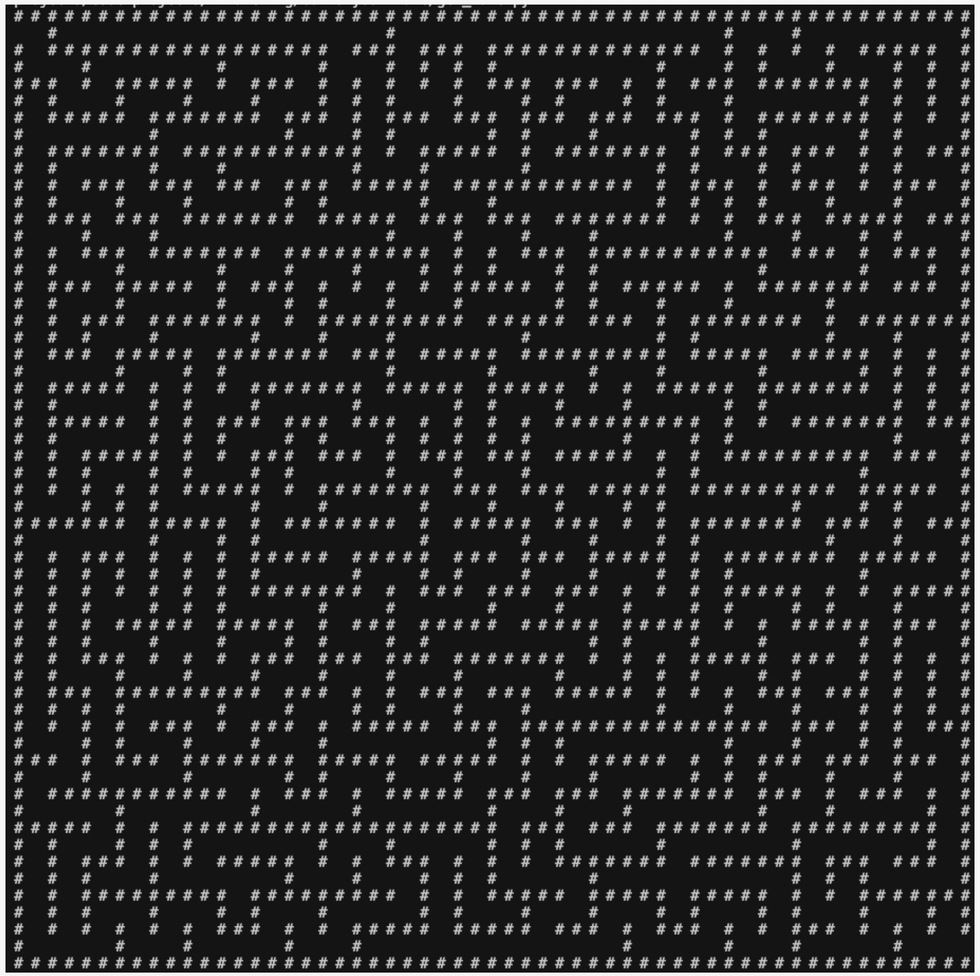
An introduction to graph traversal

Rose Nasrawi | Journal Club 29.1.24

How would you solve this maze?

