



DATA STRUCTURES

Graph traversal



Session Objectives

- To learn about graph traversals
 - Depth first traversal
 - Breadth first traversal

Session Outcomes

- At the end of this session, participants will be able to
 - Understand graph traversals

Agenda

- Graph traversals
 - DFS algorithm
 - BFS algorithm
 - Examples for DFS & BFS

Graph Traversal

Dr. B. Bharathi
SSNCE

August 27, 2019

Graph traversals

1. Depth first search
2. Breadth first search

Depth first search

Depth-first search (DFS) is an algorithm for traversing or searching tree or graph data structures. One starts at the root or selecting some arbitrary node as the root in the case of a graph and explores as far as possible along each branch before backtracking.

DFS follows the following rules:

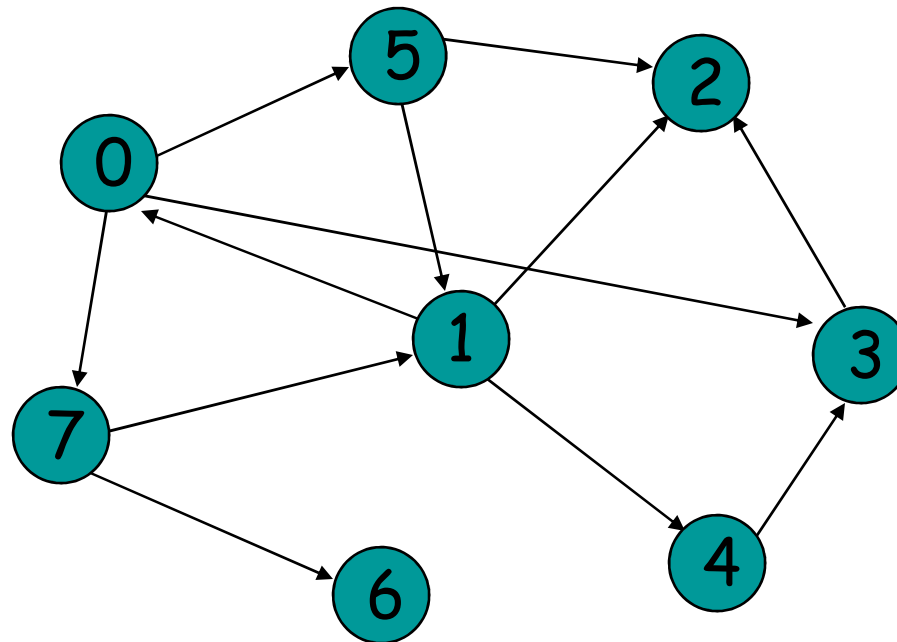
1. Select an unvisited node x , visit it, and treat as the current node
2. Find an unvisited neighbor of the current node, visit it, and make it the new current node;
3. If the current node has no unvisited neighbors, backtrack to its parent, and make that parent the new current node;
4. Repeat steps 2 and 3 until no more nodes can be visited.
5. If there are still unvisited nodes, repeat from step 1.

DFS(Pseudo code)

```
// non-recursive, depth-first search
void dfs (Node v) {
    if (v == null)
        return;
    push(v);
    while (stack is not empty) {
        pop(v);
        if (v has not yet been visited)
            mark&visit(v);

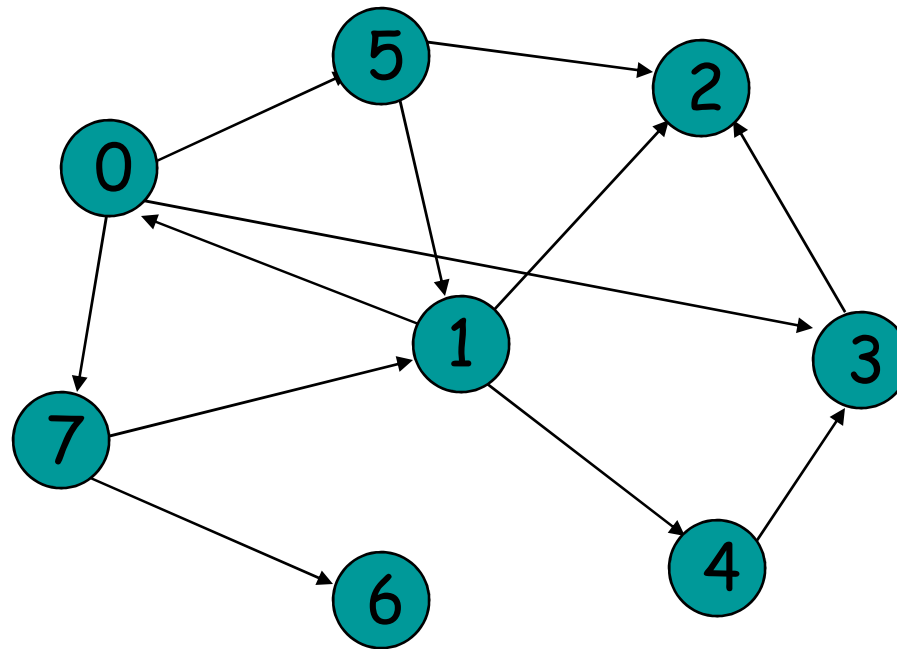
        for (each w adjacent to v)
            if (w has not yet been visited)
                push(w);
    } // while
} // dfs
```


Example



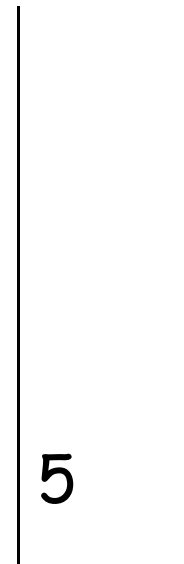
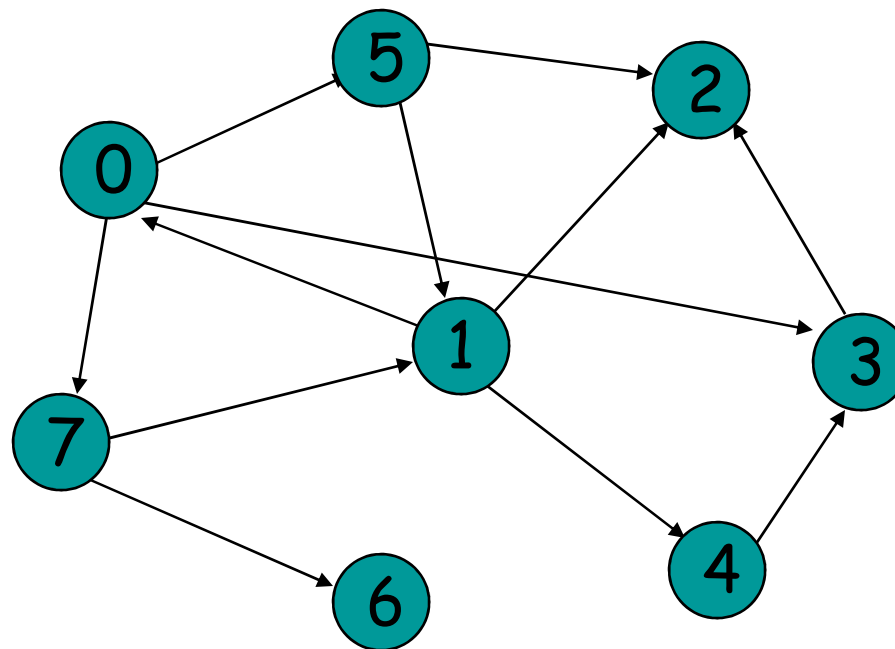
Policy: Visit adjacent nodes in increasing index order

