Graph Traversal Algorithms

In this lesson, we will learn the basic logic behind graph traversal and see how it can be done with two most famous graph traversal algorithms.

We'll cover the following

Types of Graph Traversals

Graph traversal means visiting every vertex in the graph. There are two basic techniques used for graph traversal:

1. Breadth First Search (BFS)
2. Depth First Search (DFS)

In order to understand these algorithms, we will have to view graphs from a slightly different perspective.

Any traversal needs a starting point, but a graph does not have a linear structure like lists or stacks. So how do we give graph traversal a better sense of direction?

This is where the concept of **levels** is introduced. Take any vertex as the starting point. This is the lowest level in your search. The **next level** consists of all the vertices adjacent to the starting vertex. A level higher would mean the vertices adjacent to the nodes at the lower level.

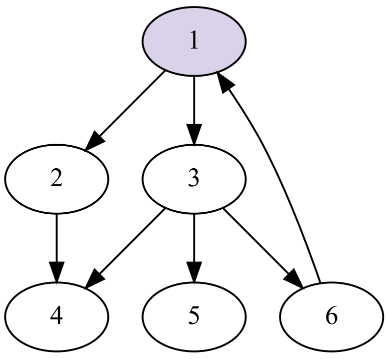
With this is in mind, let's begin our discussion on the two graph traversal algorithms.

1. Breadth First Search

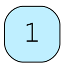
The BFS algorithm earns its name because it grows breadth-wise. All the nodes at a certain level are traversed before moving on to the next level.

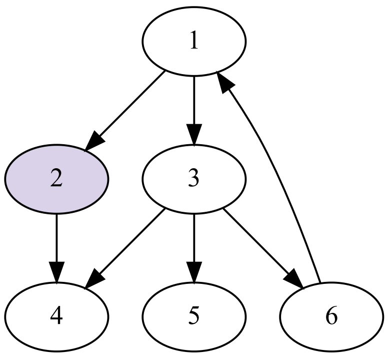
The level-wise expansion ensures that for any starting vertex, you can reach all others, one level at a time.

Let's look at the BFS algorithm in action:

