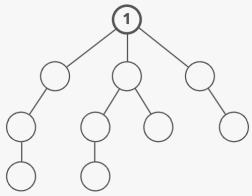
← course home

Depth-First Search (DFS) and Depth-First Traversal

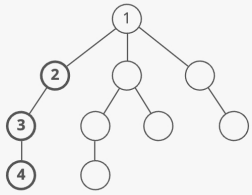
Depth-first search (DFS) is a method for exploring a tree or graph. In a DFS, you go as deep as possible down one path before backing up and trying a different one.

Depth-first search is like walking through a corn maze. You explore one path, hit a dead end, and go back and try a different one.

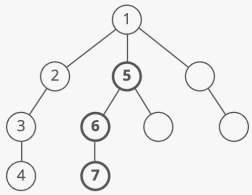
Here's a how a DFS would traverse this tree, starting with the root:



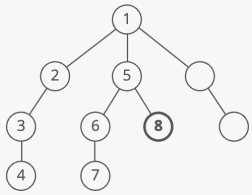
We'd go down the first path we find until we hit a dead end:



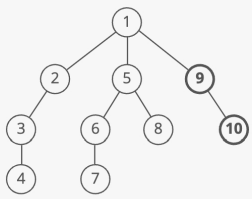
Then we'd do the same thing again—go down a path until we hit a dead end:



And again:



And again:



Until we reach the end.

Depth-first search is often compared with **breadth-first search**.

Advantages:

- Depth-first search on a binary tree *generally* requires less memory than breadth-first.
- Depth-first search can be easily implemented with recursion.

Disadvantages

- A DFS doesn't necessarily find the shortest path to a node, while breadth-first search does.

← course home

Next up: Balanced Binary Tree →