Algorithms

Graph Searching Techniques - BFS

Graph Searching

- Given: a graph G = (V, E), directed or undirected
- Goal: methodically explore every vertex and every edge
- Ultimately: build a tree on the graph
    - Pick a vertex as the root
    - Choose certain edges to produce a tree
    - Note: might also build a *forest* if graph is not connected

- There are two standard graph traversal techniques:
    - Breadth-First Search (BFS)
    - Depth-First Search (DFS)

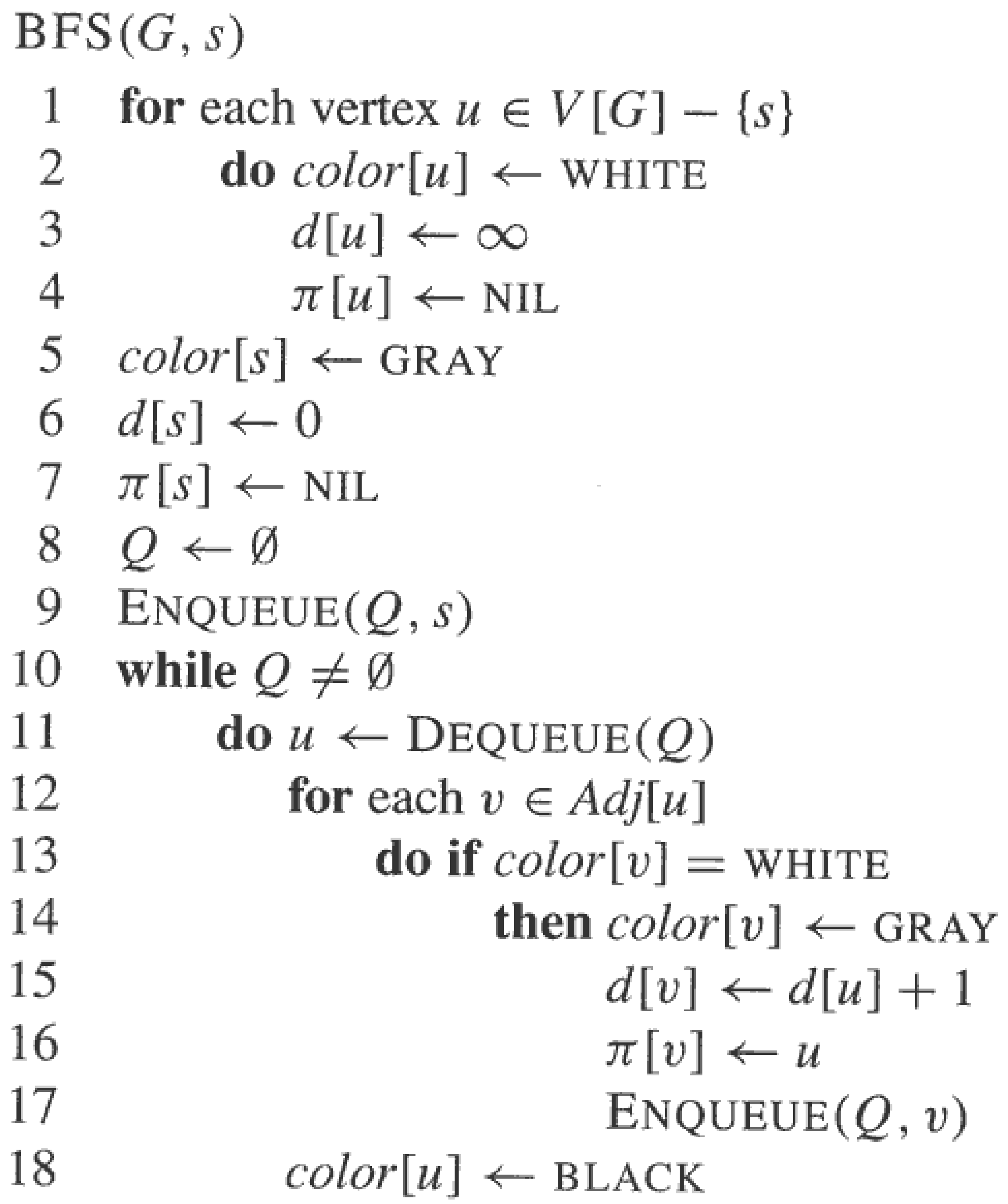
Breadth-First Search

- "Explore" a graph, turning it into a tree
  - One vertex at a time
  - Expand frontier of explored vertices across the *breadth* of the frontier
- Builds a tree over the graph
  - Pick a *source vertex* to be the root
  - Find ("discover") its children, then their children, etc.