DFS: Start with Node 5



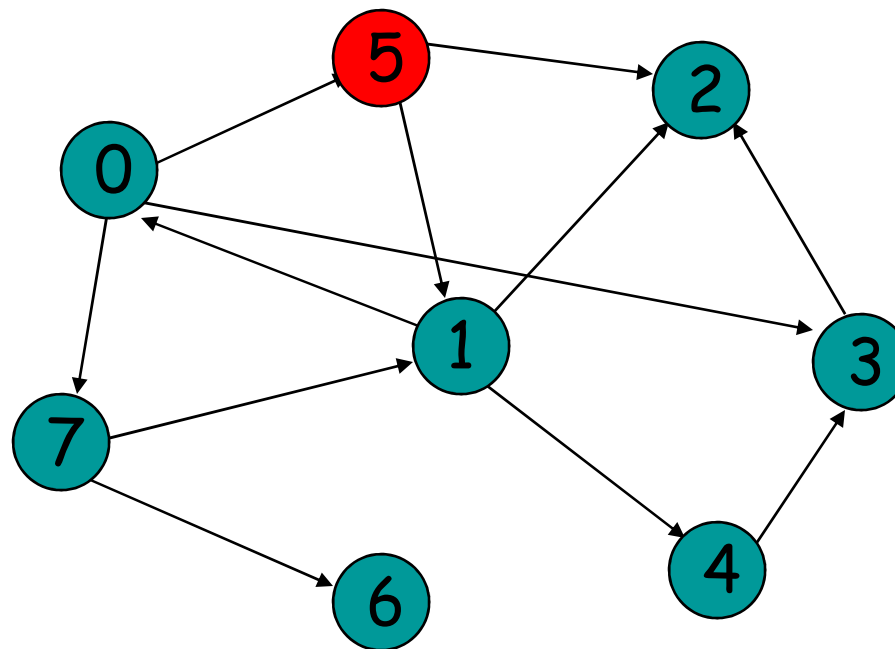
5 1 0 3 2 7 6 4

DFS: Start with Node 5



Push 5

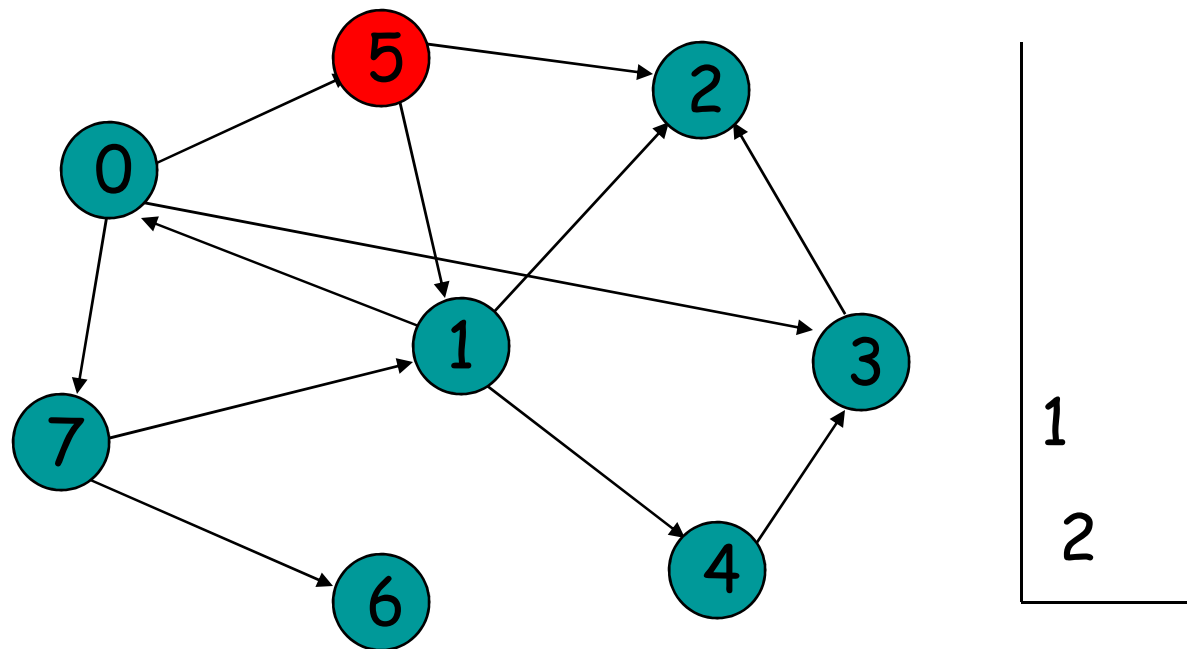
DFS: Start with Node 5



Pop/Visit/Mark 5

5

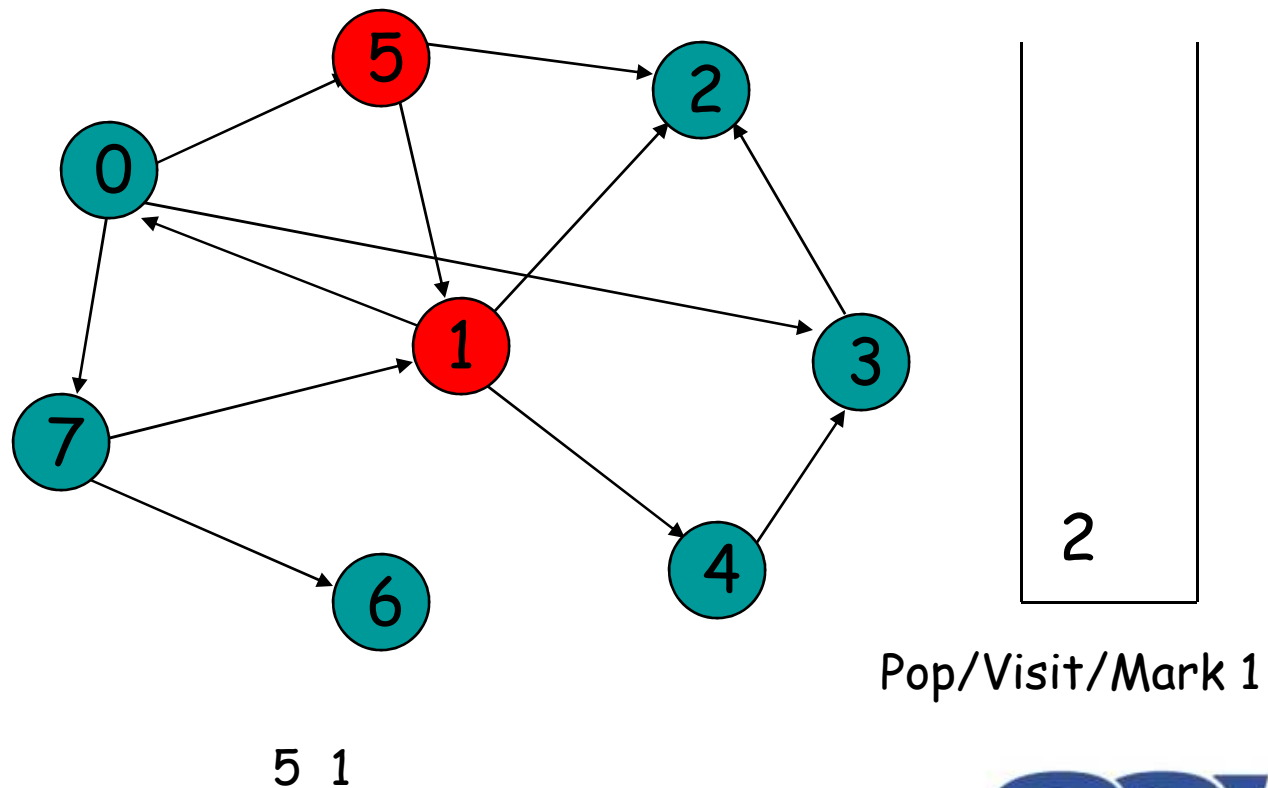
DFS: Start with Node 5



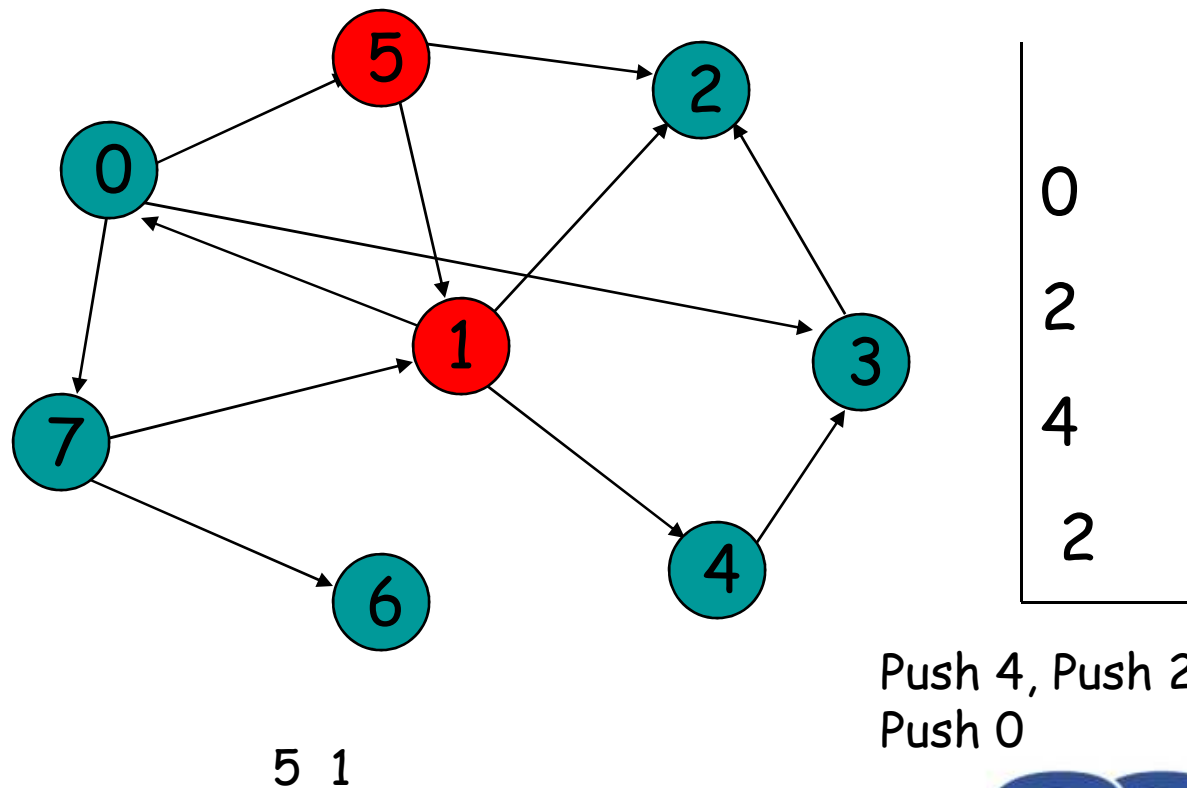
Push 2, Push 1

5

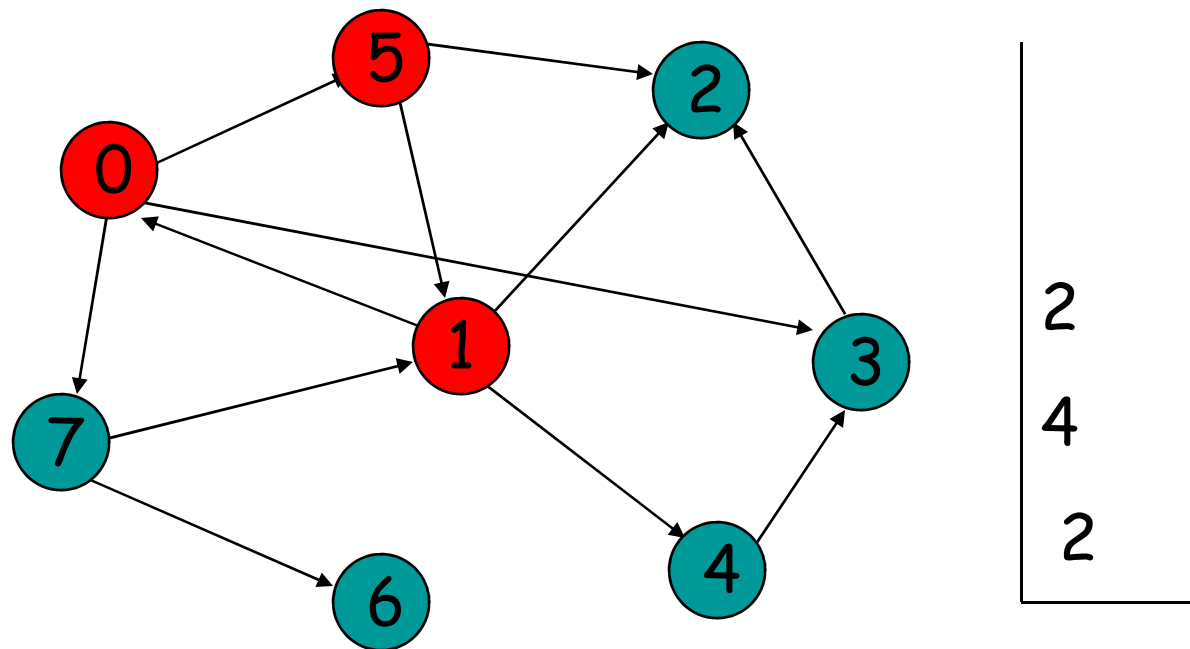
DFS: Start with Node 5



DFS: Start with Node 5



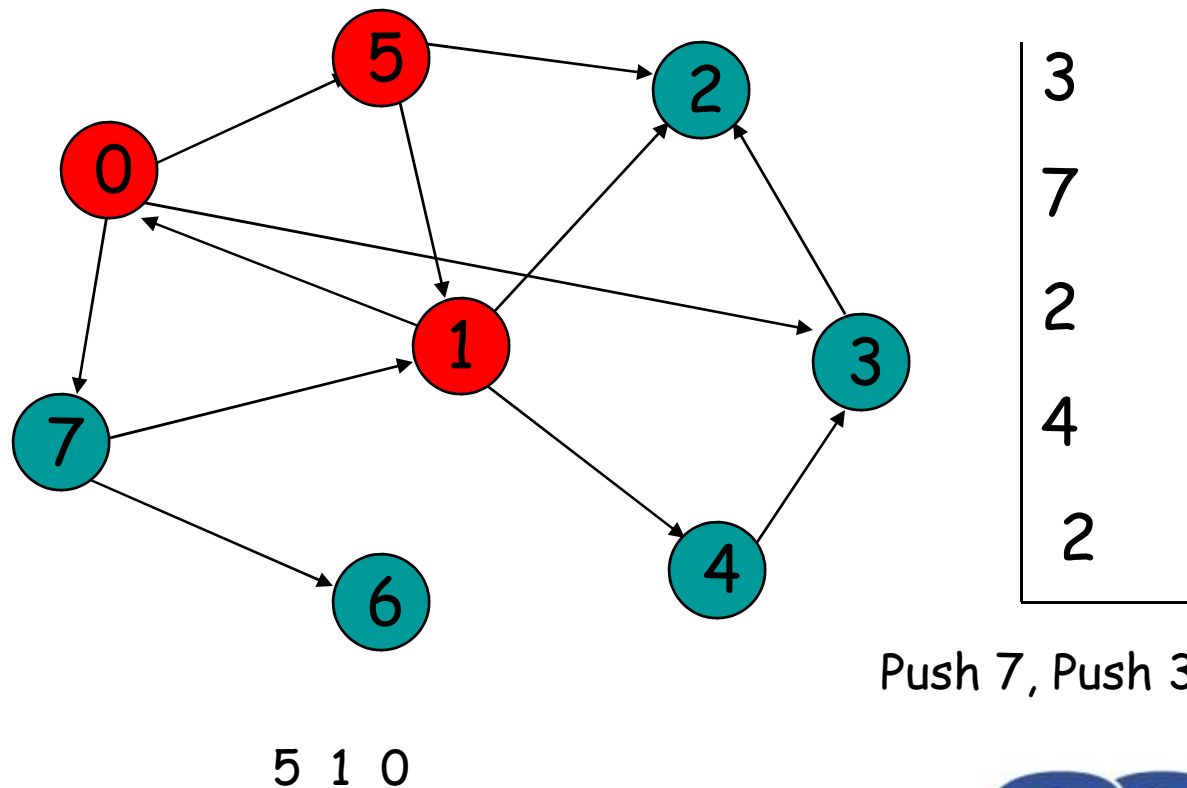
DFS: Start with Node 5



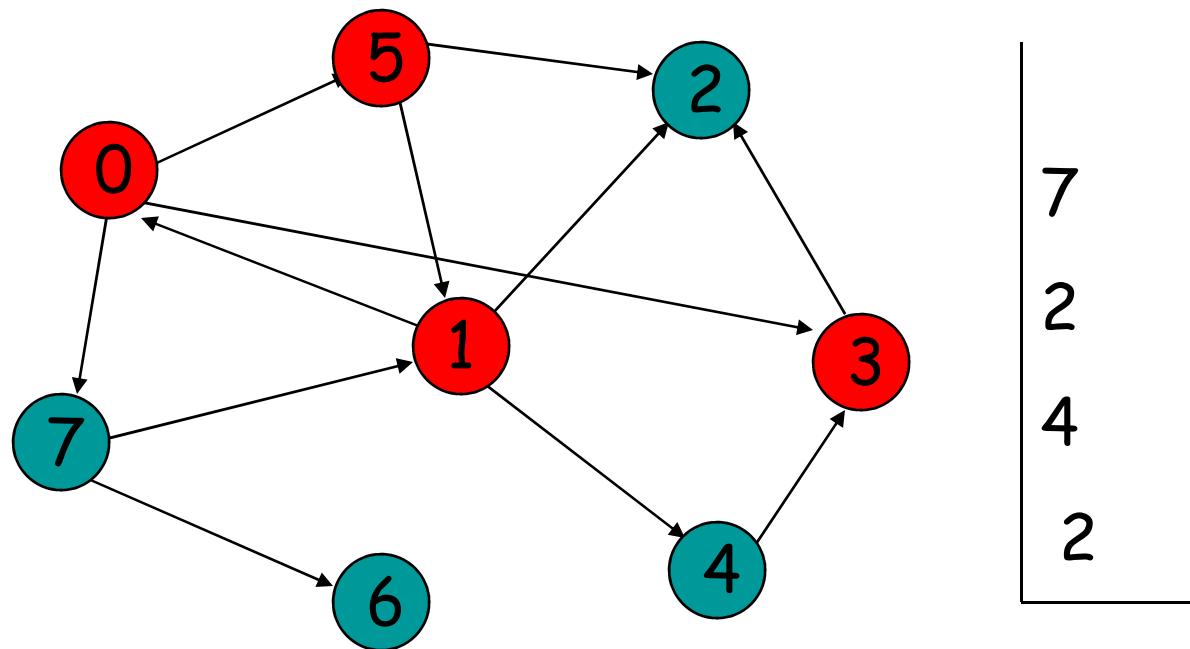
Pop/Visit/Mark 0

5 1 0

DFS: Start with Node 5



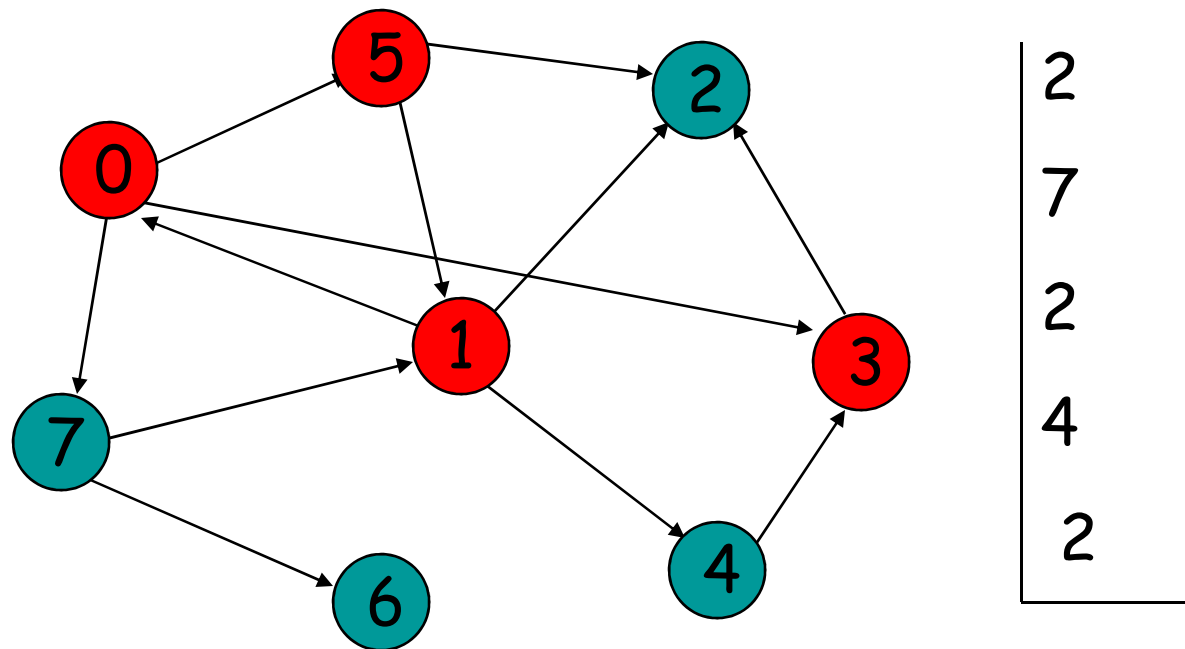
DFS: Start with Node 5



Pop/Visit/Mark 3

5 1 0 3

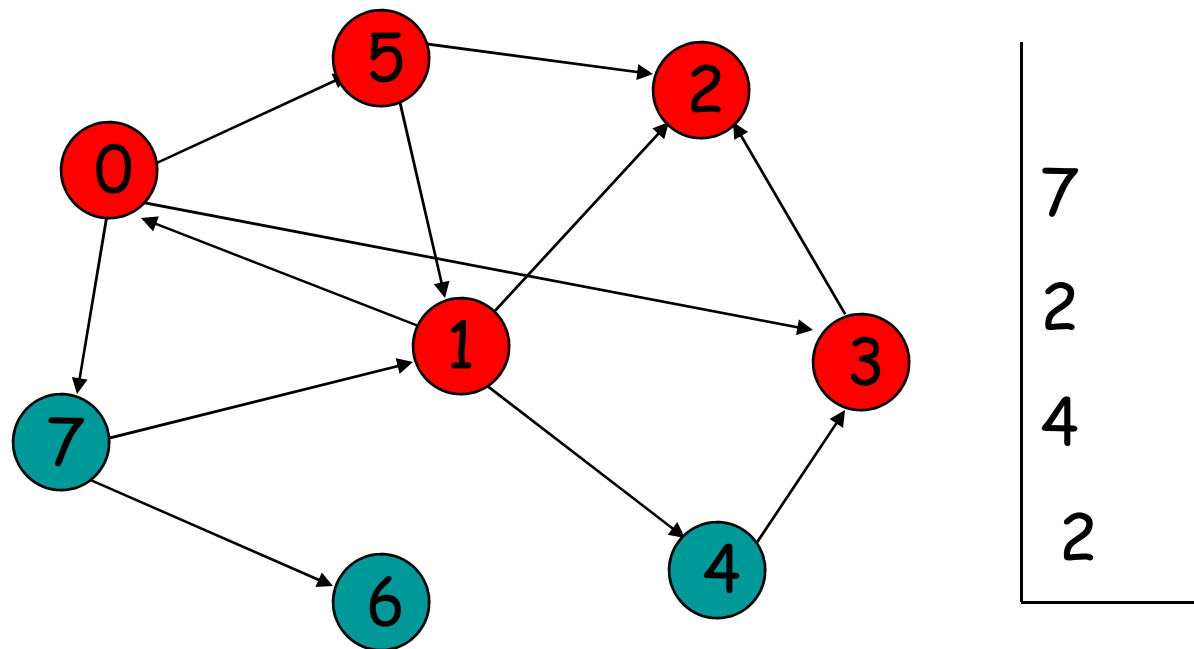
DFS: Start with Node 5



Push 2

5 1 0 3

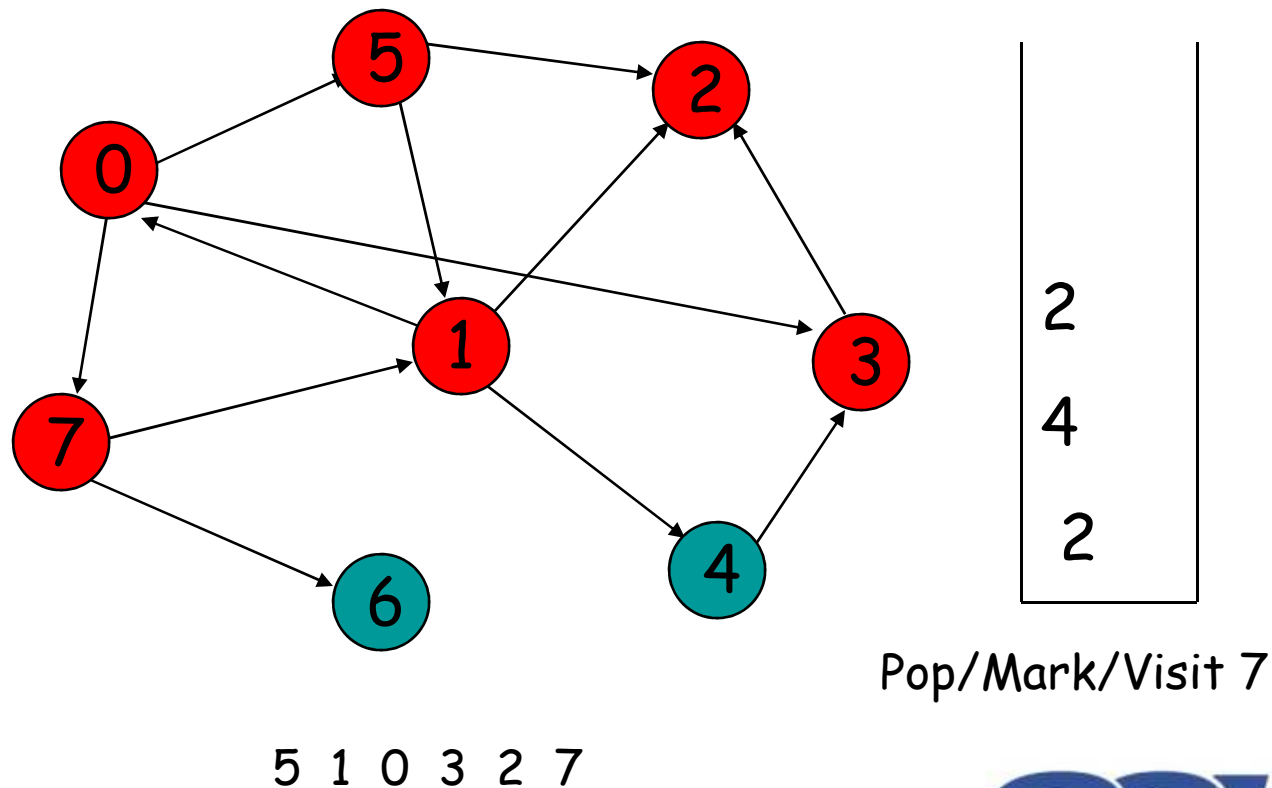
DFS: Start with Node 5



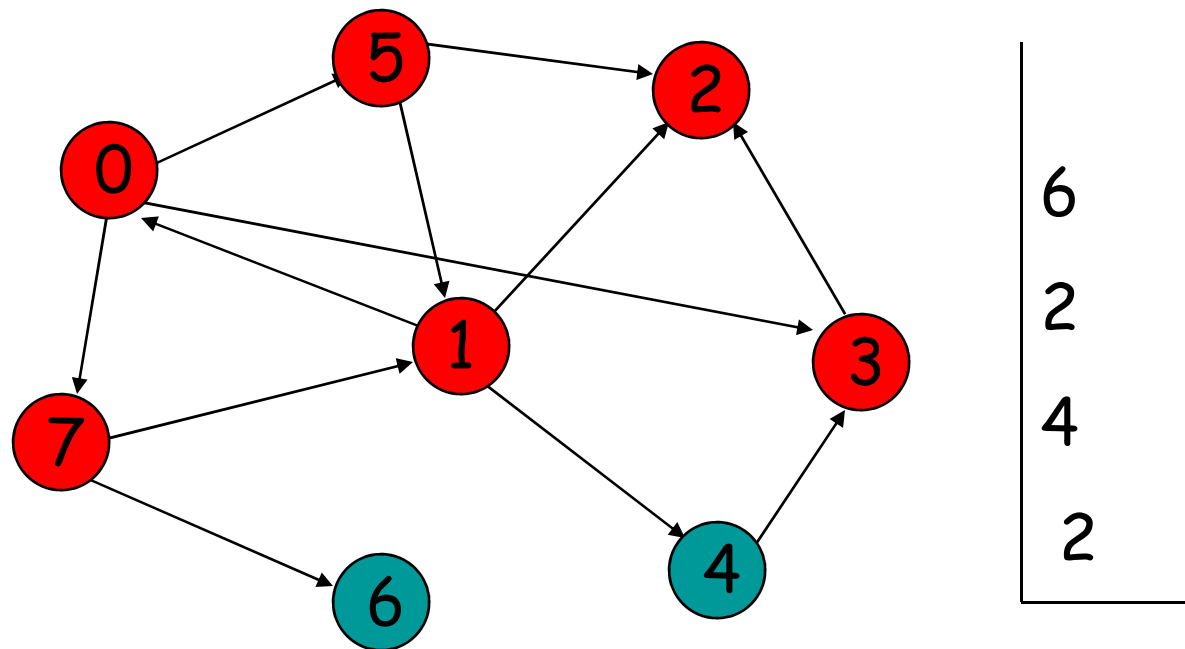
Pop/Mark/Visit 2

5 1 0 3 2

DFS: Start with Node 5



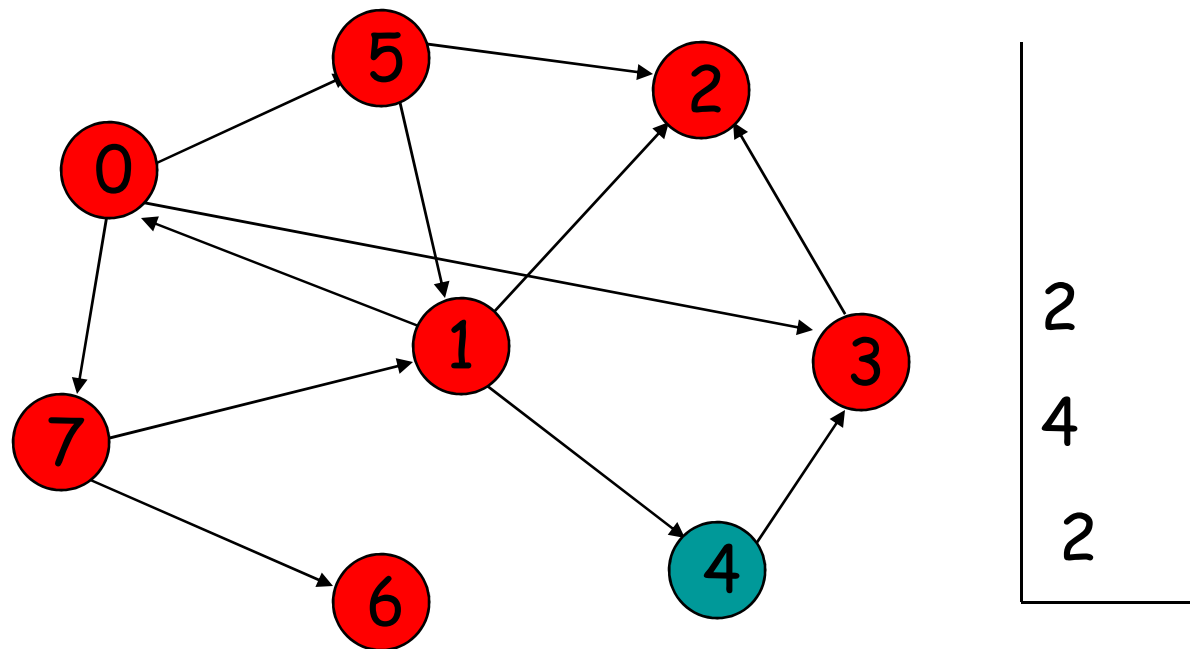
DFS: Start with Node 5



Push 6

5 1 0 3 2 7

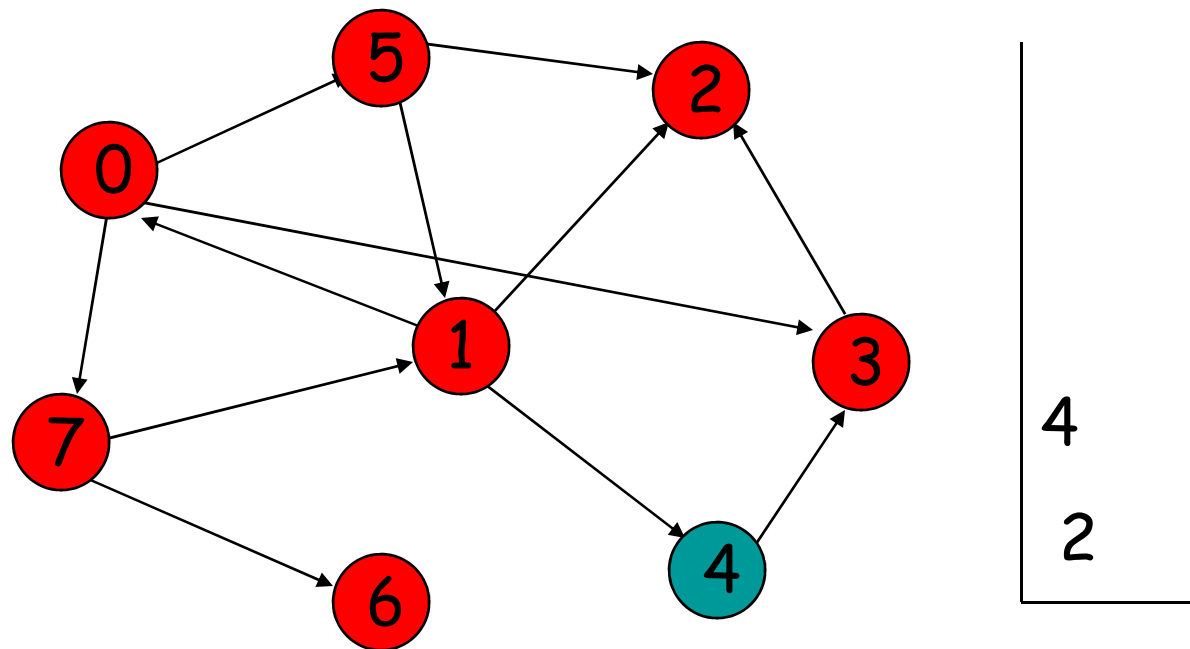
DFS: Start with Node 5



Pop/Mark/Visit 6

5 1 0 3 2 7 6

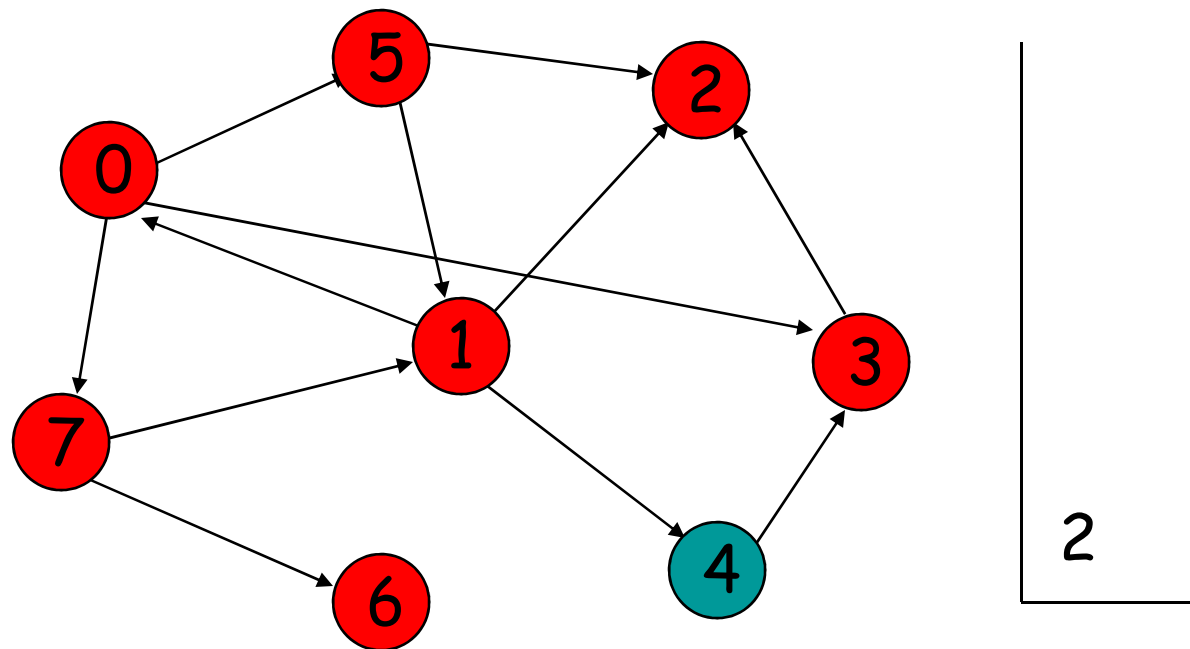
DFS: Start with Node 5



Pop (don't visit) 2

5 1 0 3 2 7 6

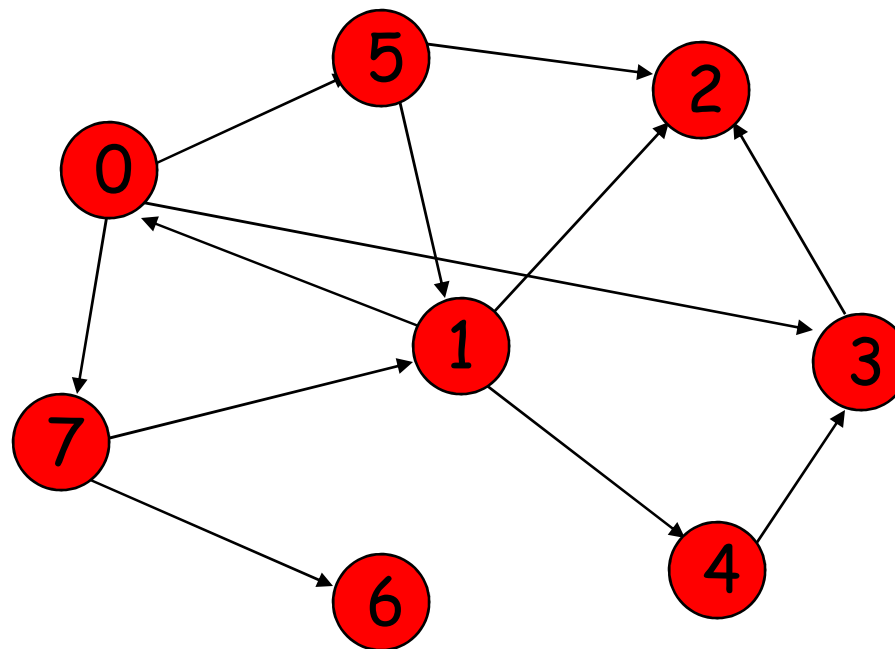
DFS: Start with Node 5