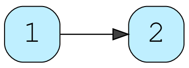


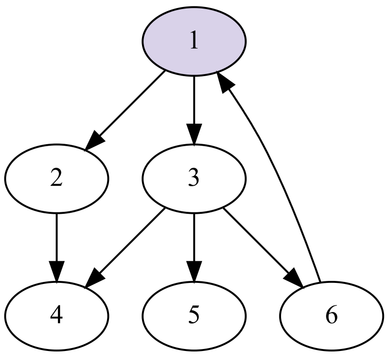
Starting vertex



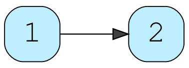


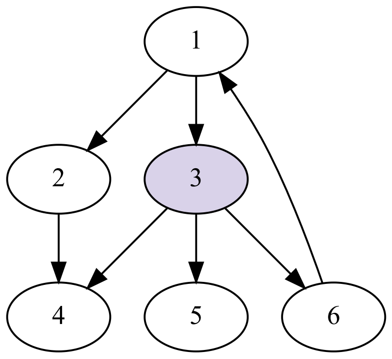
First adjacent node is 2



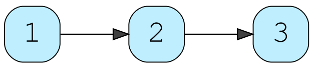


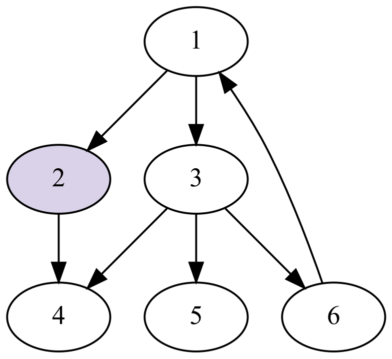
Move back up

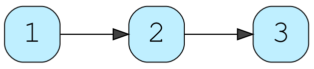


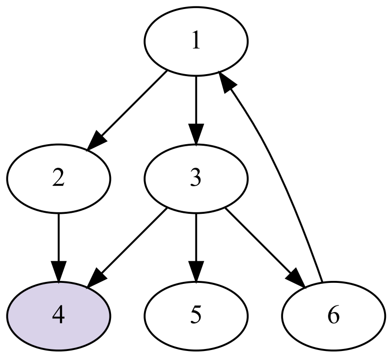


Level completed

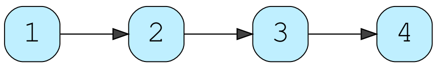


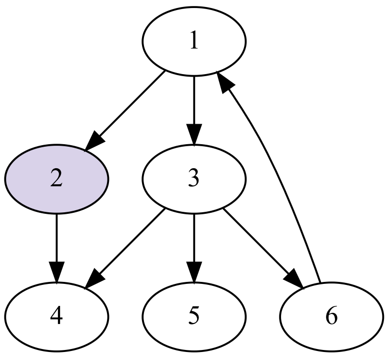


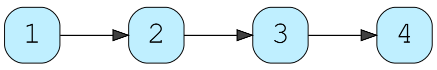