Breadth-First Search

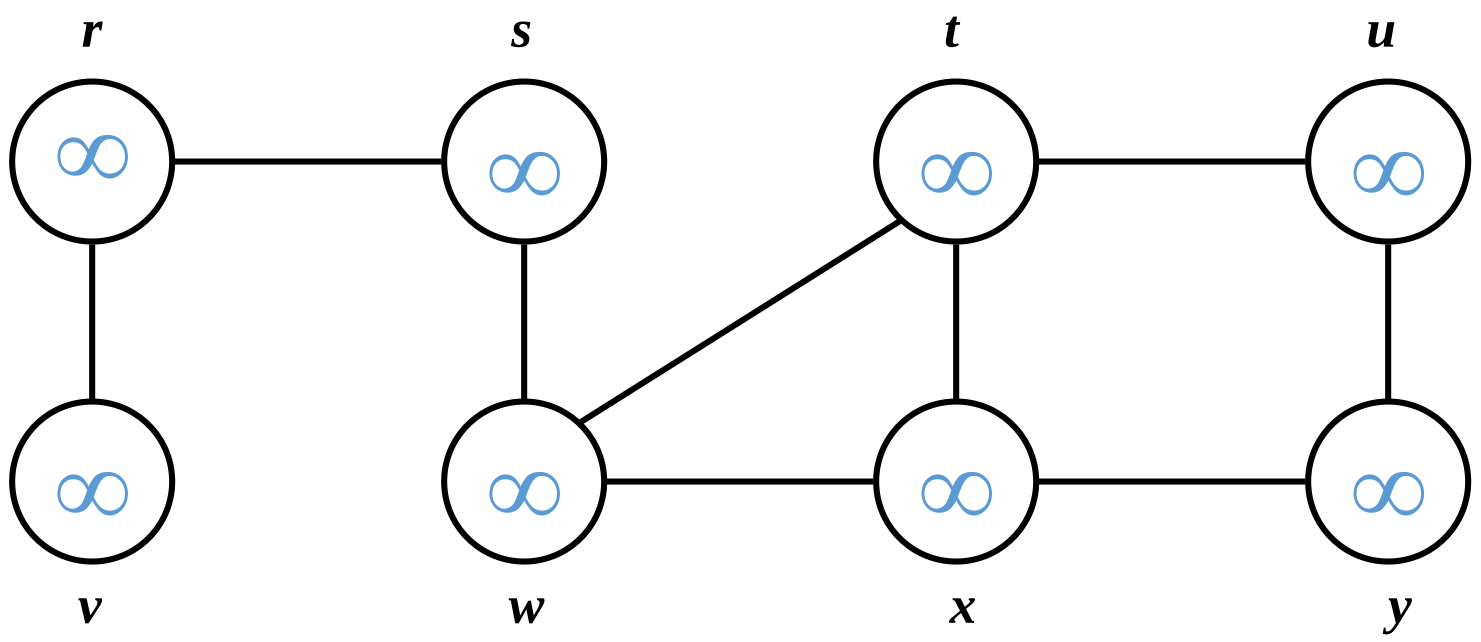
- Again will associate vertex “colors” to guide the algorithm
  - White vertices have not been discovered
    - All vertices start out white
  - Grey vertices are discovered but not fully explored
    - They may be adjacent to white vertices
  - Black vertices are discovered and fully explored