Start →



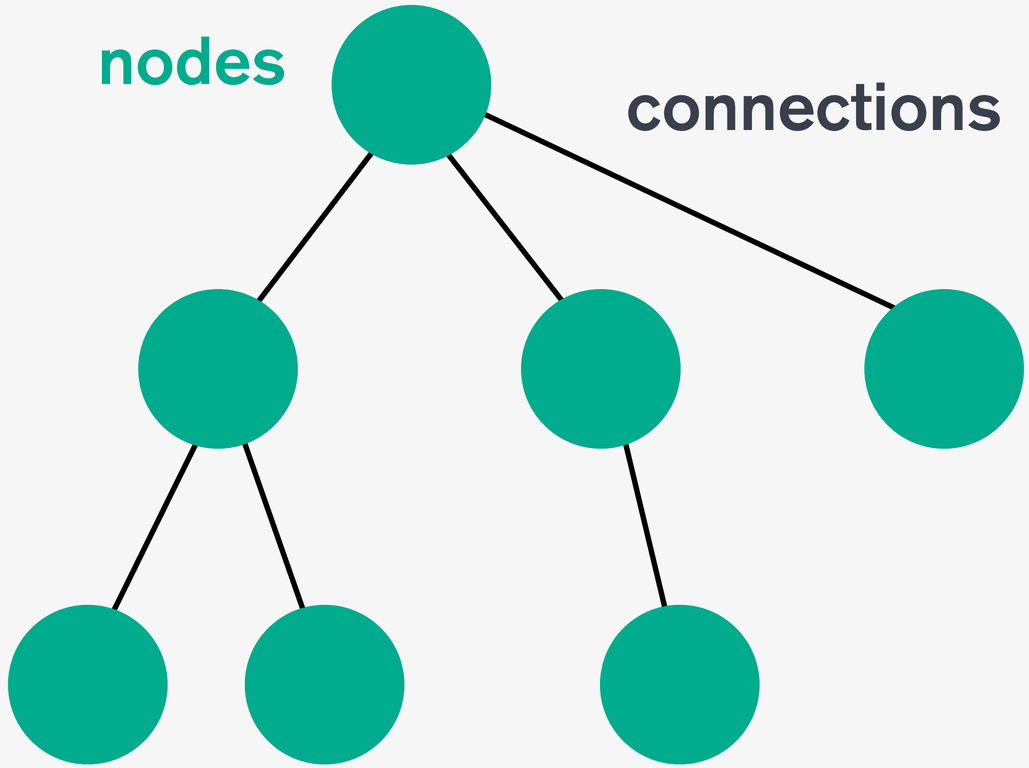
→ End

How would you solve *this* maze?