Pop/Mark/Visit 4

5 1 0 3 2 7 6 4

DFS: Start with Node 5



Pop (don't visit) 2

5 1 0 3 2 7 6 4

Breadth first traversal

The **Breadth-first search (BFS)** begins at a node and inspects all the neighbouring nodes. Then for each of those neighbour nodes in turn, it inspects their neighbour nodes which were unvisited, and so on.

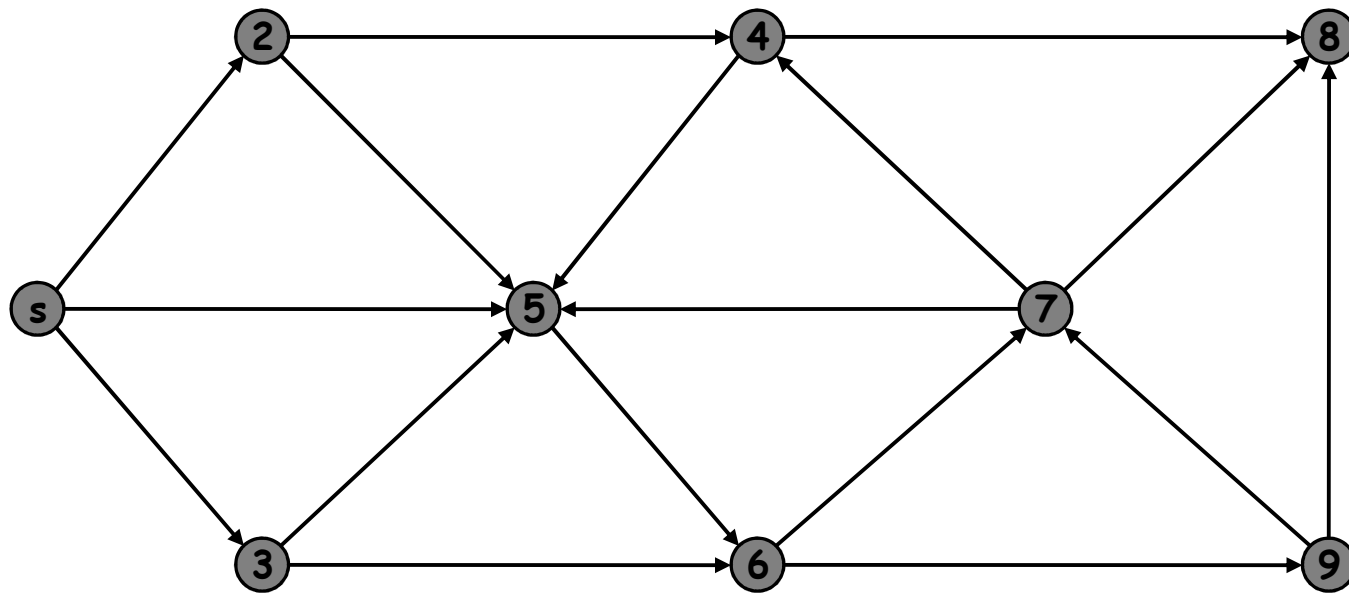
BFS follows the following rules:

1. Select an unvisited node x , visit it, have it be the root in a BFS tree being formed. Its level is called the current level.
2. From each node z in the current level, in the order in which the level nodes were visited, visit all the unvisited neighbors of z . The newly visited nodes from this level form a new level that becomes the next current level.
3. Repeat step 2 until no more nodes can be visited.
4. If there are still unvisited nodes, repeat from Step 1.

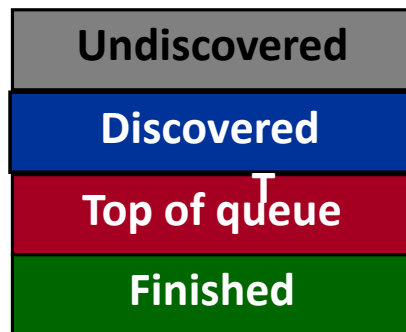
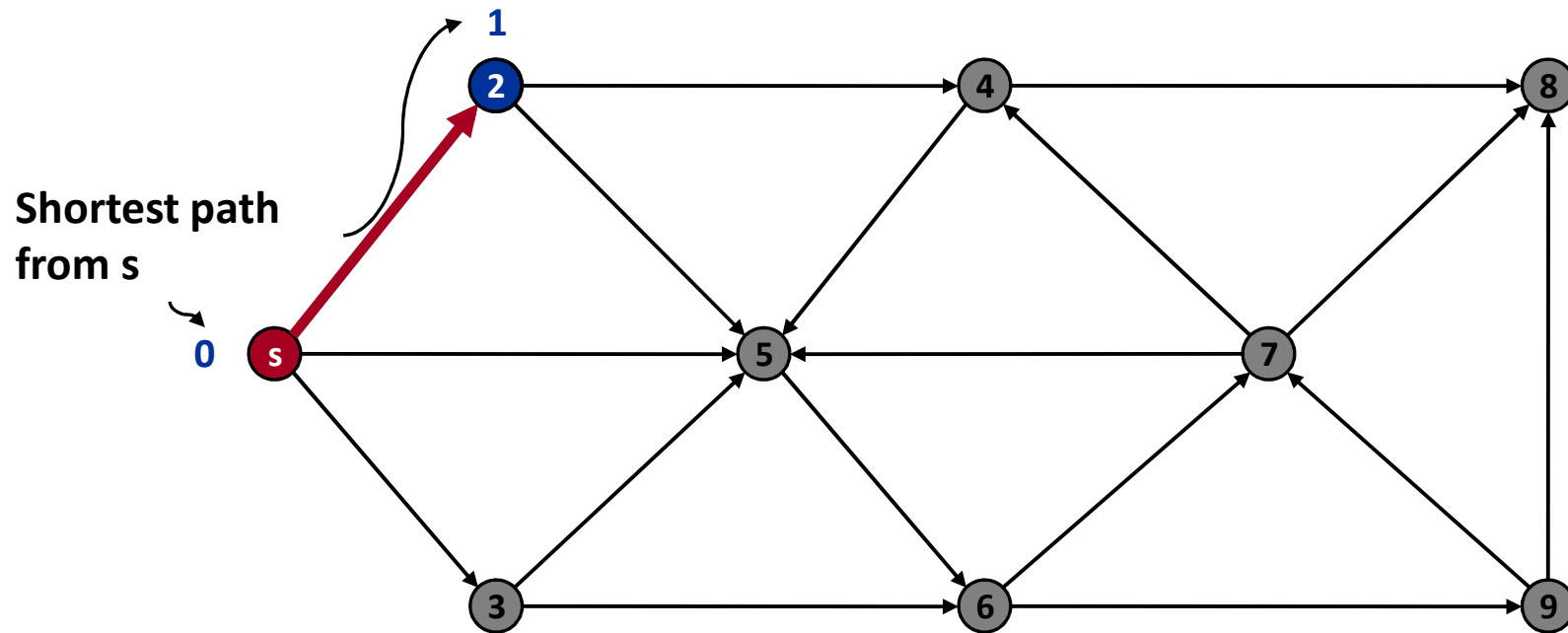
BFS(Pseudo code)

```
BFS(input: graph G)
{
  Queue Q;  Integer x, z, y;
  while (G has an unvisited node x)
  {
    visit(x);
    Enqueue(x,Q);
    while (Q is not empty)
    {
      z := Dequeue(Q);
      for all (unvisited neighbor y of z)
      {
        visit(y);
        Enqueue(y,Q);
      }
    }
  }
}
```