    - They are adjacent only to black and grey vertices
- Explore vertices by scanning adjacency list of grey vertices

Breadth-First Search

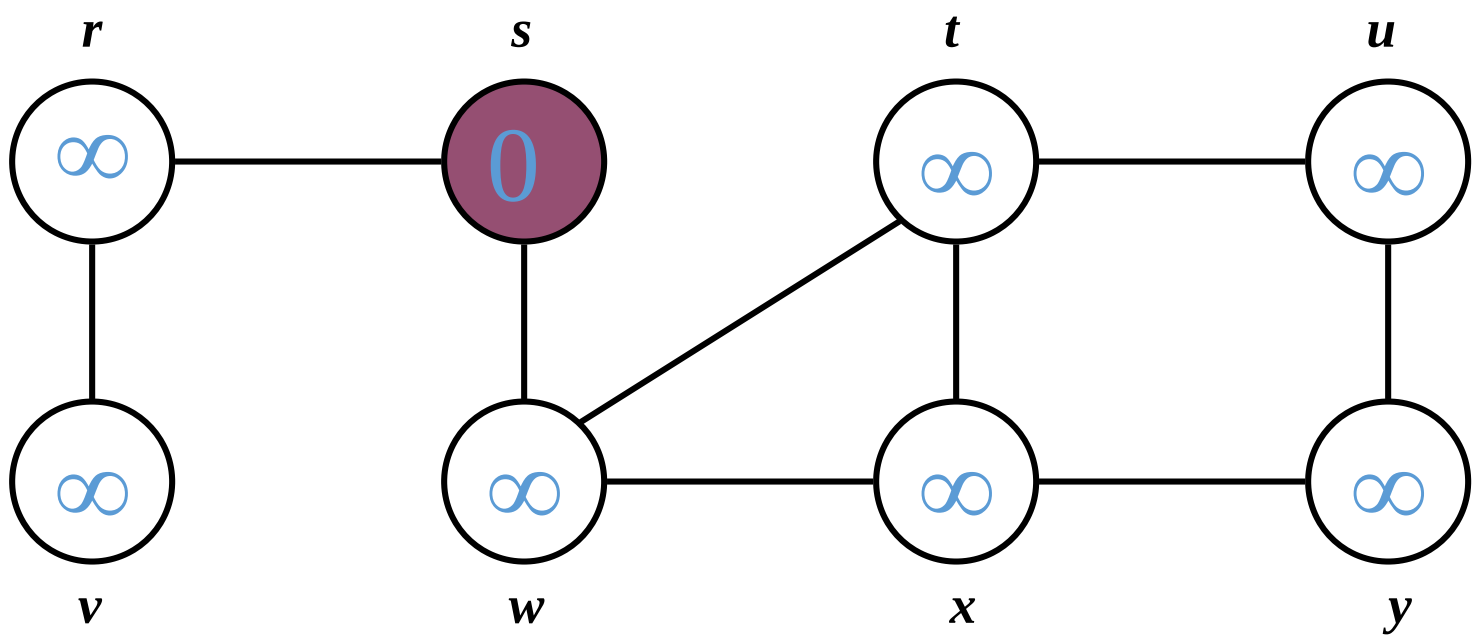


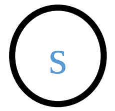
```
BFS($G, s$)
for each vertex $u \in V[G] - \{s\}$
    do $color[u] \leftarrow$ WHITE
       $d[u] \leftarrow \infty$
       $\pi[u] \leftarrow$ NIL
$color[s] \leftarrow$ GRAY