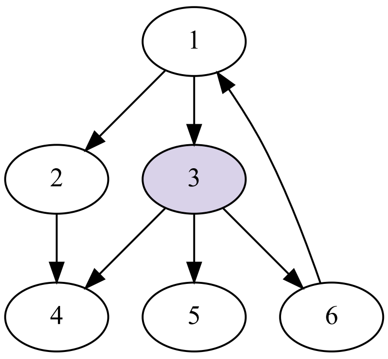


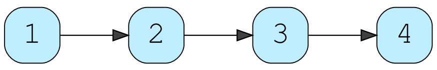
New level

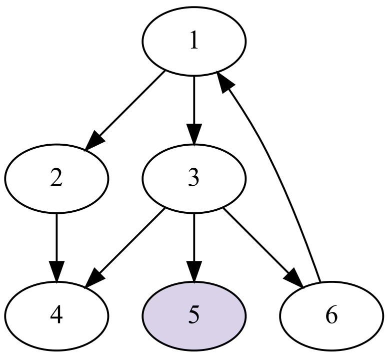


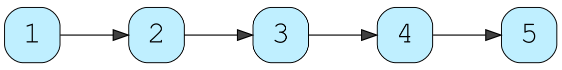

1
2
3
4
5
6

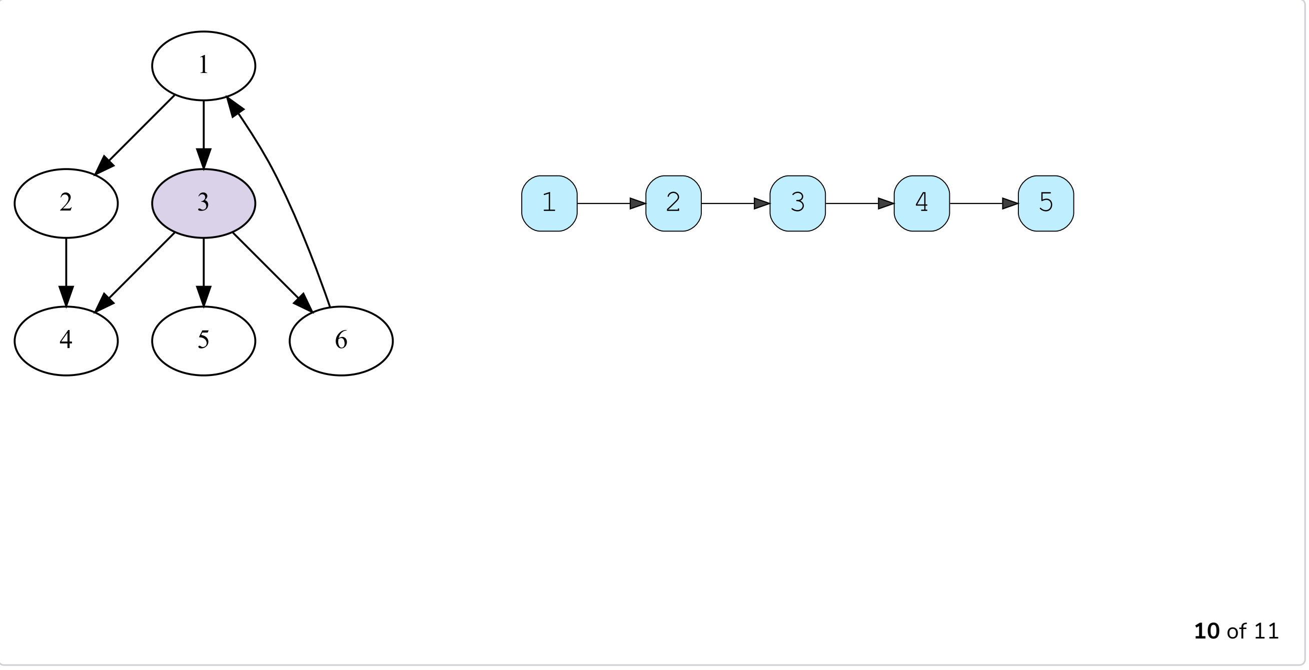

1
2
3
4


1
2
3
4
5
6

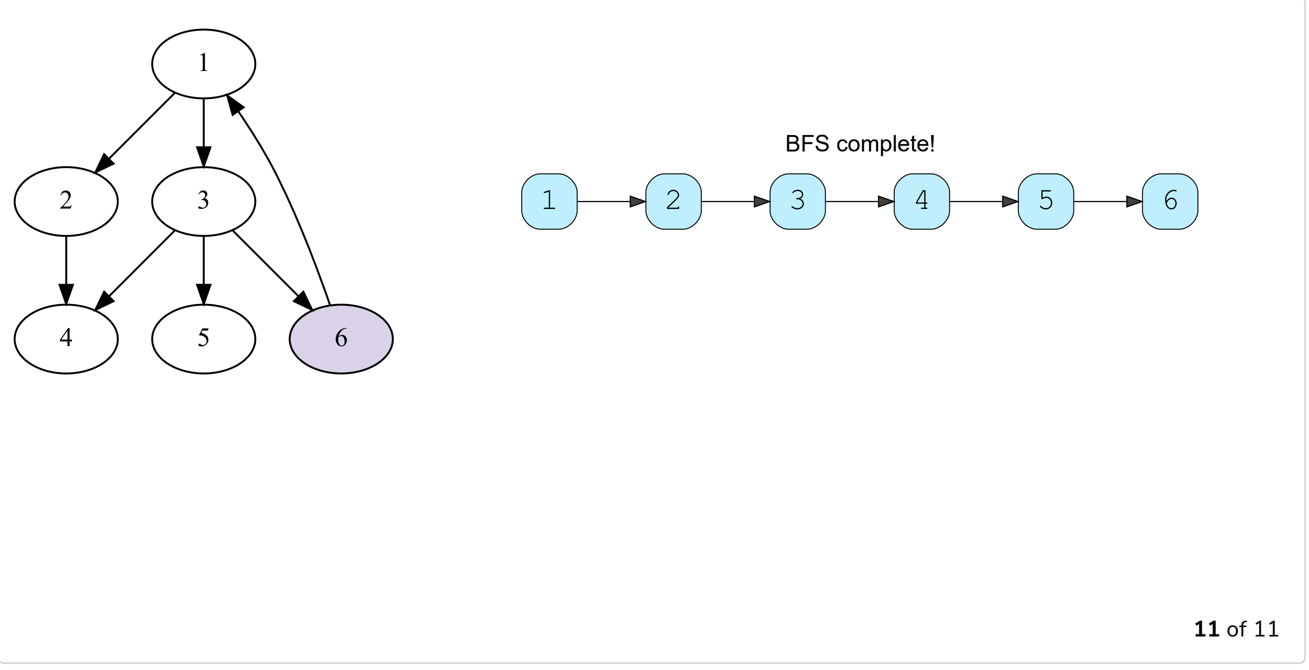

1
2
3
4


1
2
3
4
5
6


1
2
3
4
5



10 of 11



11 of 11

As you can observe, all adjacent vertices are visited before moving on to the next level. If this was an undirected graph, the traversal would look like this:

