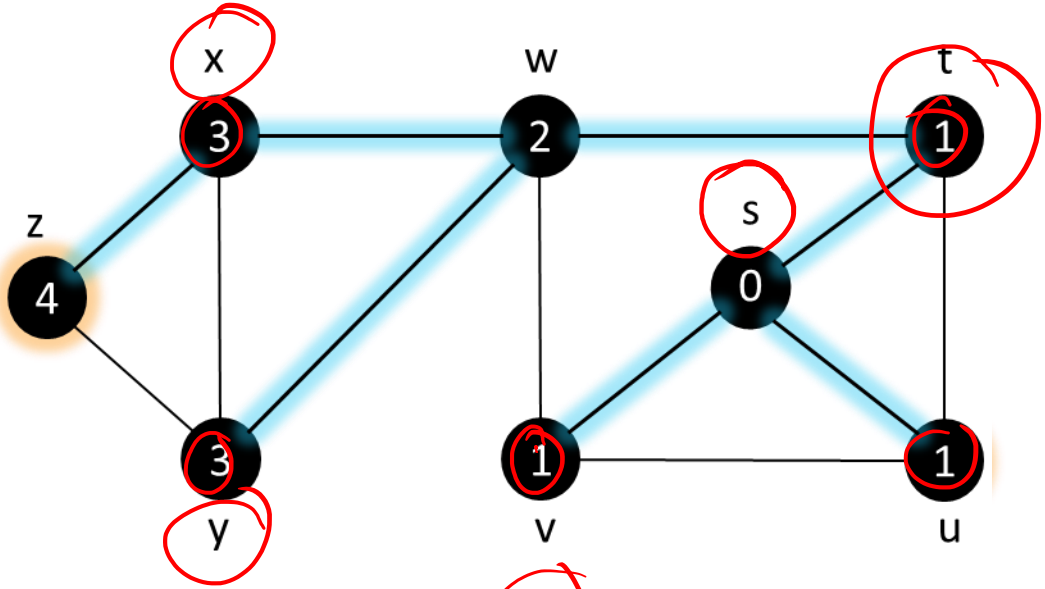
Determines the grey vertex u at the head of the queue Q and removes it from Q.

If **v** is white, then it has not yet been discovered. Paint vertex **v** gray, set **v**'s distance v.d to u.d + 1, record **u** as **v**'s parent **v**. π and place **v** at the tail of the queue Q.

Once the procedure has examined all the vertices on u's adjacency list, blacken u, indicating that u is now behind the frontier. .

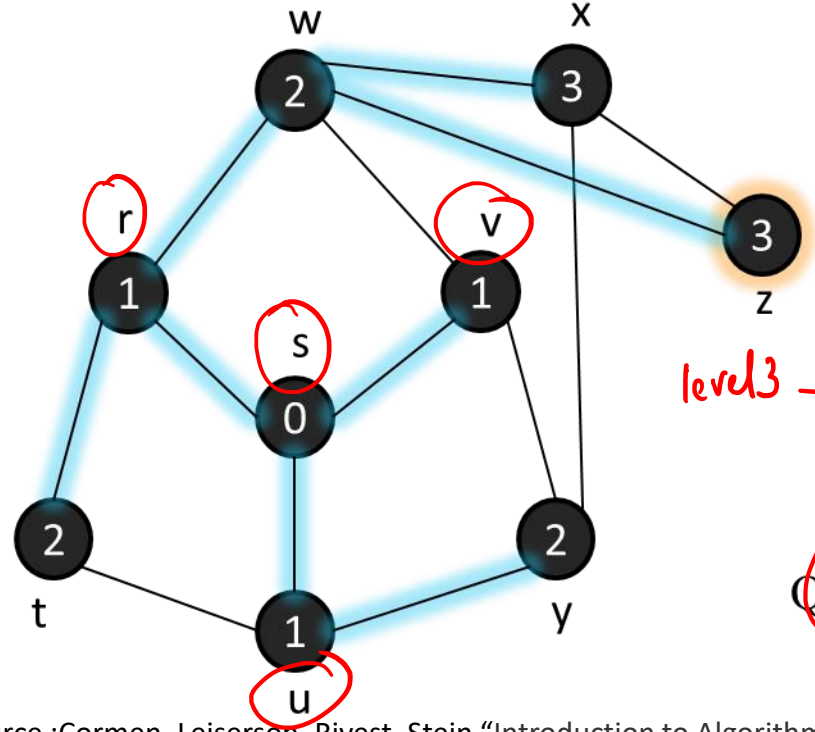
BFS: Example - 1

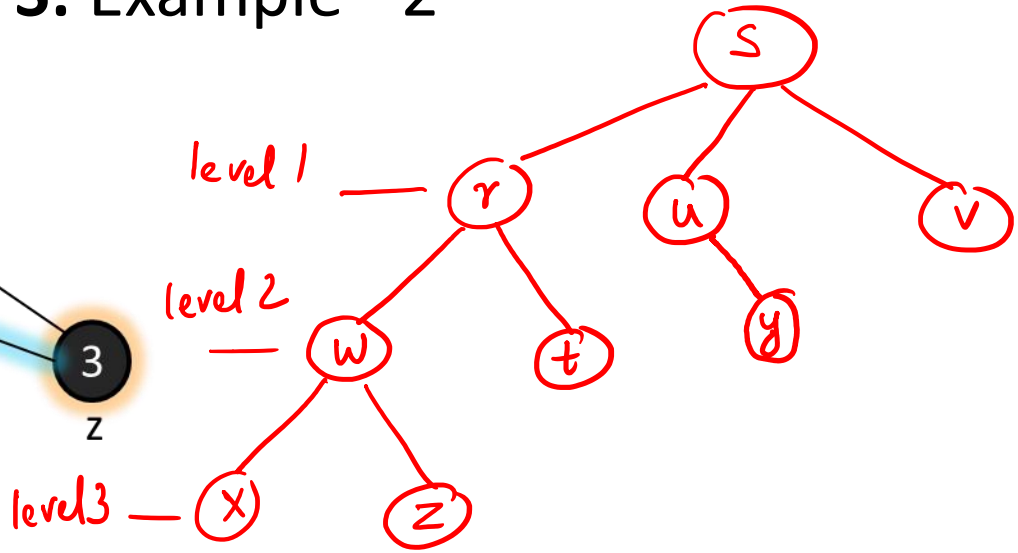
FIFO Data Structure



$Q = \emptyset$

BFS: Example - 2



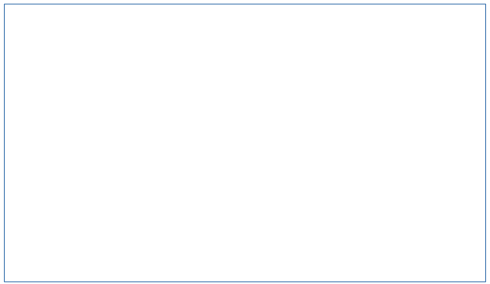


$Q = \emptyset$

Source :Cormen, Leiserson, Rivest, Stein "Introduction to Algorithms"

Summary

- Discussed Graph search algorithms: DFS and BFS



Thank You