$d[s] \leftarrow 0$
$\pi[s] \leftarrow$ NIL
$Q \leftarrow \emptyset$
ENQUEUE($Q, s$)
while $Q \neq \emptyset$
    do $u \leftarrow$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

Breadth-First Search: Example



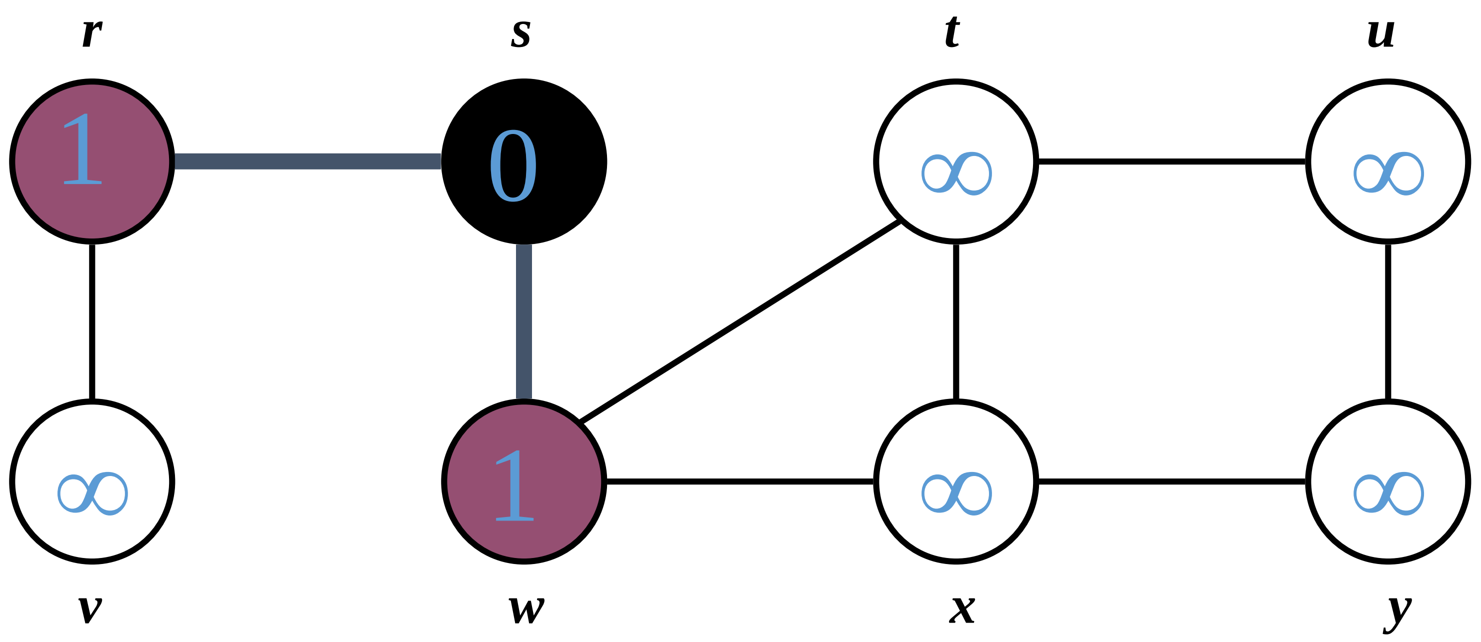
Breadth-First Search: Example

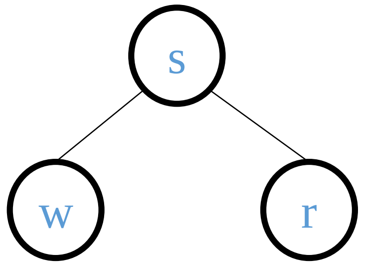




Q: s

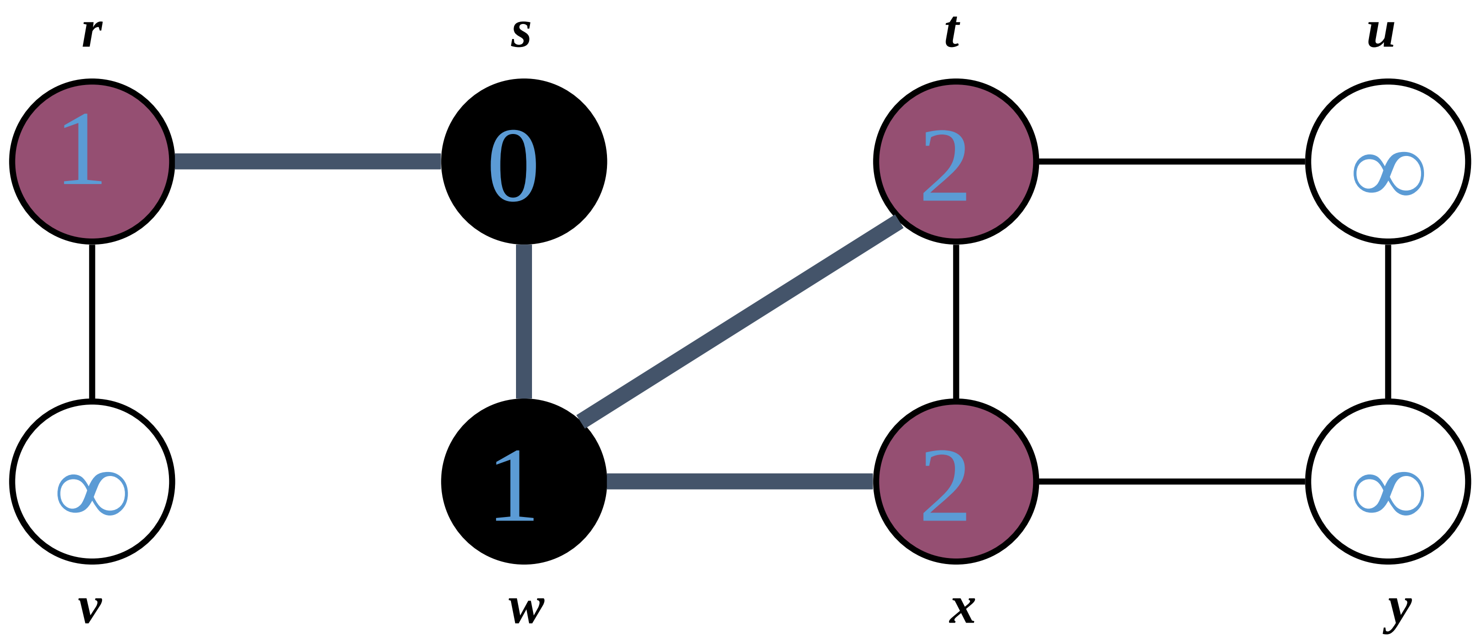