Breadth First Search

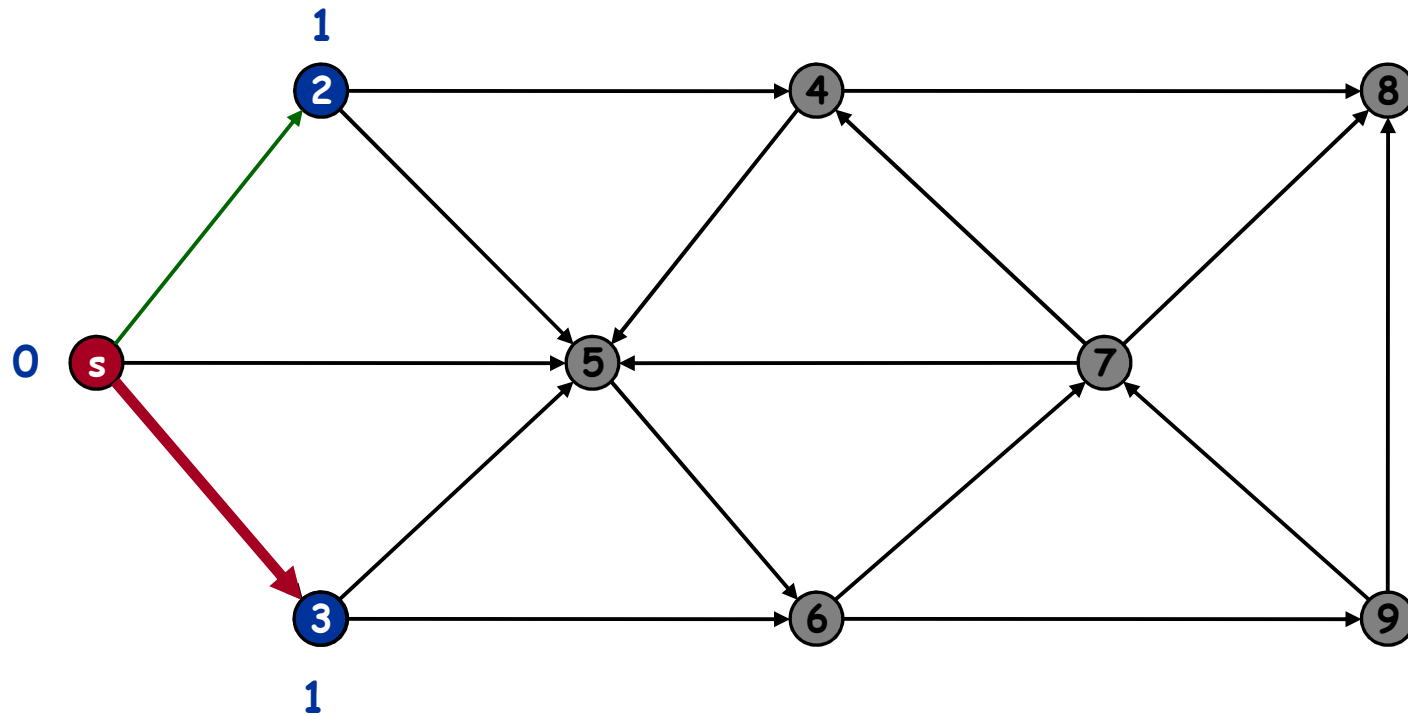


Breadth First Search



Queue: s

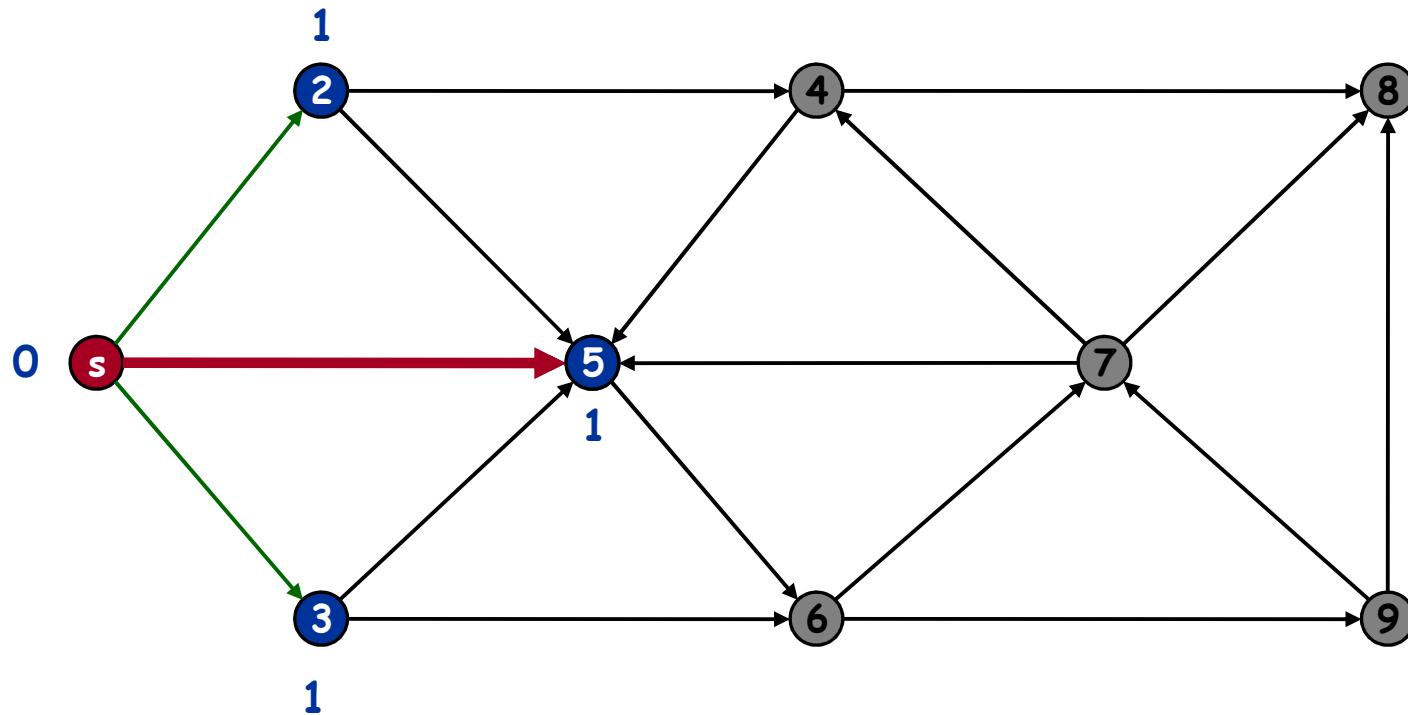
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: s 2

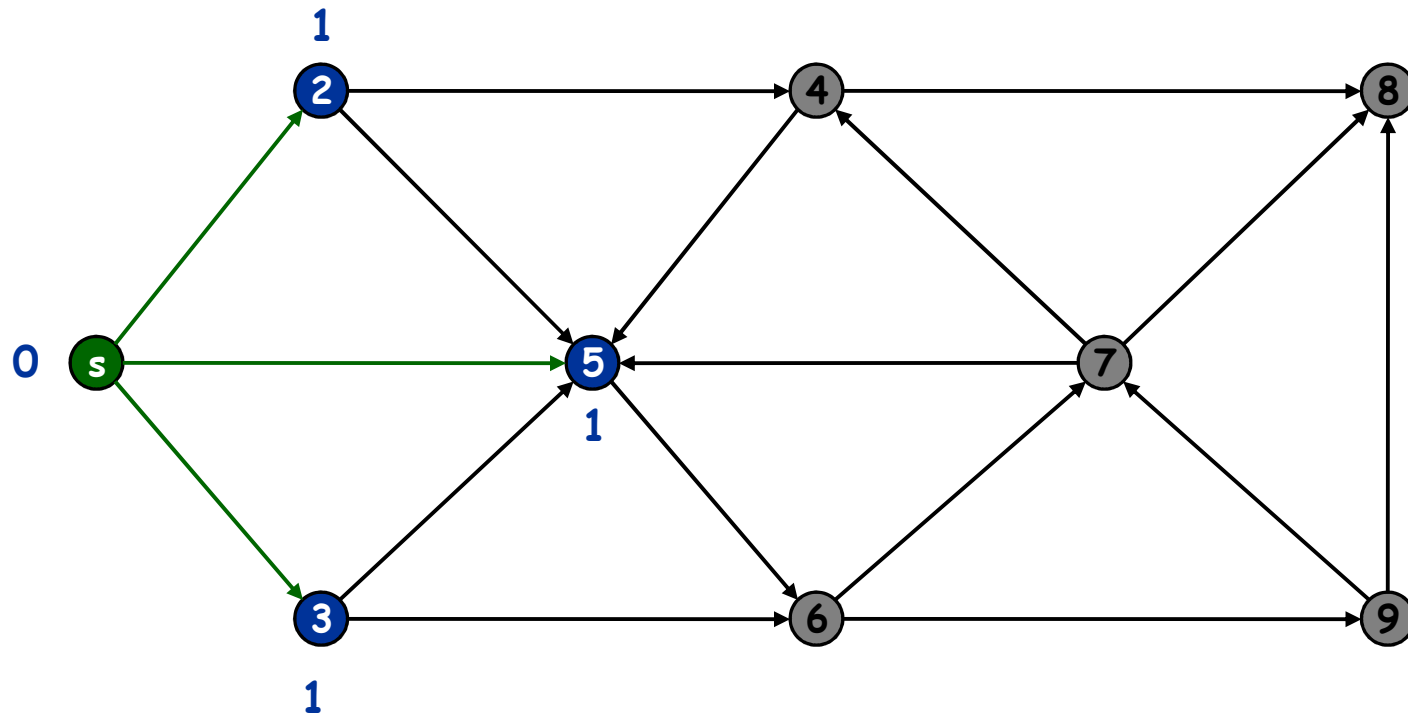
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: s 2 3

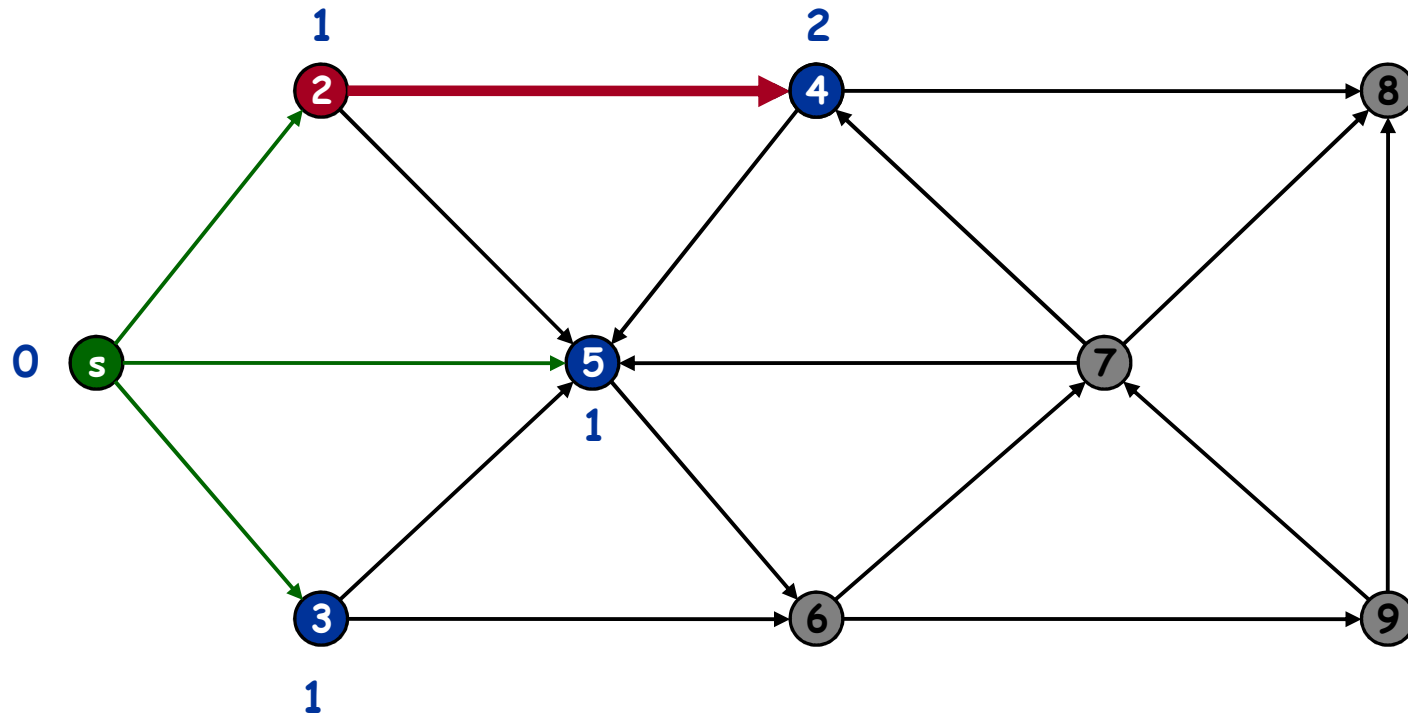
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 2 3 5

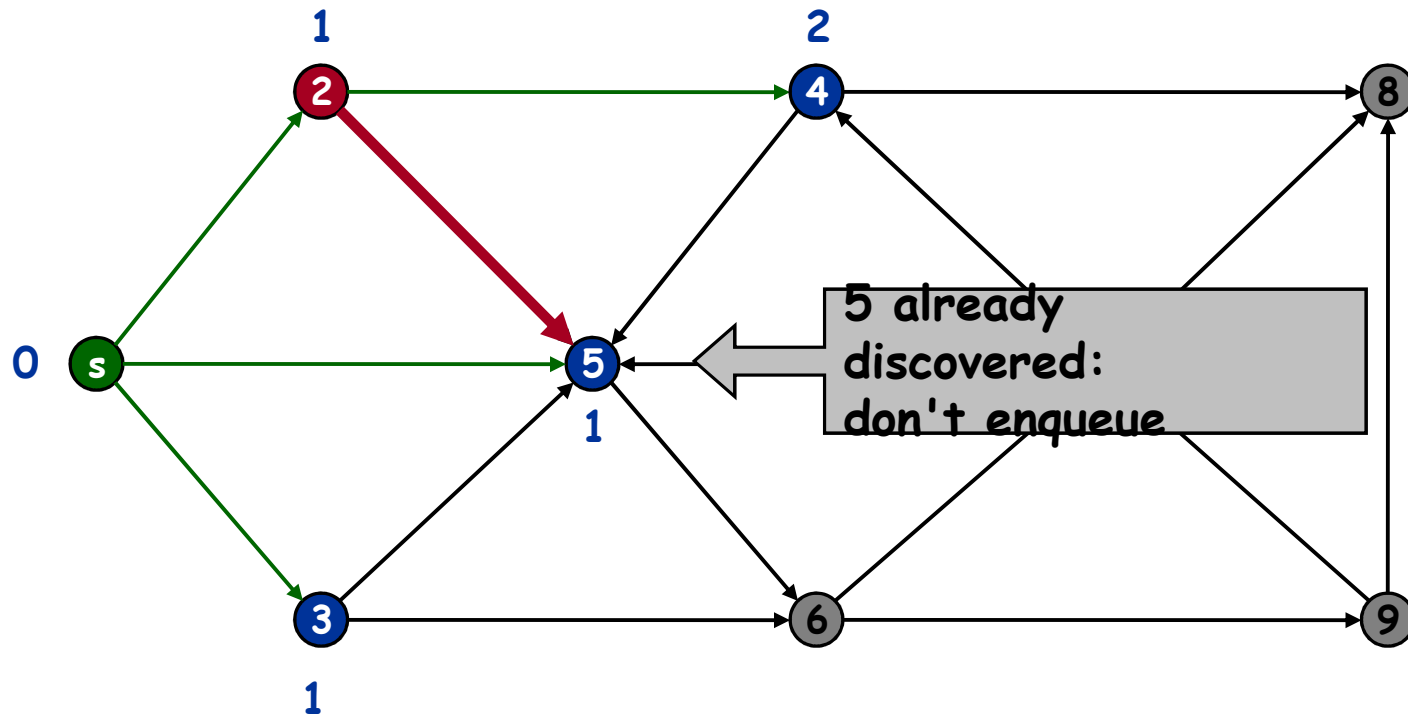
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 2 3 5

