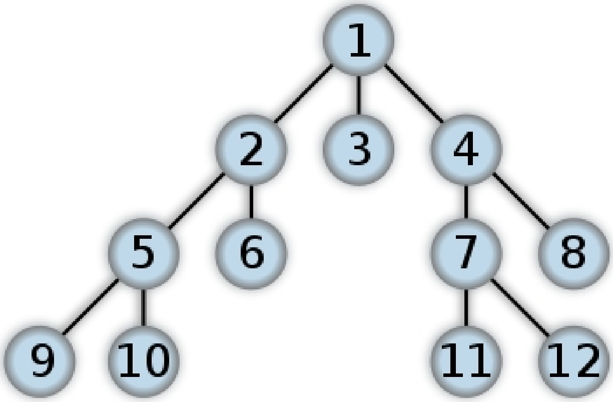
Breadth First Search (BFS)

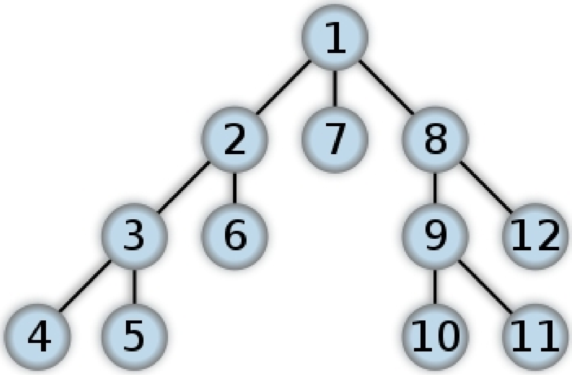
In **Breadth First Search**, the algorithm starts at the root node and then traverses through every node of every level finishing one level of the tree at a time.



The order of the numbers tells us how BFS progresses.

Depth First Search (DFS)

In **Depth First Search**, the algorithm starts at the root node and traverses to the furthest node of the tree to find the goal, if the goal isn't found, the algorithm backtracks every node that has unexplored children/leaf-nodes.



The order of the numbers tells us how DFS progresses.

Comparison between DFS and BFS

In BFS the algorithm uses a FIFO data structure called a queue. This algorithm stores all the nodes that are traversed and hence the size of the memory that needs to be used is very large. On the other hand, DFS uses a LIFO data structure called a stack.

On whether which algorithm is better, it depends on that data the algorithm is used.

In the maps given:

Map/Algorithm	**DFS**	**BFS**
Map 1	11	14
Map 2	7	12
Map 3	29	29

In **Map 1**, the BFS has to traverse through all the nodes to reach the goal node, hence the time taken by BFS is more than DFS.

In **Map 2,** BFS takes longer because the algorithm expands in such a way that it traverses all nodes at a particular height before it reaches the last level, hence the time taken is longer.

In **Map 3**, DFS takes almost the same time as BFS because the algorithm traverses the deepest node on the left side of the tree first and then comes back to the root node and then traverses the the deepest node on the other half of the tree whereas, BFS traverses every height in the map, making computation time almost similar to find the last goal node.