Breadth-First Search: Example

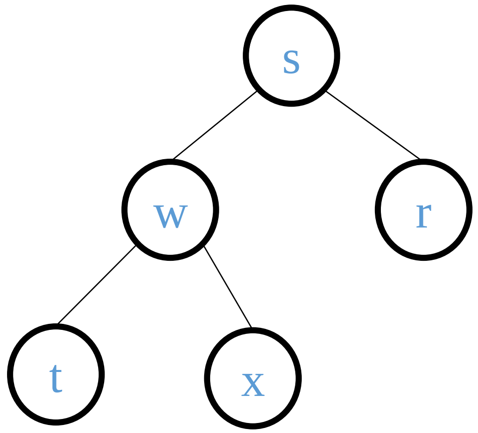




Q:	*w*	*r*

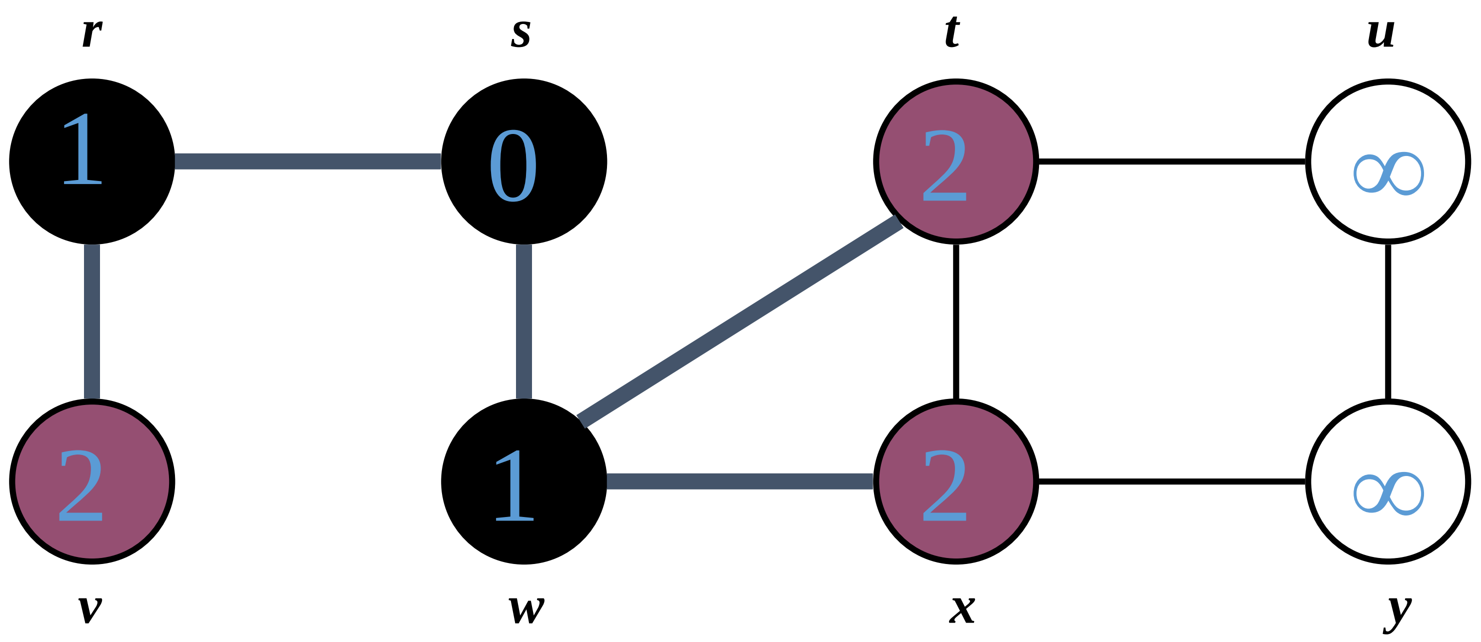
Breadth-First Search: Example

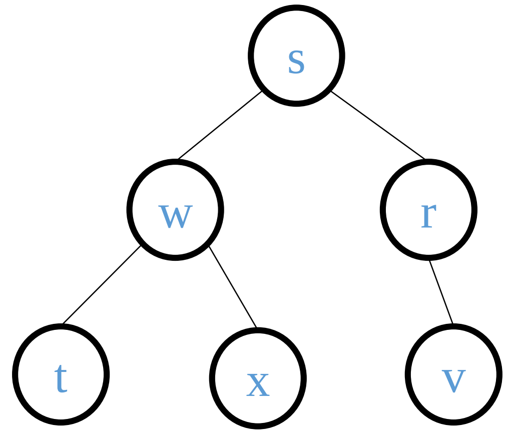




Q:	r	t	x

Breadth-First Search: Example

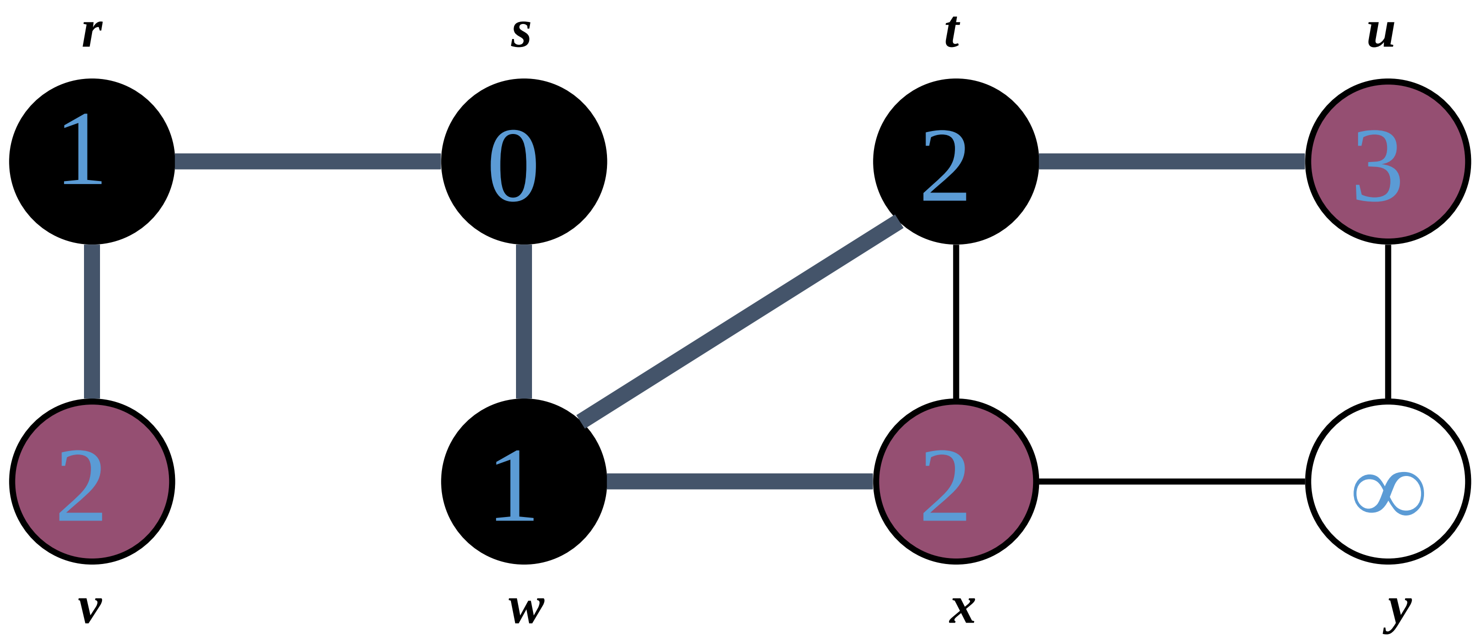


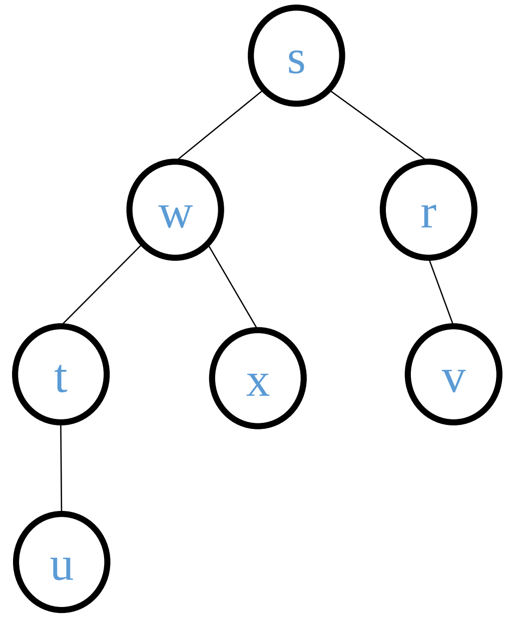


Q:

t	*x*	*v*

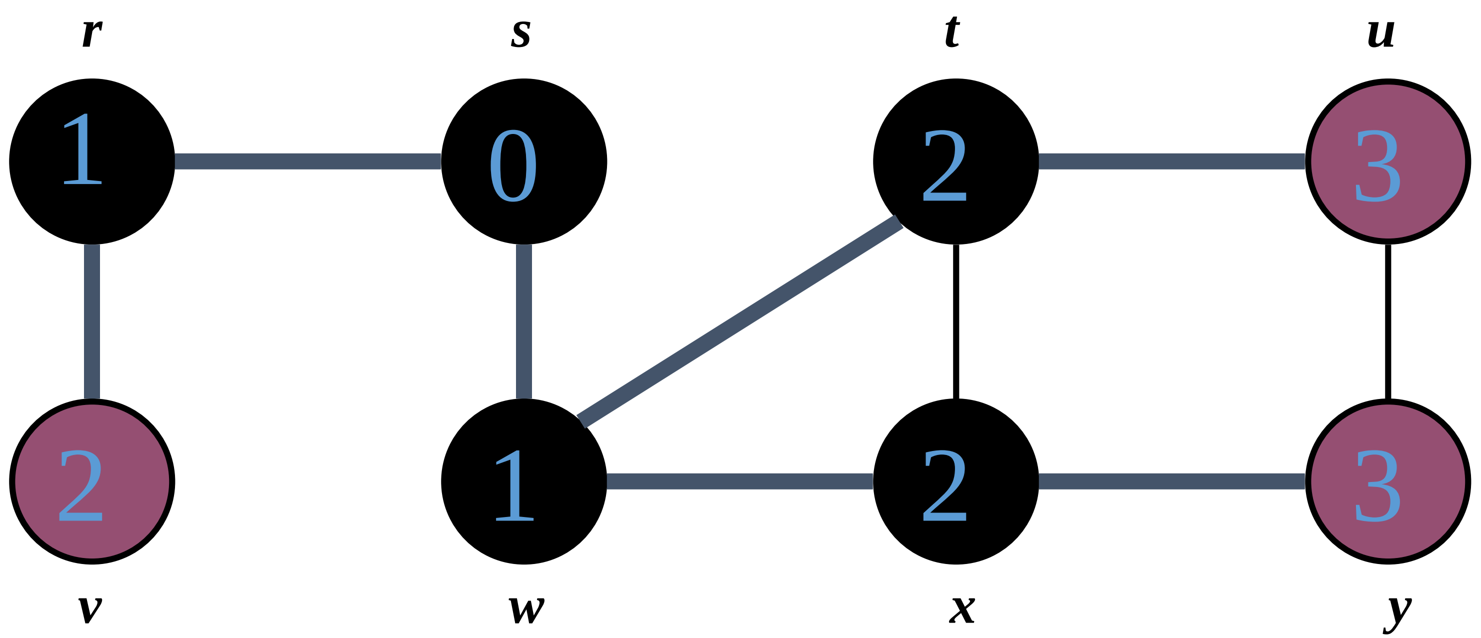