$1 \rightarrow 2 \rightarrow 3 \rightarrow 6 \rightarrow 4 \rightarrow 5$

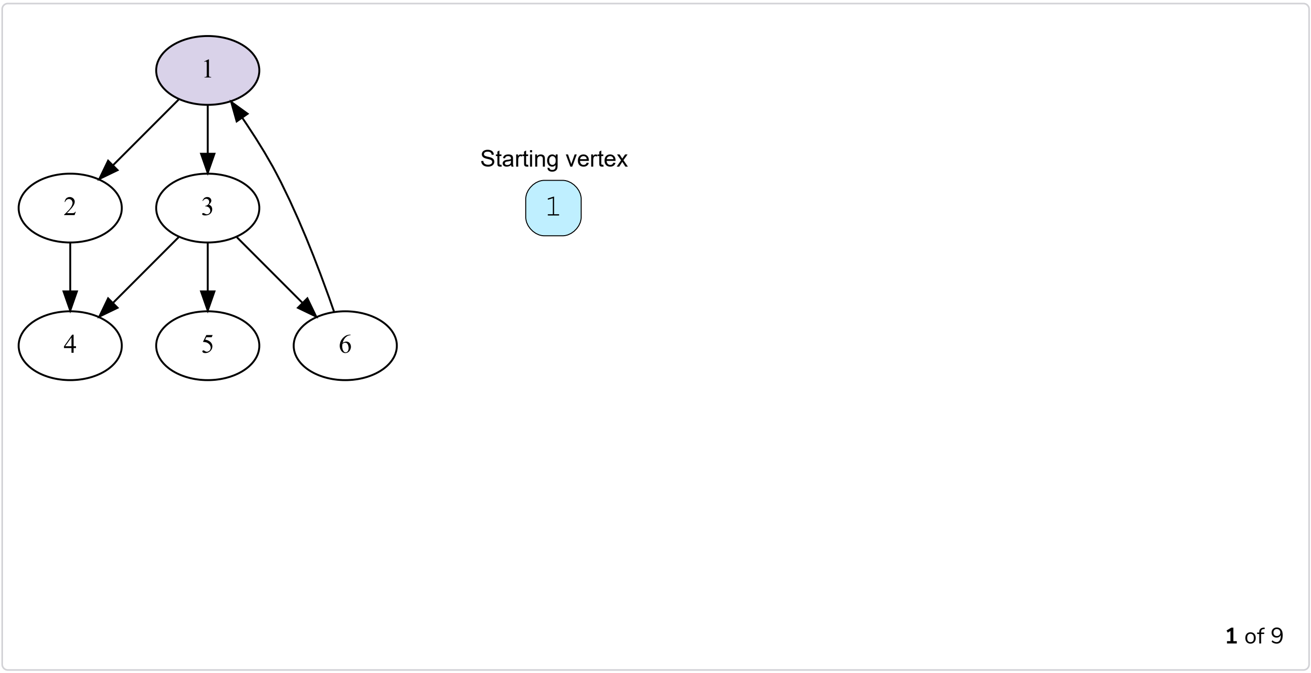
This happens because there is a bidirectional link between 1 and 6, making them adjacent to each other.