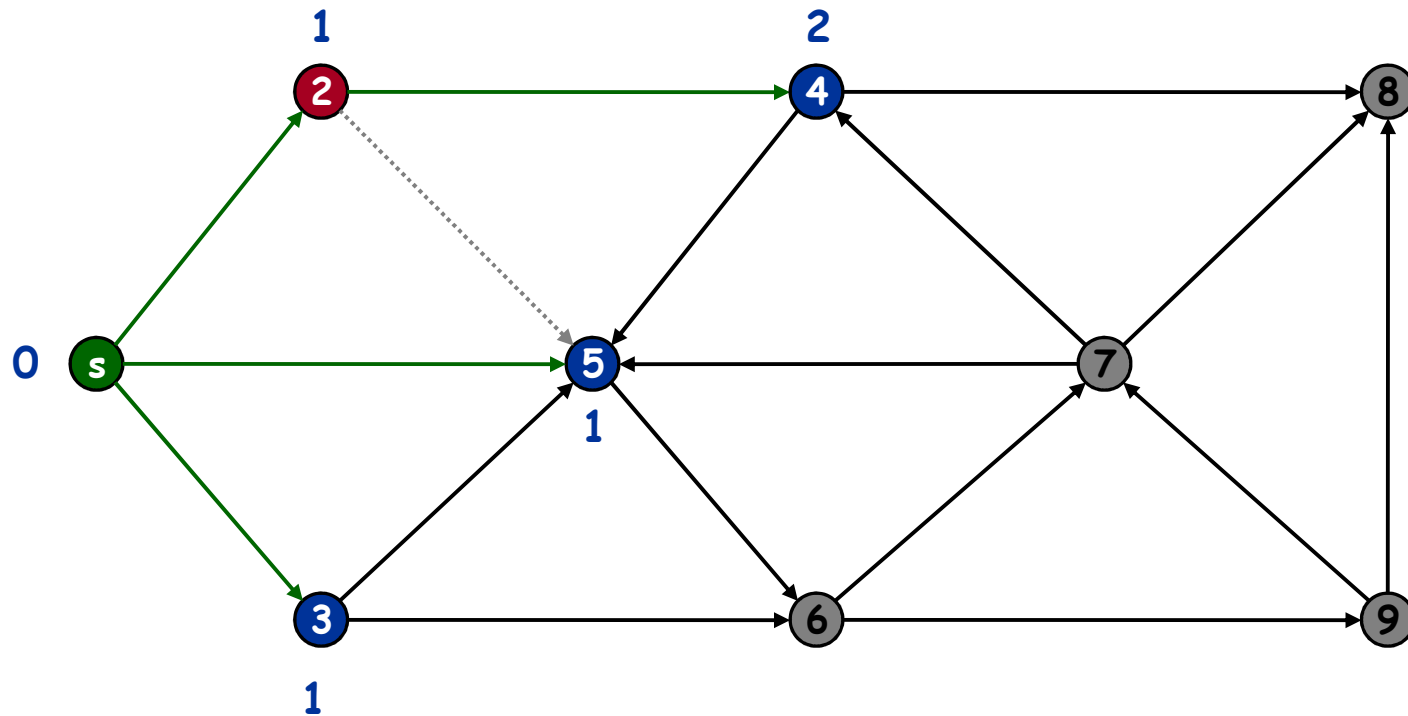
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 2 3 5 4

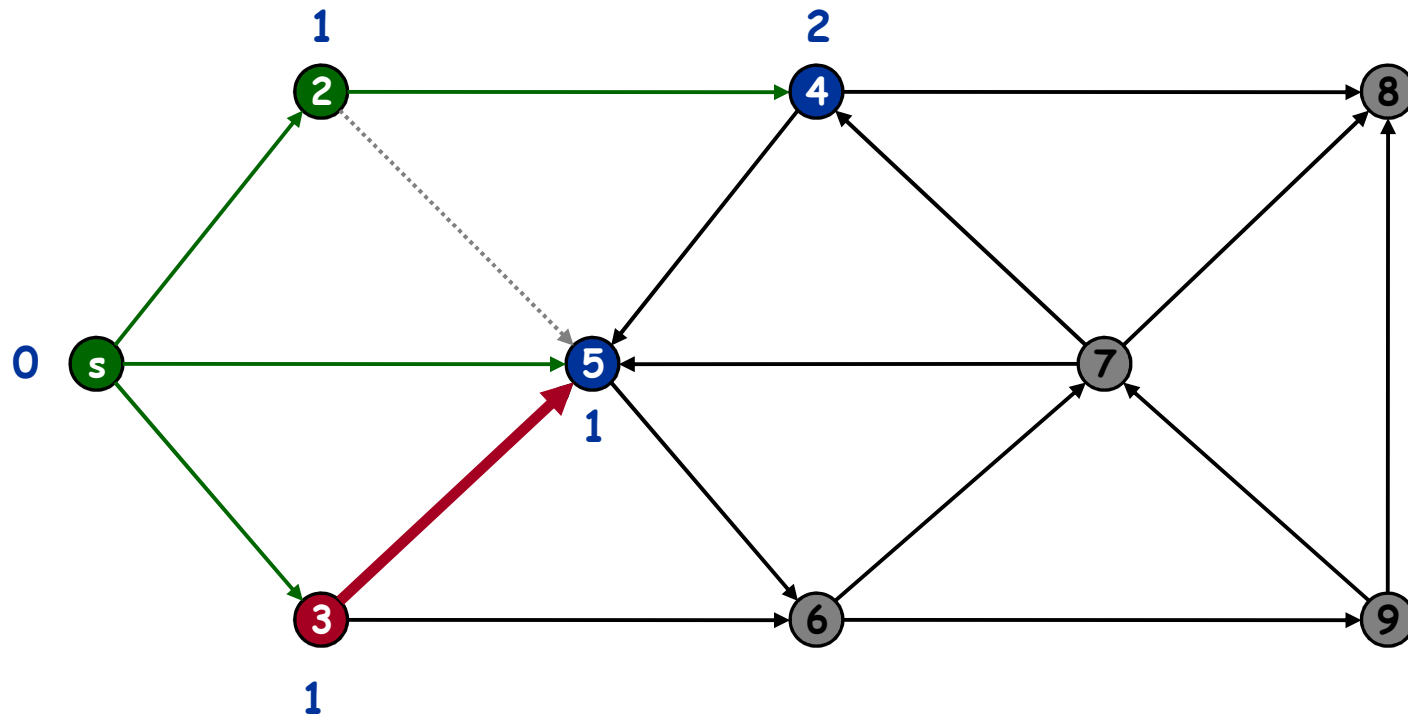
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 2 3 5 4

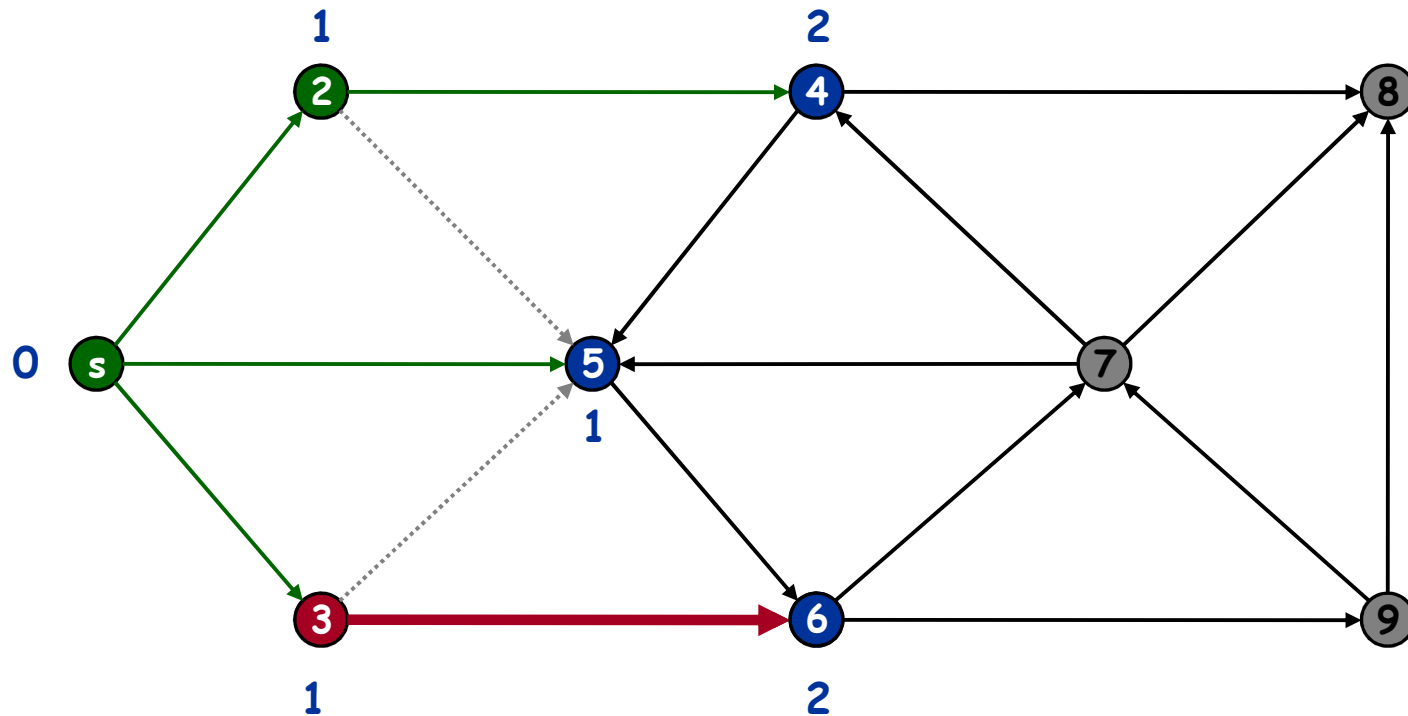
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 3 5 4

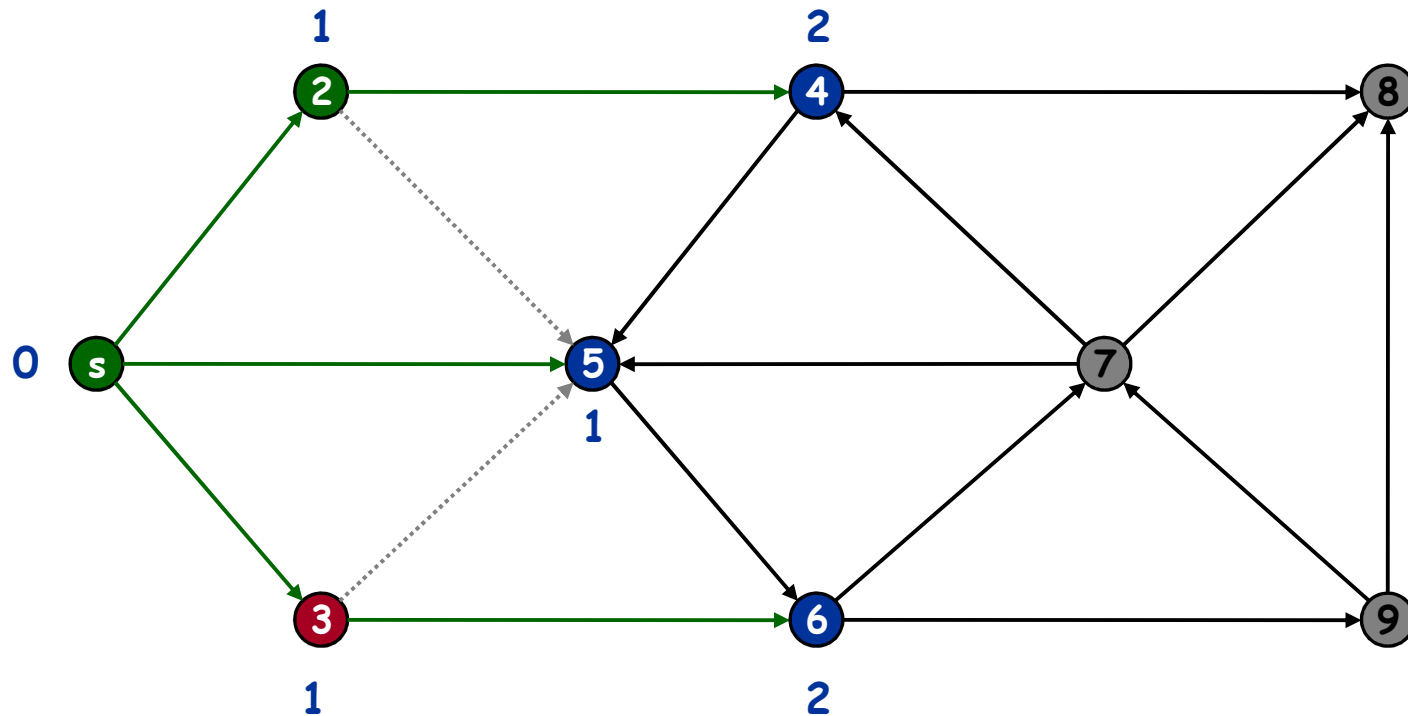
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 3 5 4

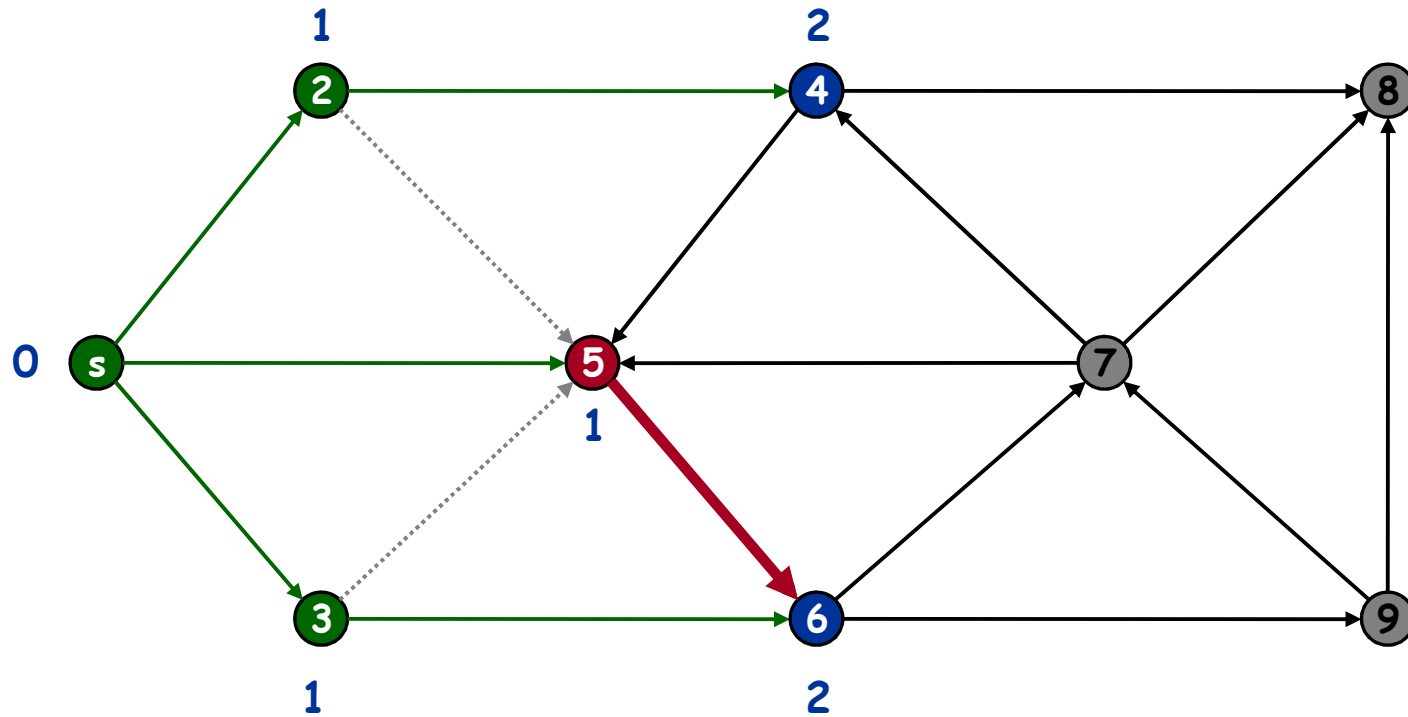
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 3 5 4 6

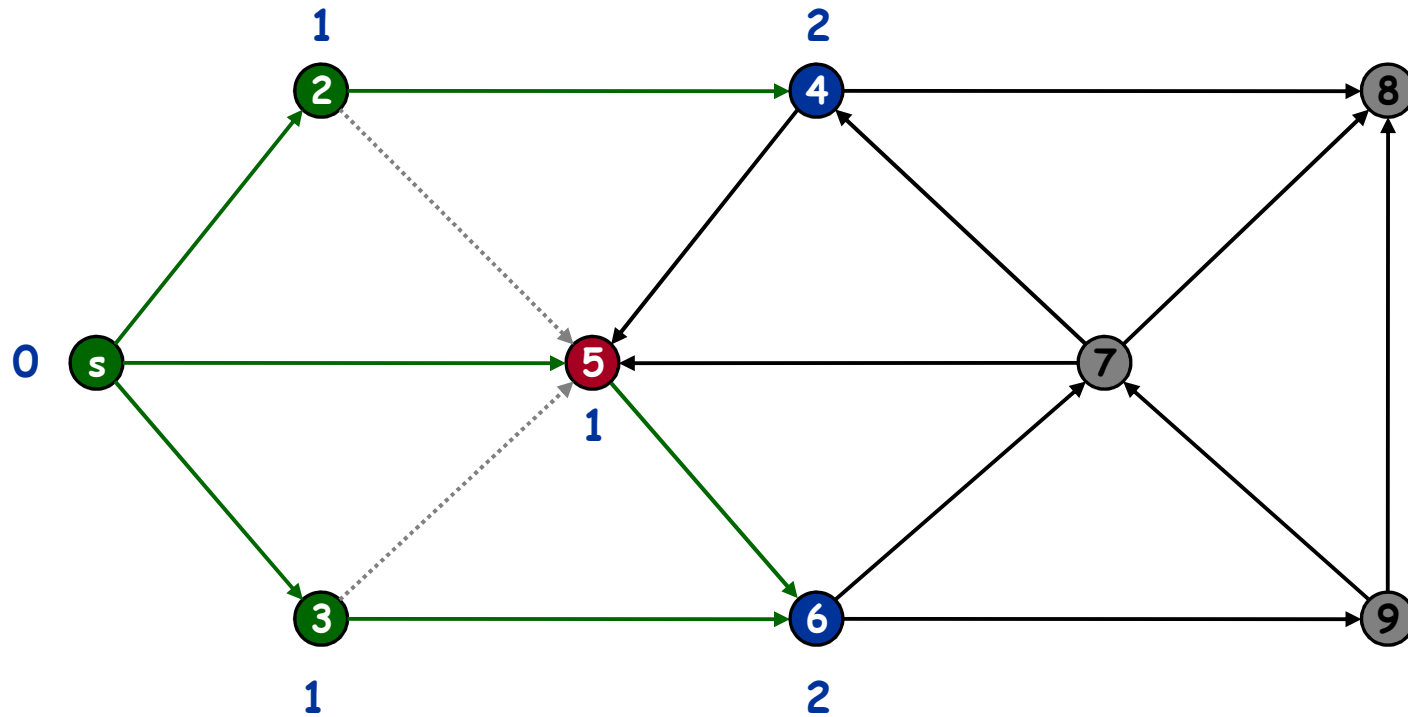
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 5 4 6