Breadth-First Search: Example

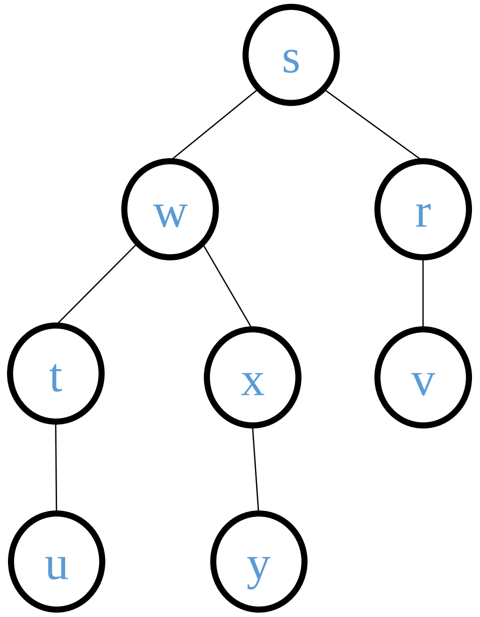




Q: | *x* | *v* | *u* |

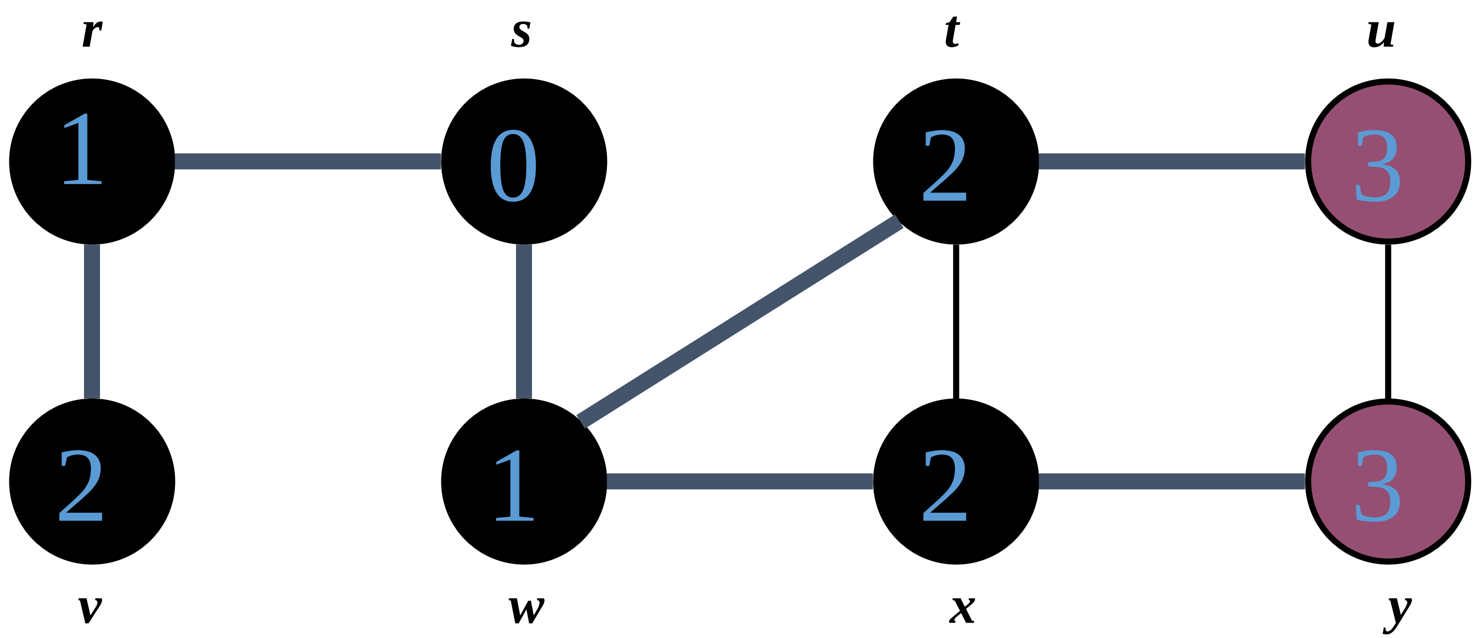
Breadth-First Search: Example

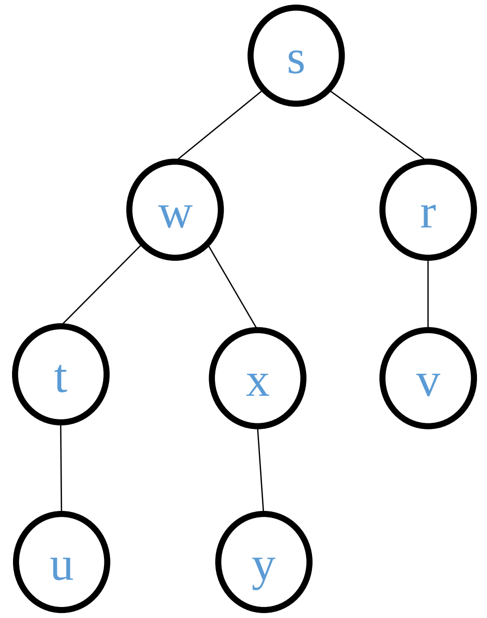




Q:	*v*	*u*	*y*

Breadth-First Search: Example

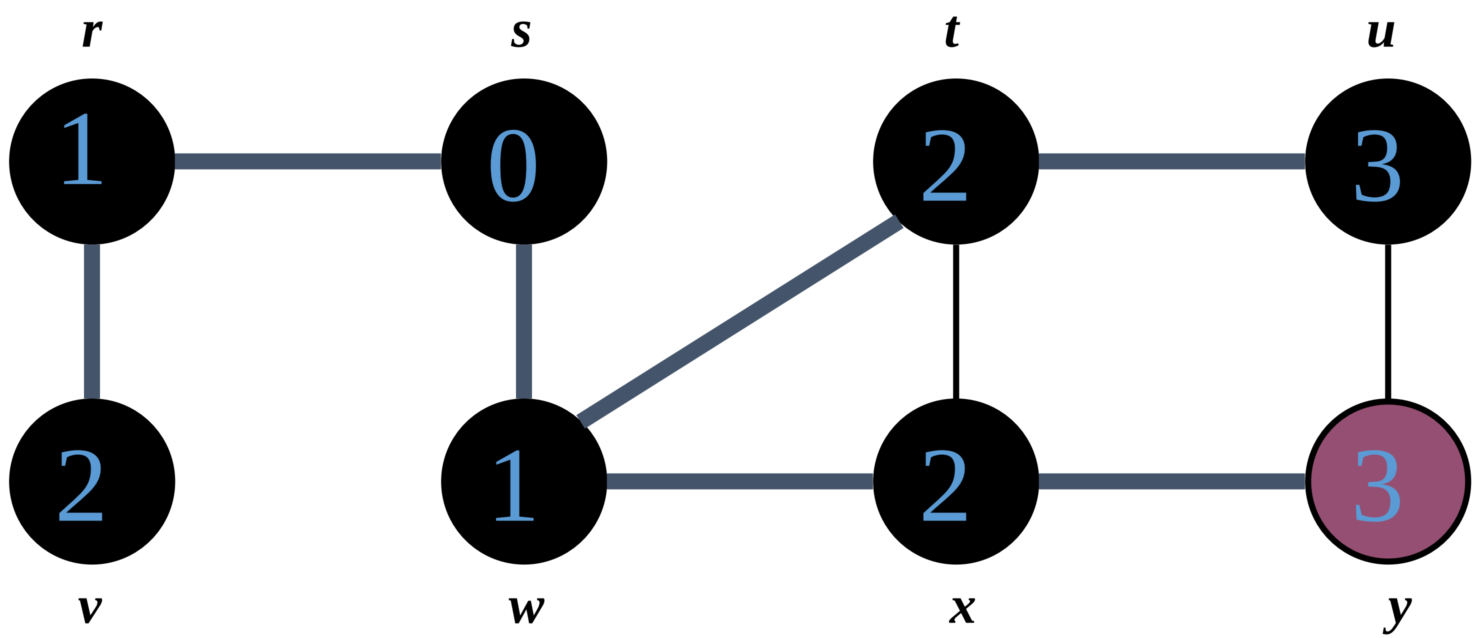


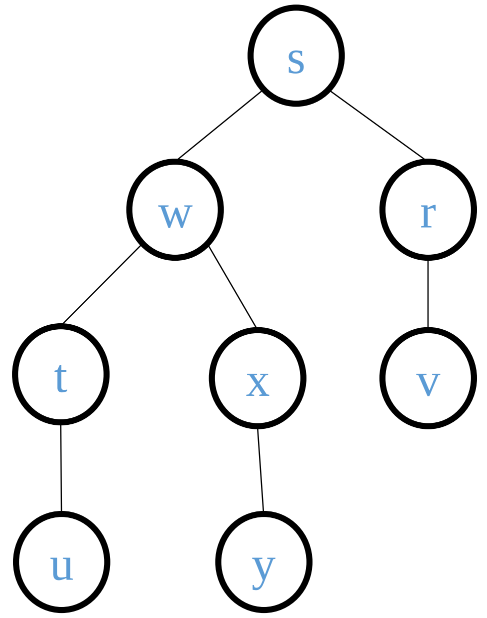


Q:

u	*y*