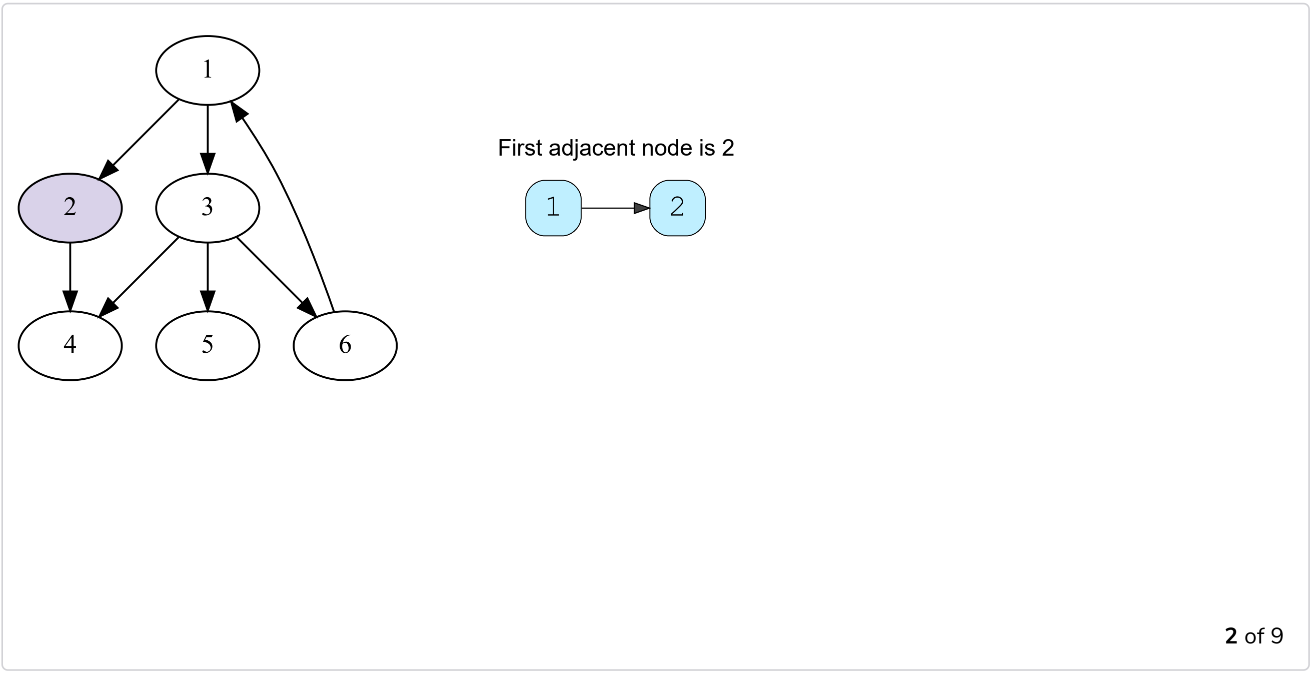
2. Depth First Traversal

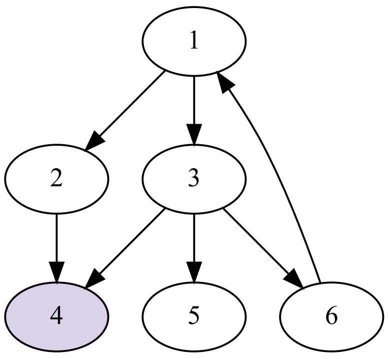
The DFS algorithm is the opposite of BFS in the sense that it grows depth-wise.

Starting from any node, we keep moving to an adjacent node until we reach the farthest level. Then we move back to the starting point and pick another adjacent node. Once again, we probe to the farthest level and move back. This process continues until all nodes are visited.

Here is the visual representation of the DFS algorithm:



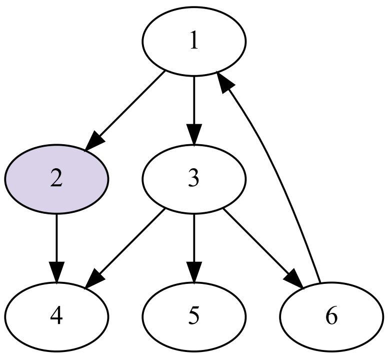




Move deeper

$$\begin{tikzcd} \boxed{1} \arrow[r] & \boxed{2} \arrow[r] & \boxed{4} \end{tikzcd}$$