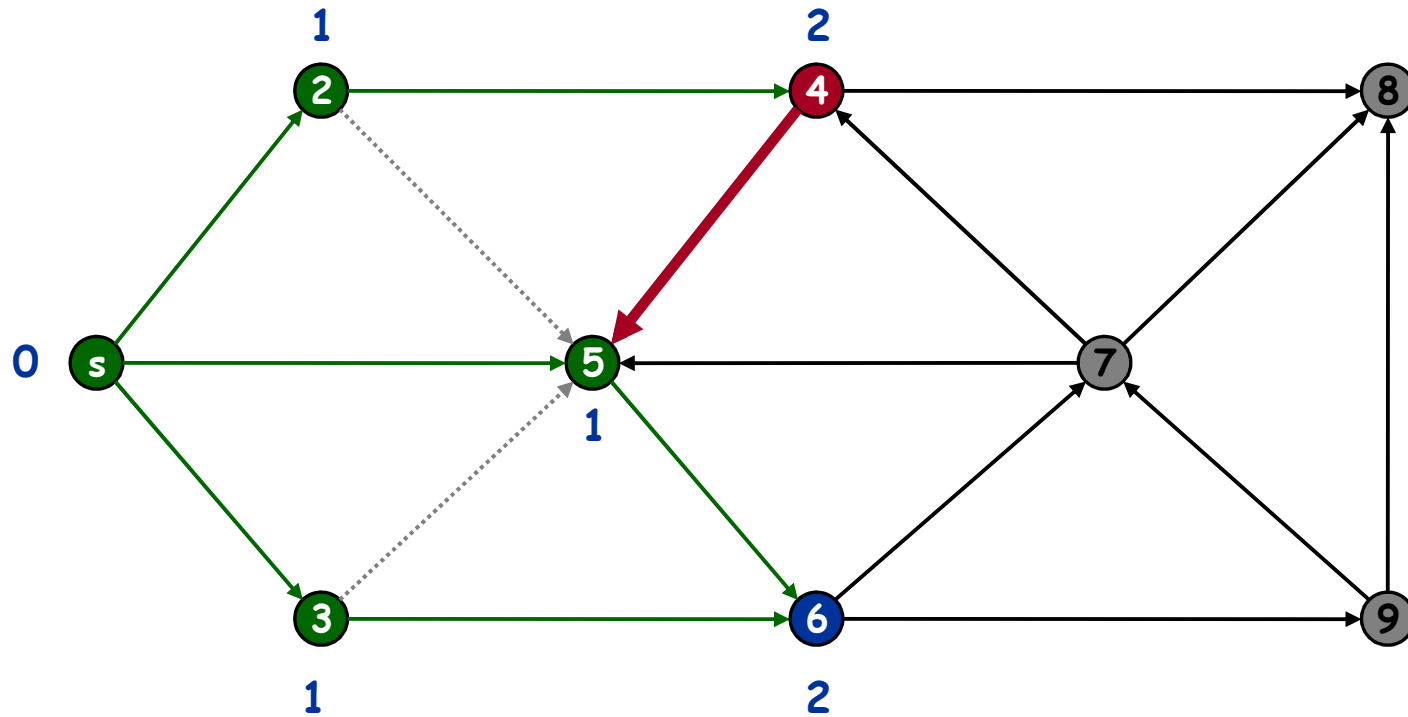
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 5 4 6

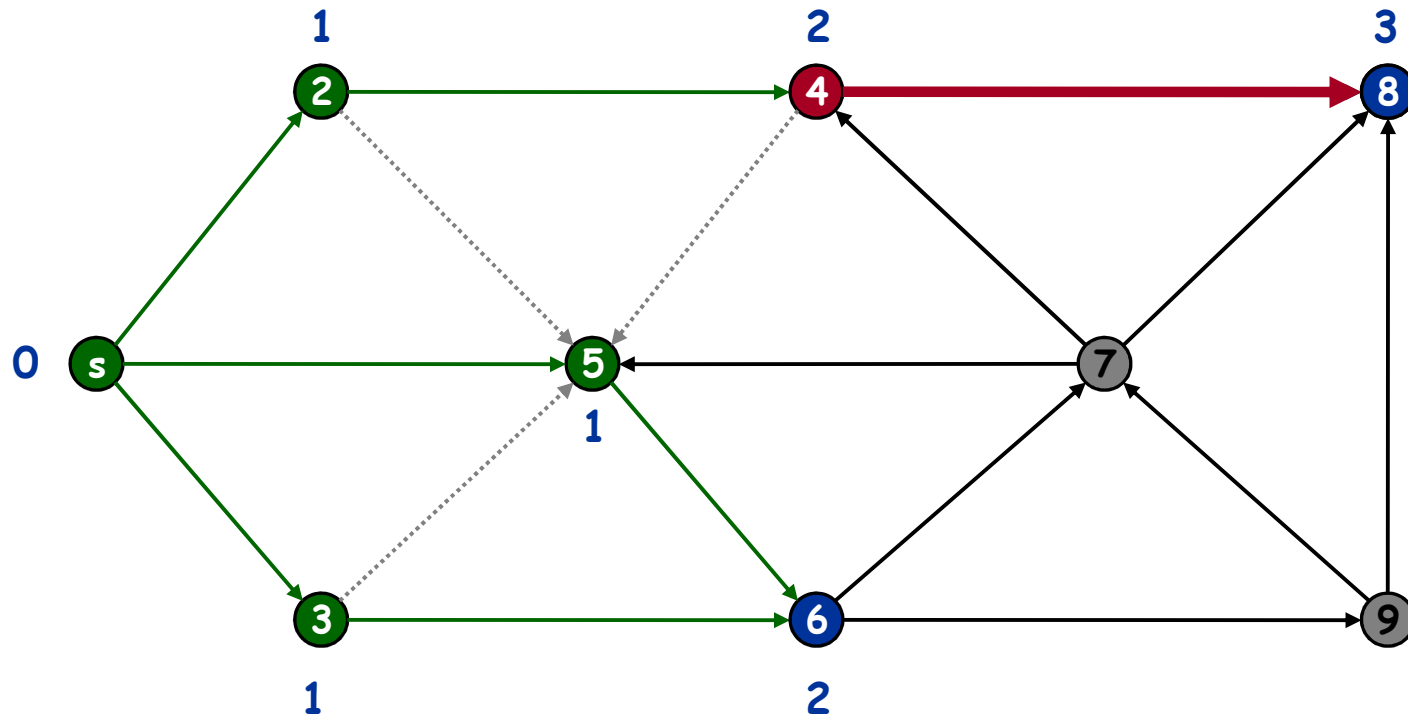
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 4 6

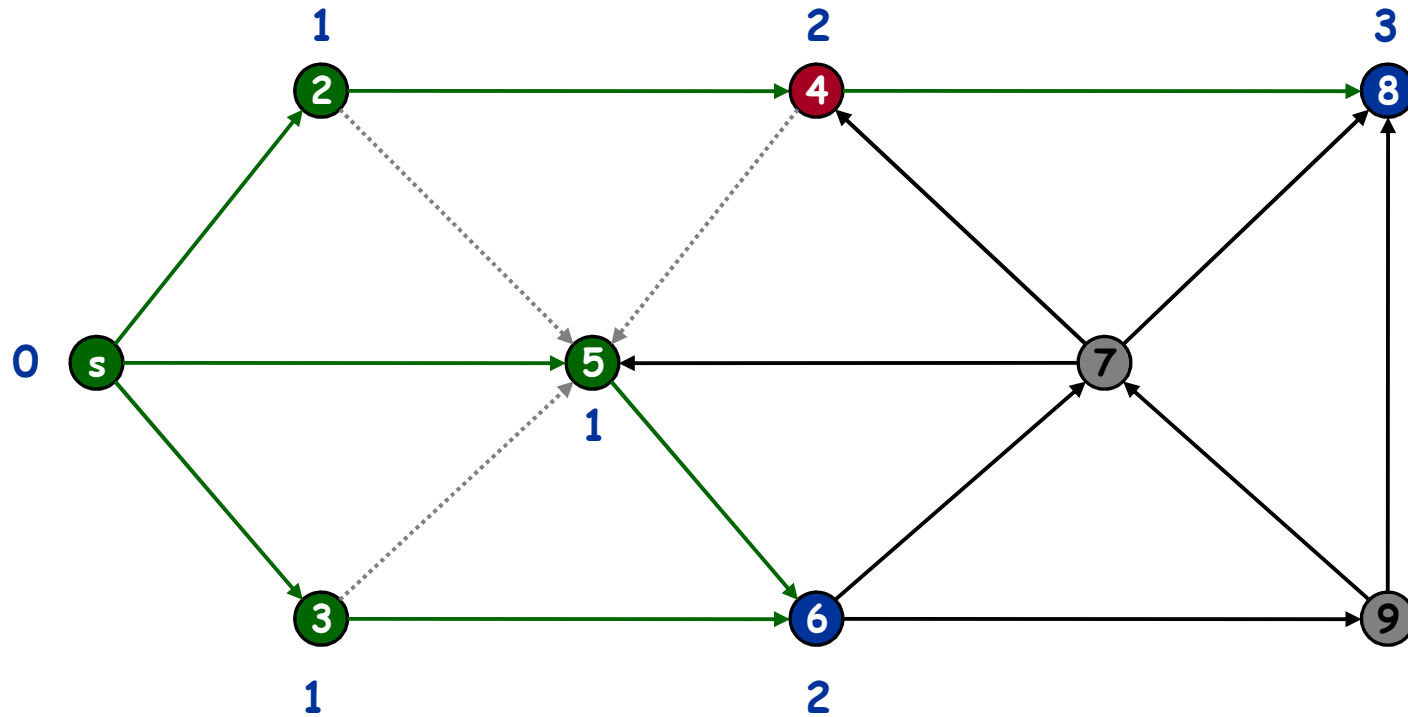
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 4 6

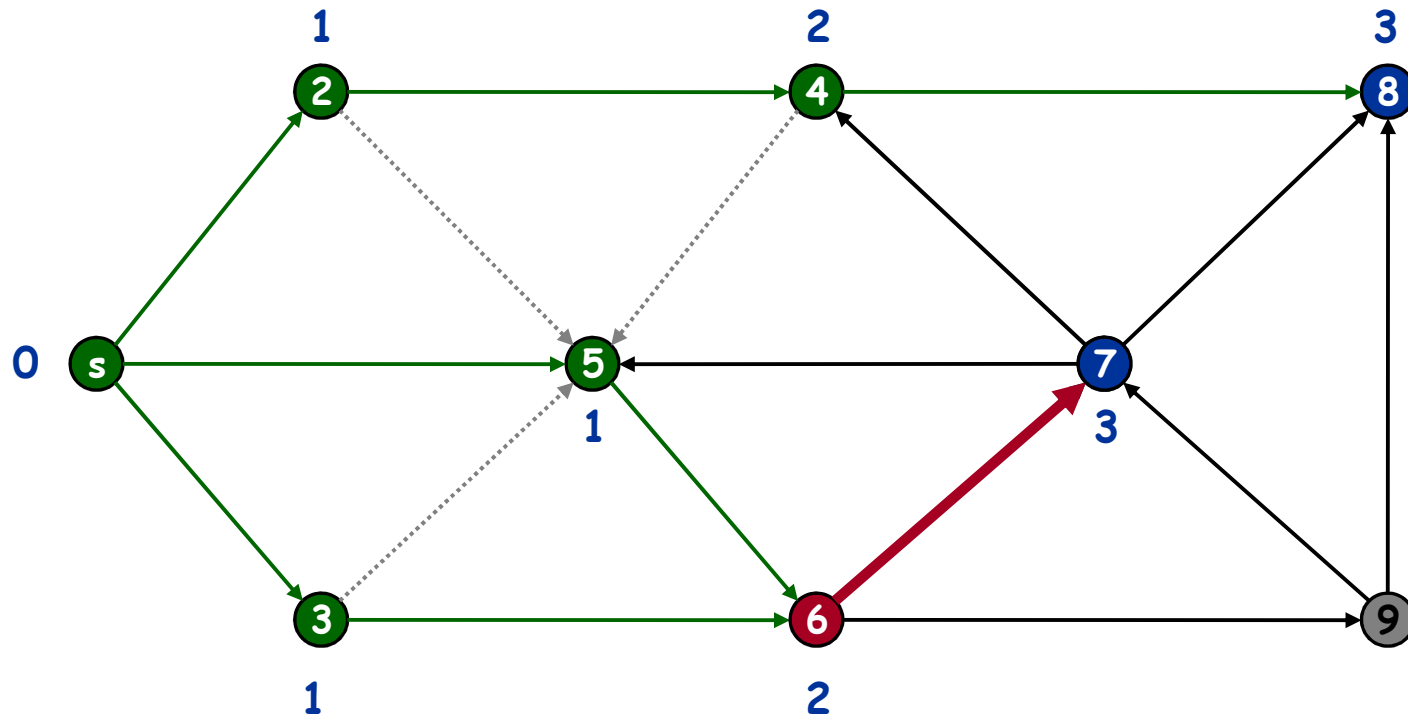
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 4 6 8

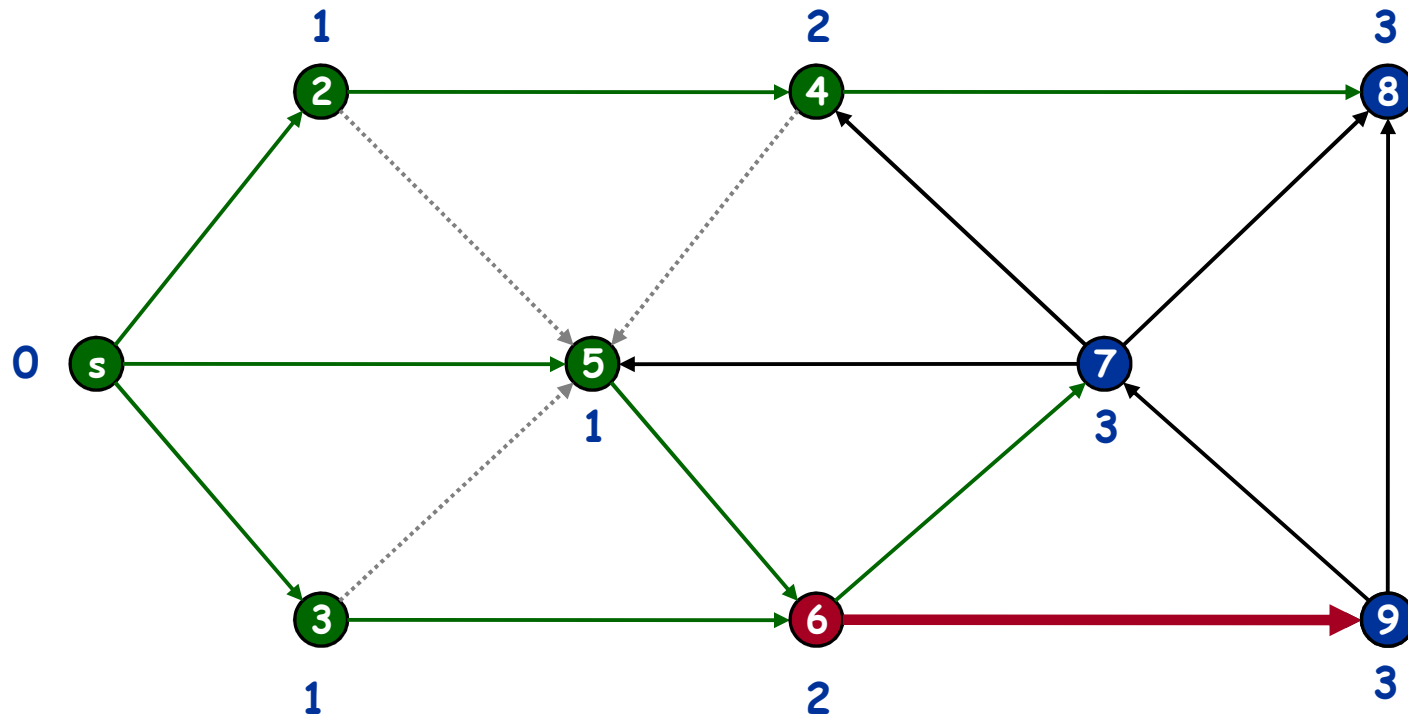
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 6 8