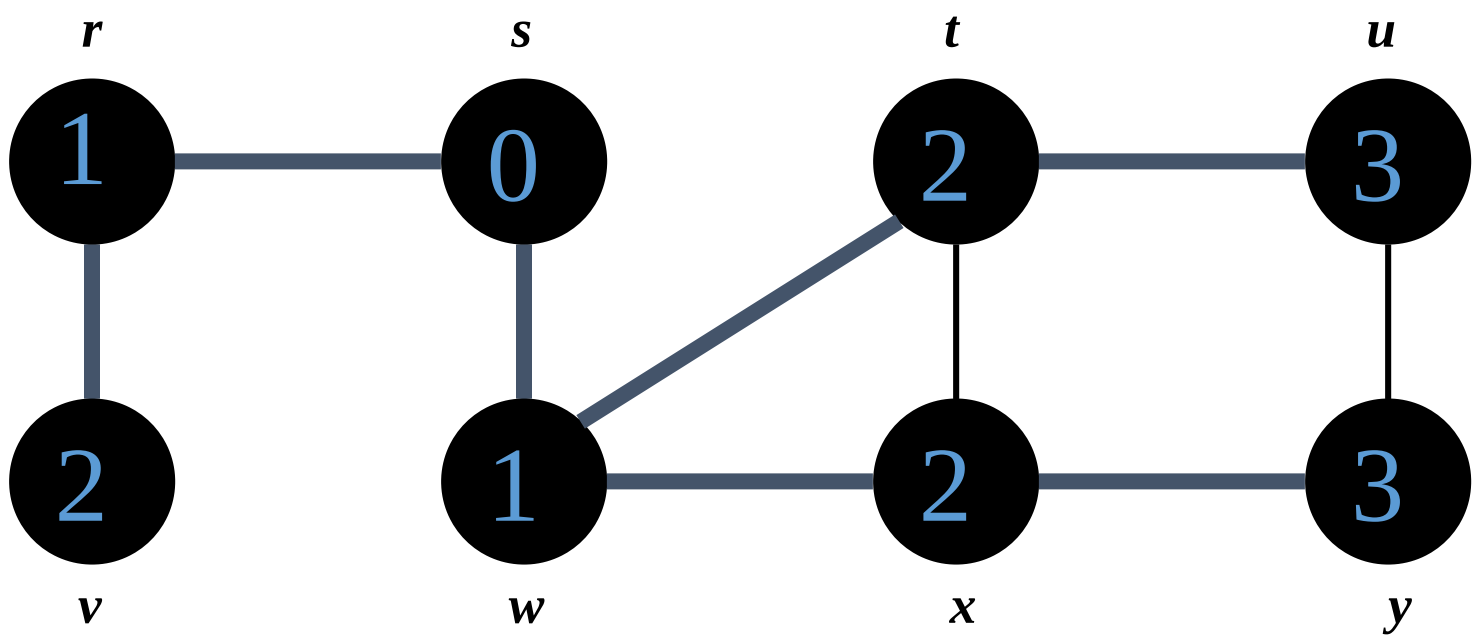
Breadth-First Search: Example

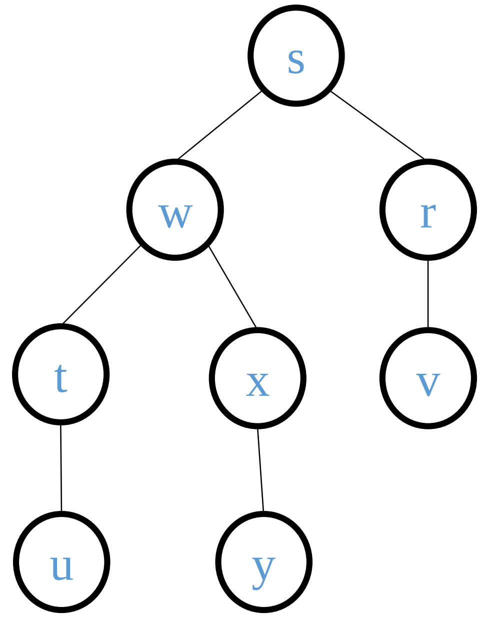




Q: $\boxed{y}$

Breadth-First Search: Example





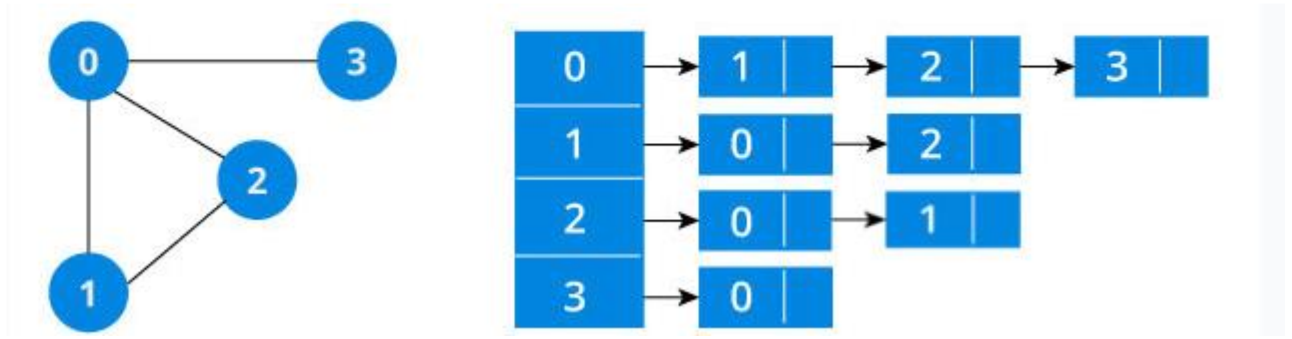
***Q*: Ø**

Breadth-First Search - Implementation

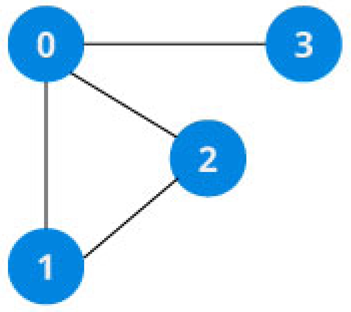
- Graph Input from FILE
- Adjacency List/ Matrix – DS: Array of linked list or 2D Array
- BFS   – DS: queue
- Output sequence of vertices for the BFS traversal

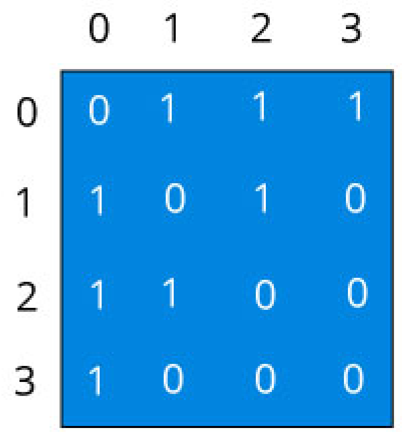
Breadth-First Search - Implementation

- Adjacency List (list)



- Adjacency Matrix (2D Array)





$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 2 & 1 & 1 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 \end{array}$$

Let’s Implementation