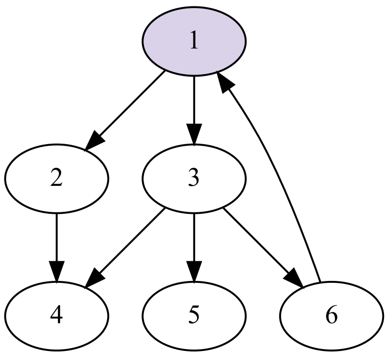
3 of 9



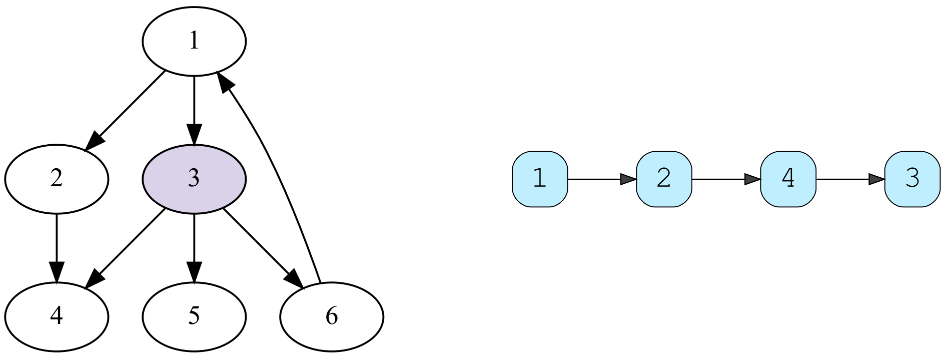
Move back up

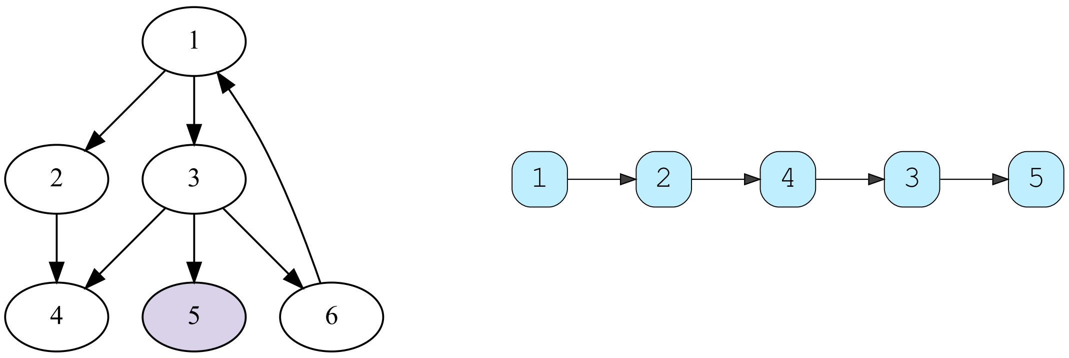
$$\begin{tikzcd} \boxed{1} \arrow[r] & \boxed{2} \arrow[r] & \boxed{4} \end{tikzcd}$$

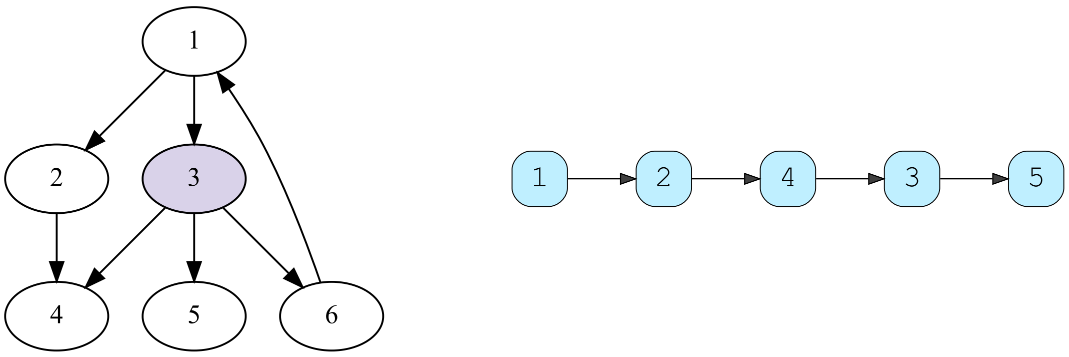
4 of 9

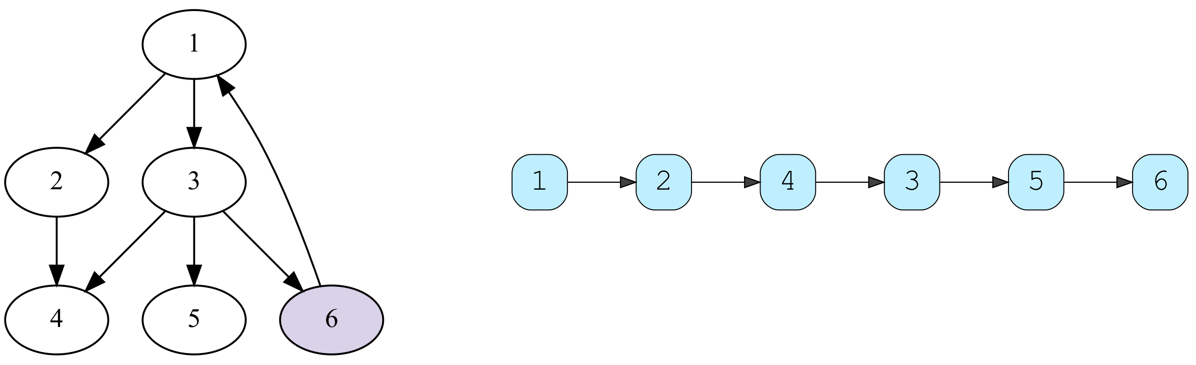


$$\begin{tikzcd} \boxed{1} \arrow[r] & \boxed{2} \arrow[r] & \boxed{4} \end{tikzcd}$$


1
2
3
4
5
6
1
2
4
3


1
2
3
4
5
6
1
2
4
3
5


1
2
3
4
5
6
1
2
4
3
5



9 of 9

Do you see how its behavior is the opposite of BFS? At every step, a node on a new level is visited.

In the upcoming lesson, we will implement the BFS algorithm in Python.

← Back
What is a Bipartite Graph?

Next →
Challenge 1: Implement Breadth First ...

✓ Mark as Completed