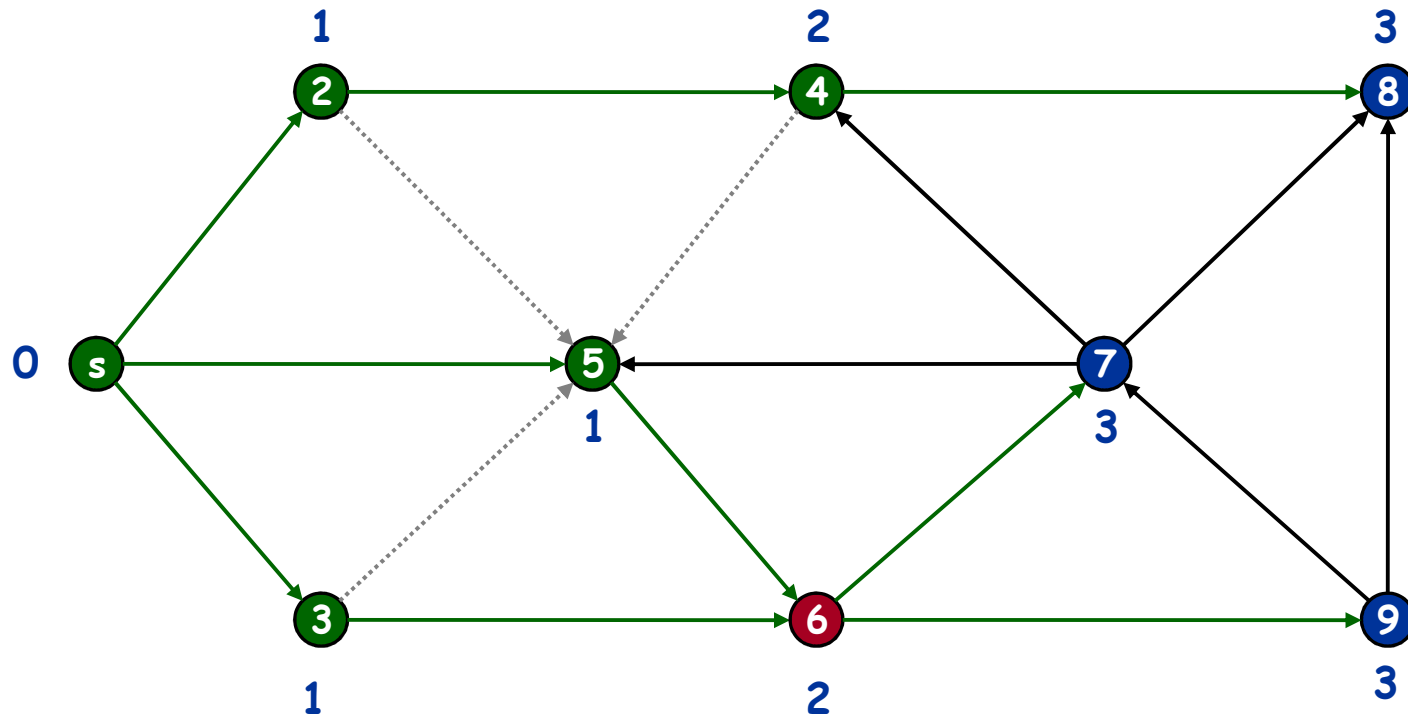
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 6 8 7

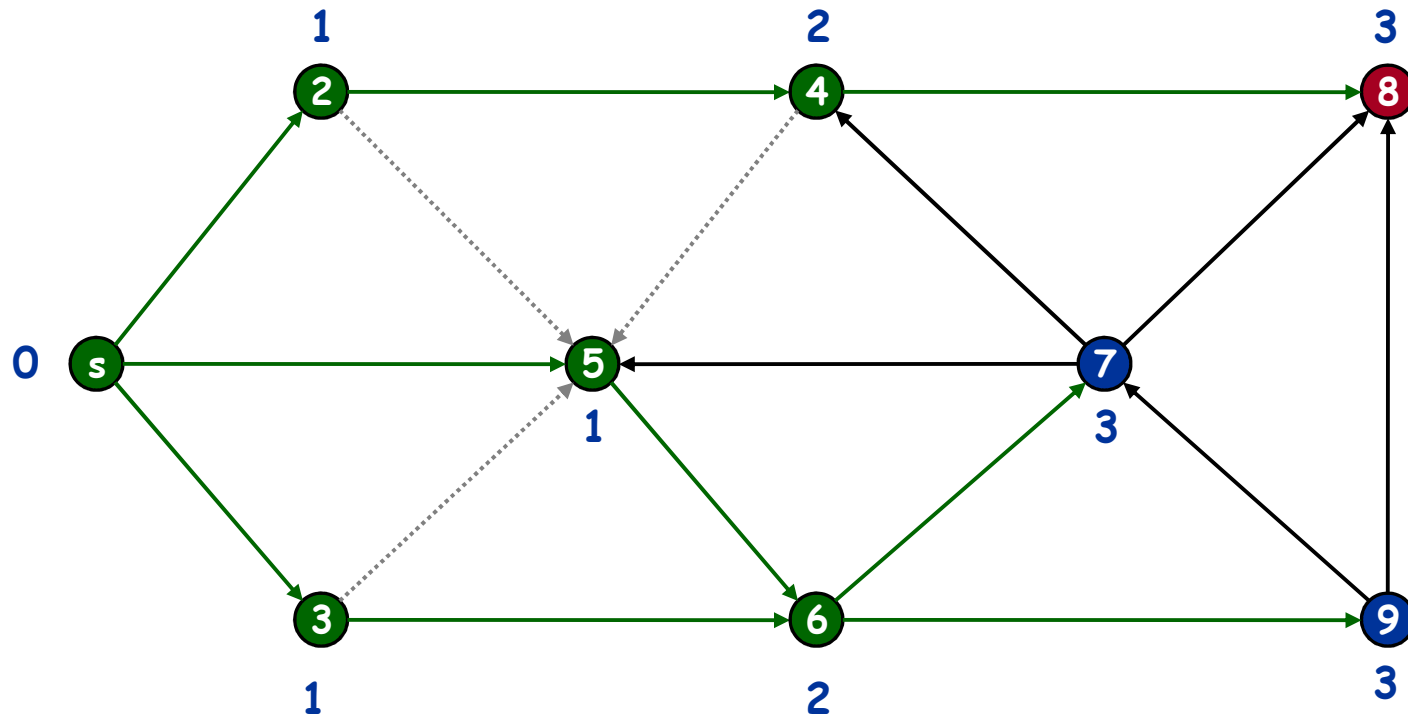
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 6 8 7 9

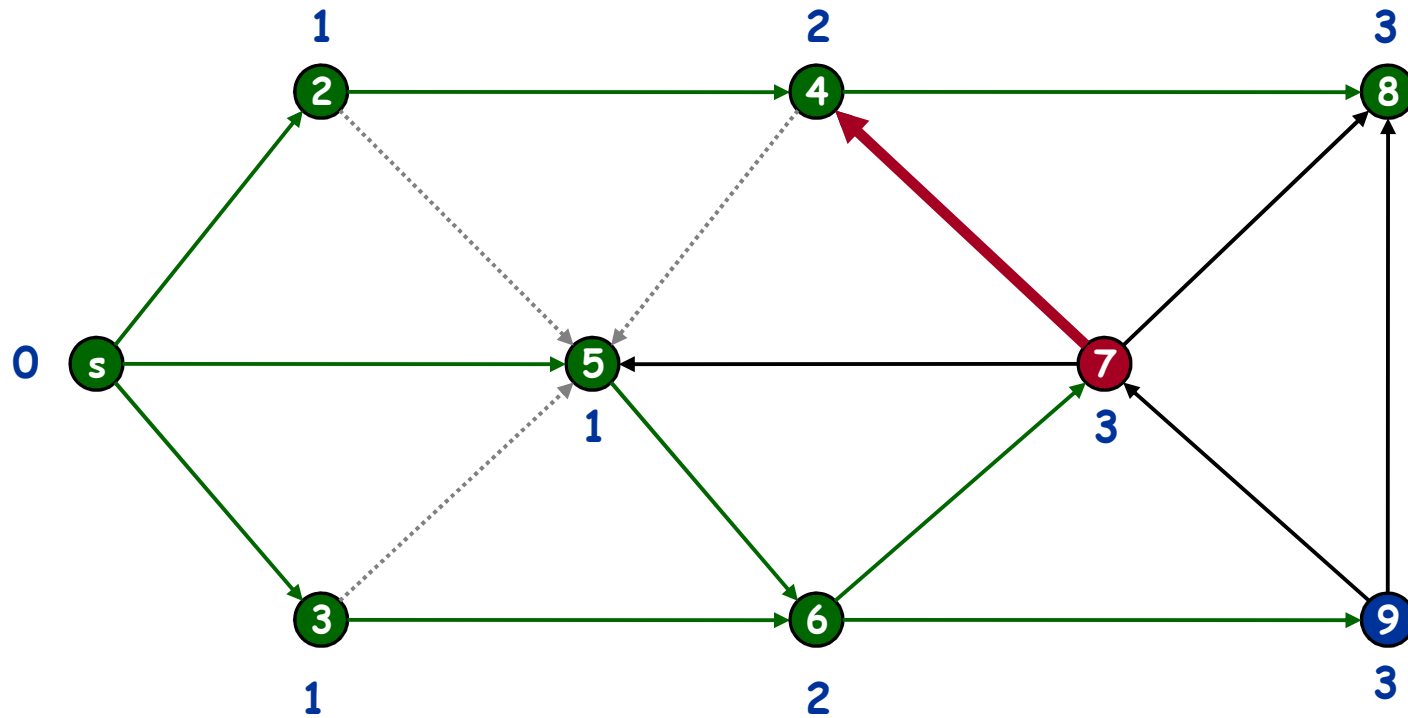
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 8 7 9

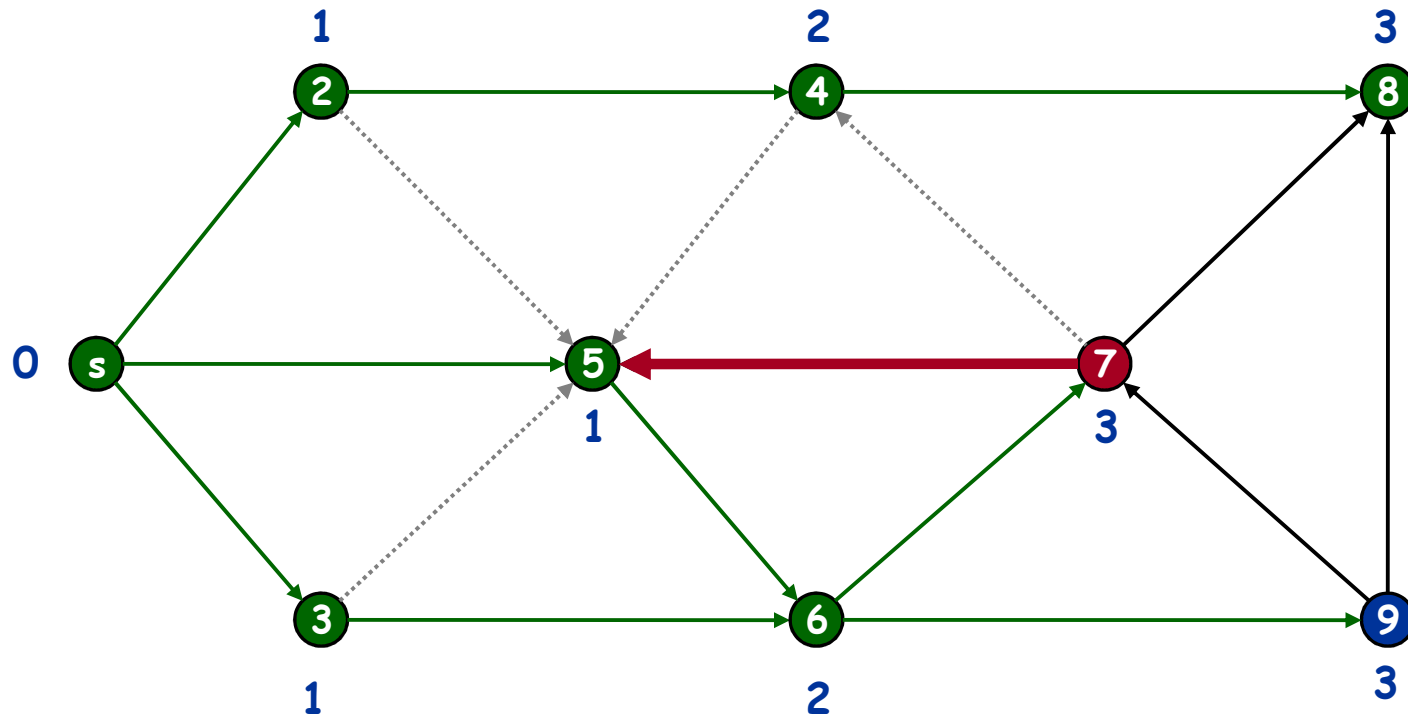
Breadth First Search



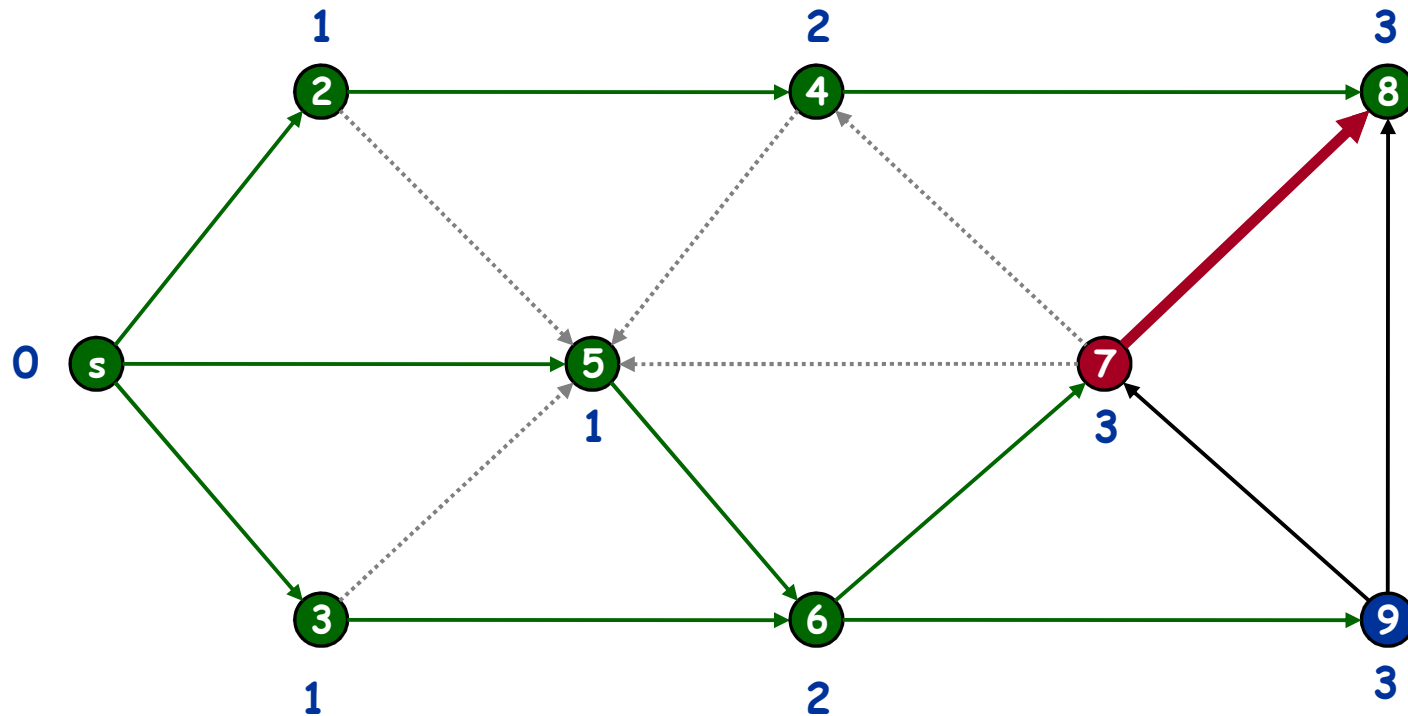
Undiscovered
Discovered
Top of queue
Finished

Queue: 7 9

Breadth First Search



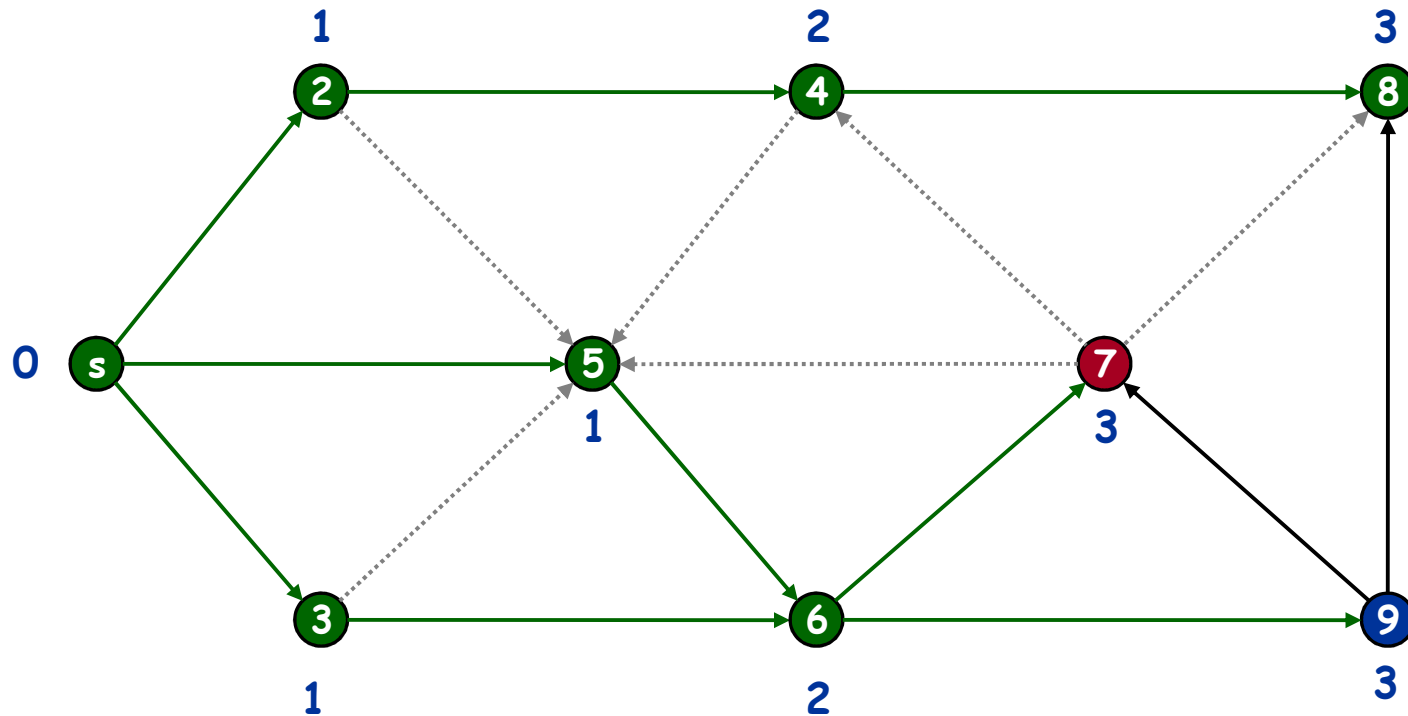
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 7 9

