Interview Cake

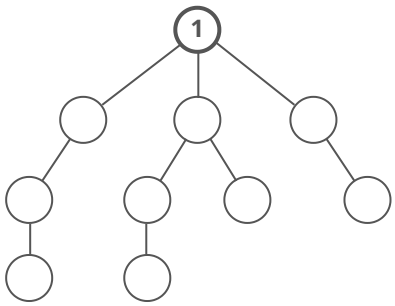
← course home (/table-of-contents)

Breadth-First Search (BFS) and Breadth-First Traversal

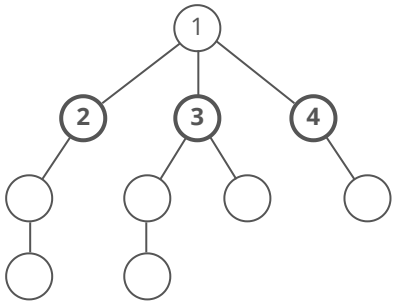
Breadth-first search (BFS) is a method for exploring a tree or graph. In a BFS, you first explore all the nodes one step away, then all the nodes two steps away, etc.

Breadth-first search is like throwing a stone in the center of a pond. The nodes you explore "ripple out" from the starting point.

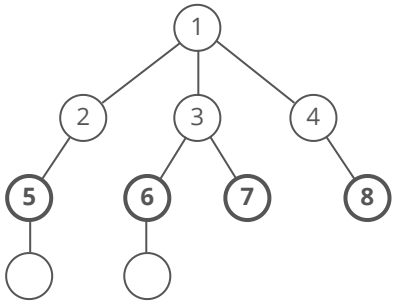
Here's a how a BFS would traverse this tree, starting with the root:



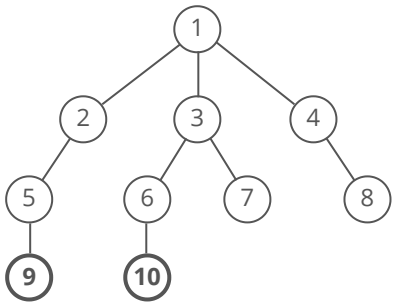
We'd visit all the immediate children (all the nodes that're one step away from our starting node):



Then we'd move on to all *those* nodes' children (all the nodes that're *two steps* away from our starting node):



And so on:



Until we reach the end.

Breadth-first search is often compared with **depth-first search**.

Advantages:

- A BFS will find the **shortest path** between the starting point and any other reachable node. A depth-first search will not necessarily find the shortest path.

Disadvantages

- A BFS on a binary tree *generally* requires more memory than a DFS.

← course home (/table-of-contents)

Next up: Depth-First Search (DFS) ➡ (/concept/dfs?course=fc1&section=trees-graphs)