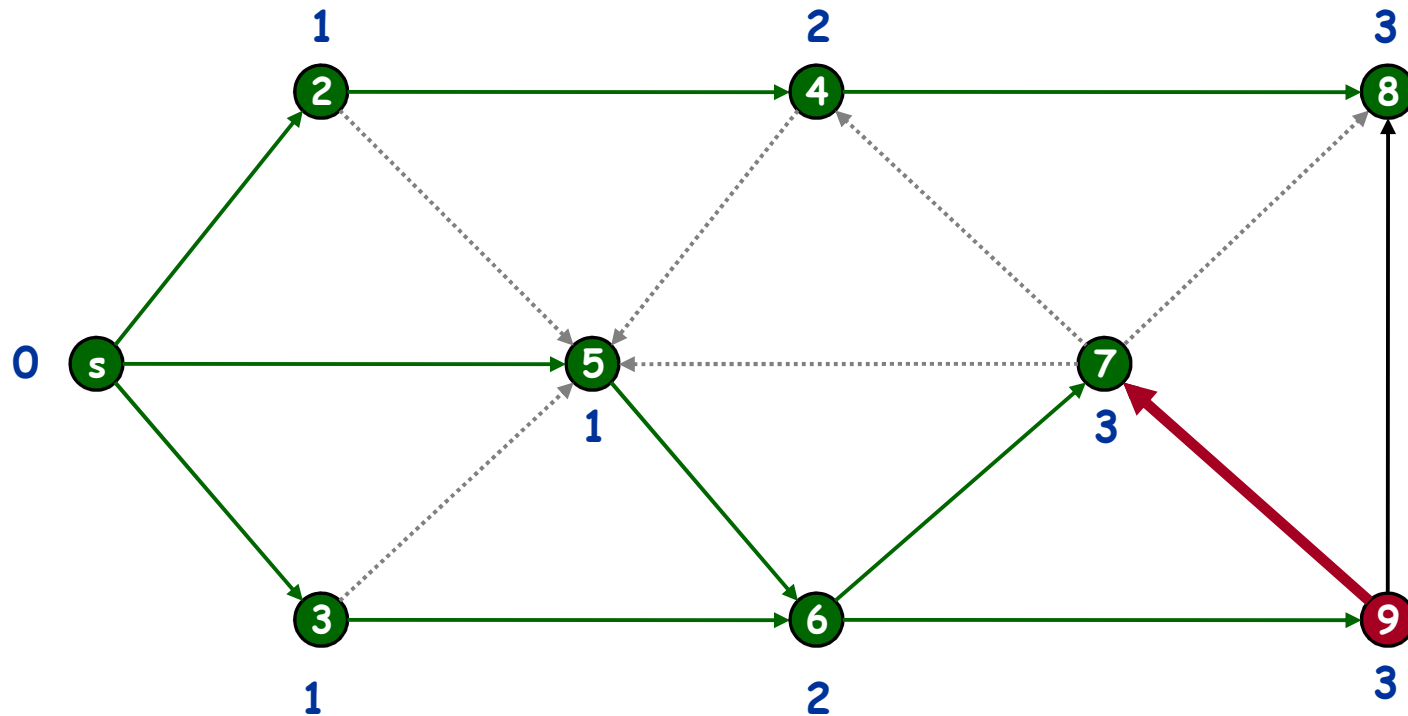
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 7 9

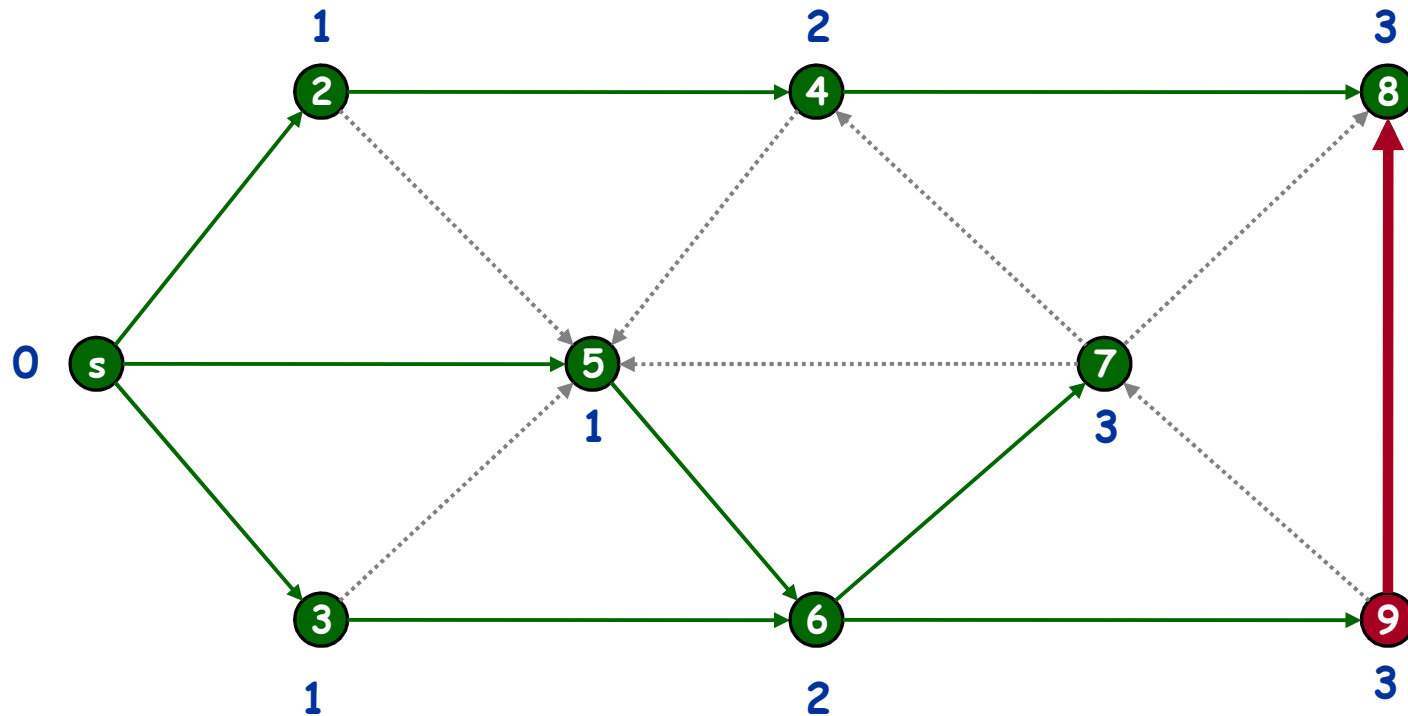
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 9

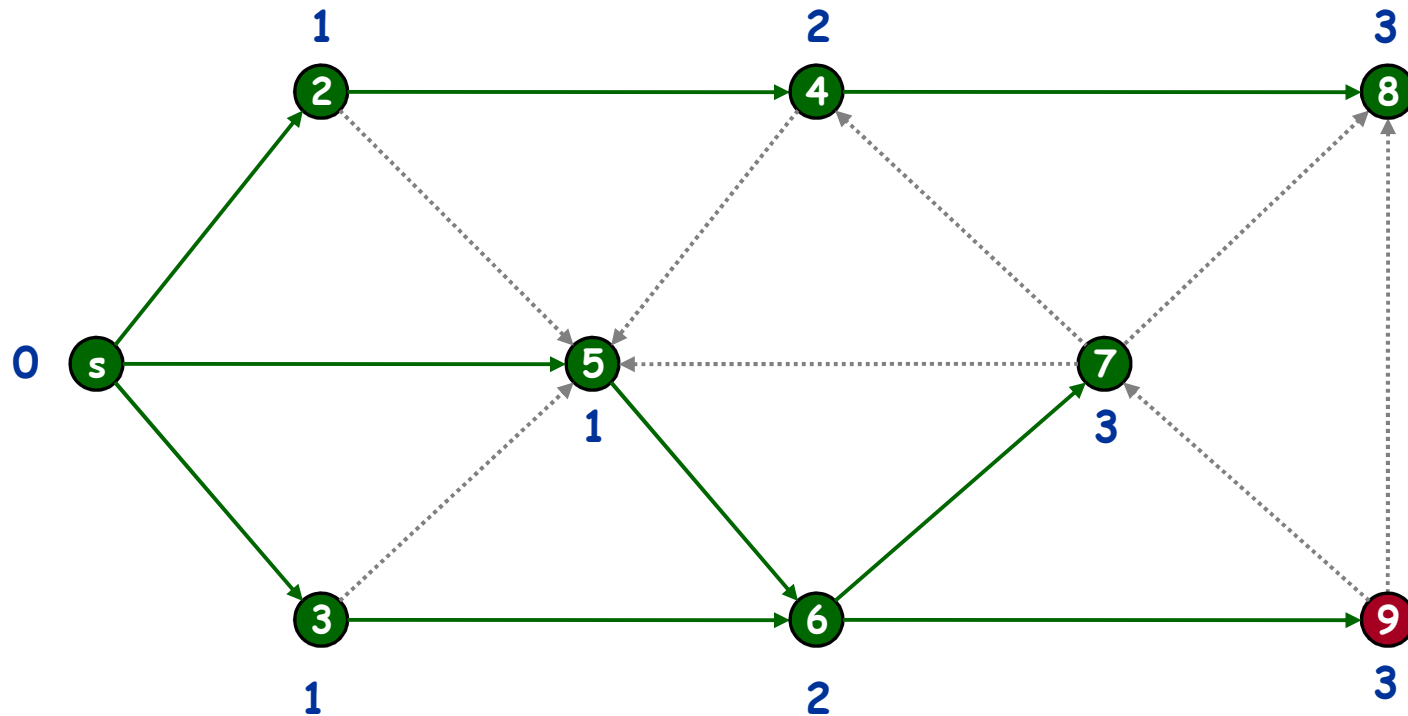
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 9

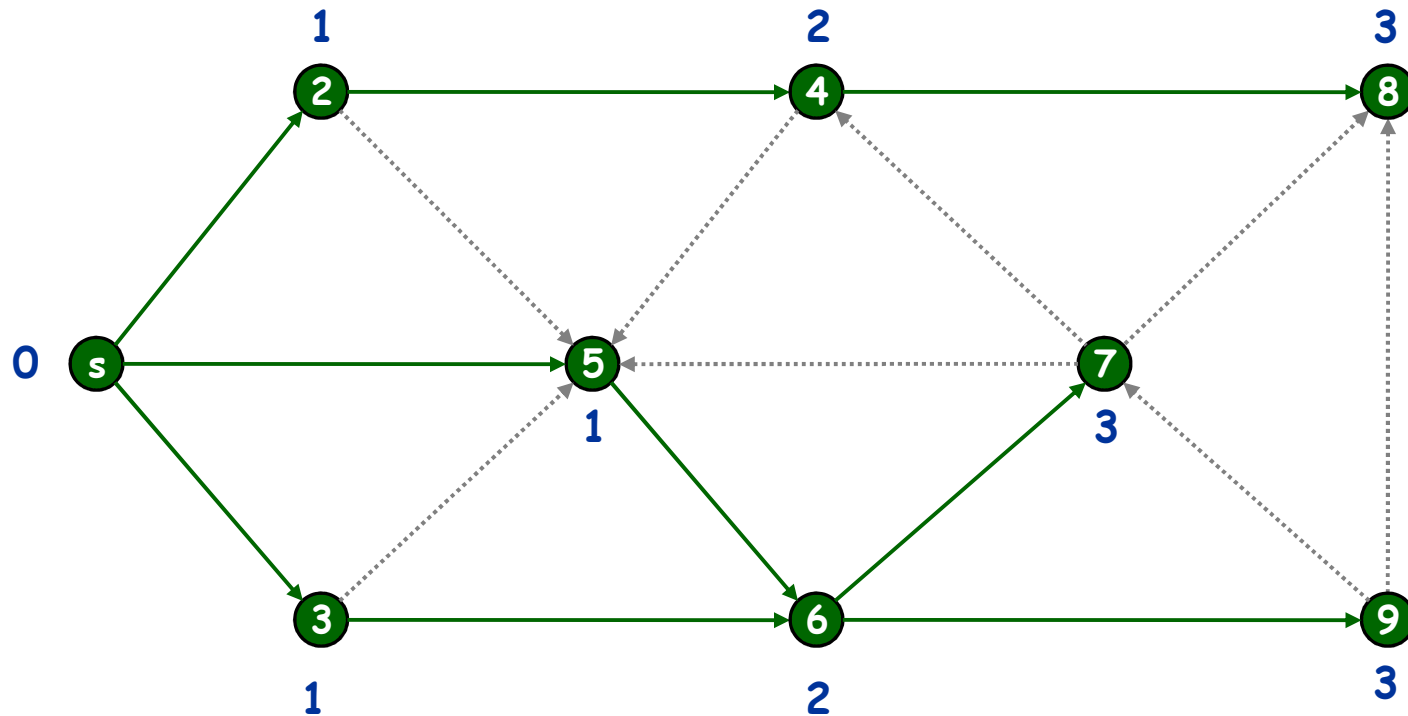
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue: 9

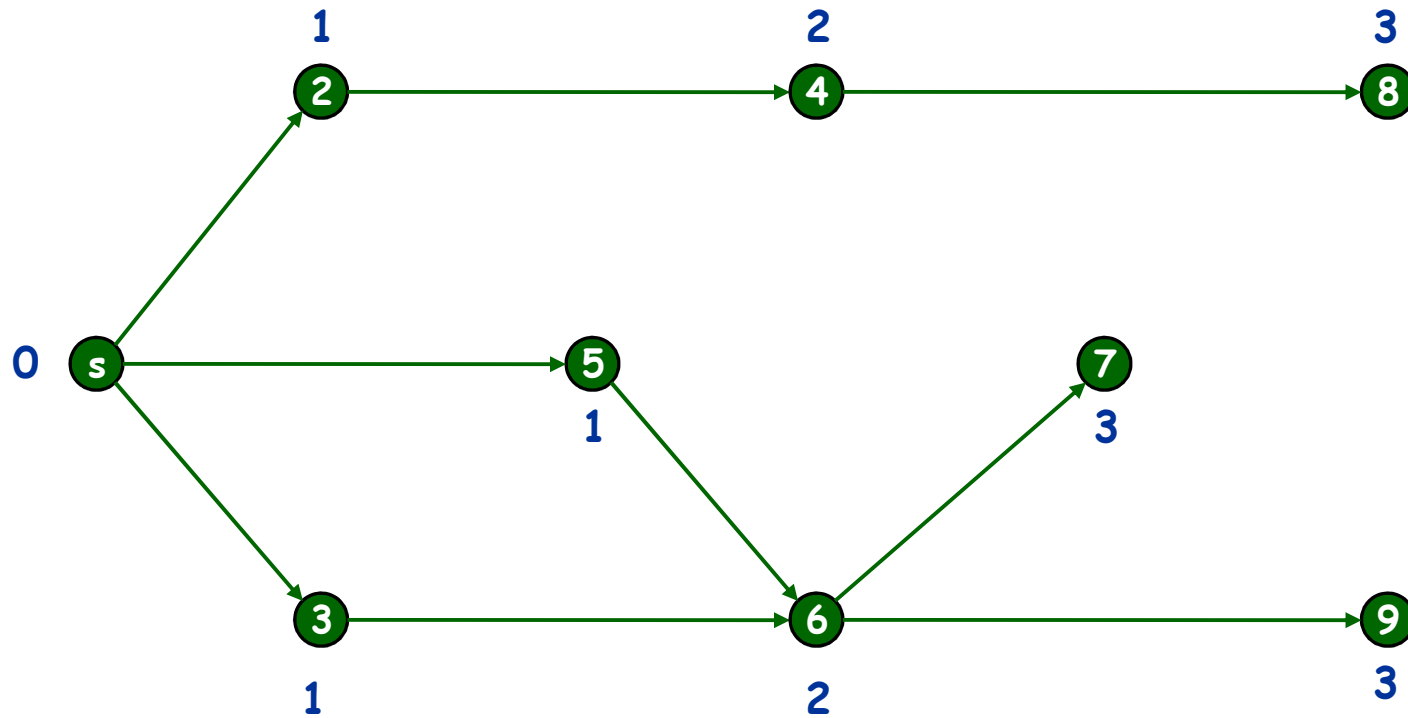
Breadth First Search



Undiscovered
Discovered
Top of queue
Finished

Queue:

Breadth First Search



Summary

- Depth first traversal
- Breadth first traversal