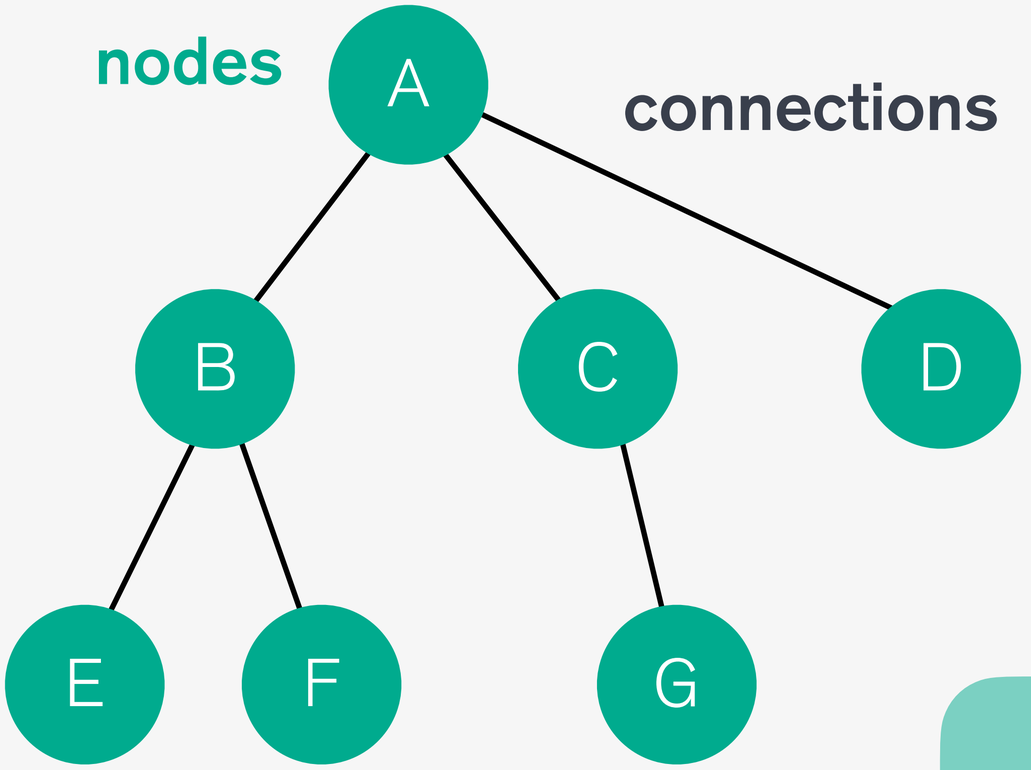
Start →

→ End

An alternative way of drawing a maze: Graph

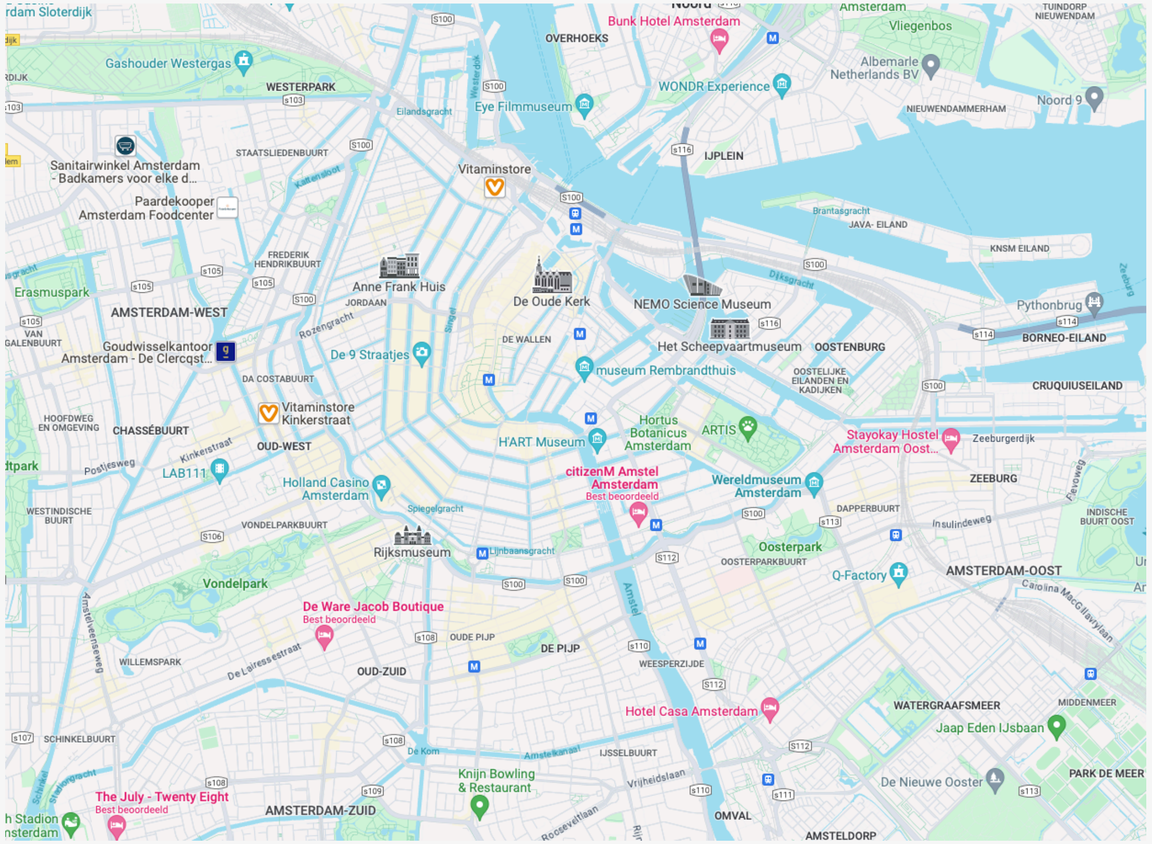


Path finding: Graph traversal

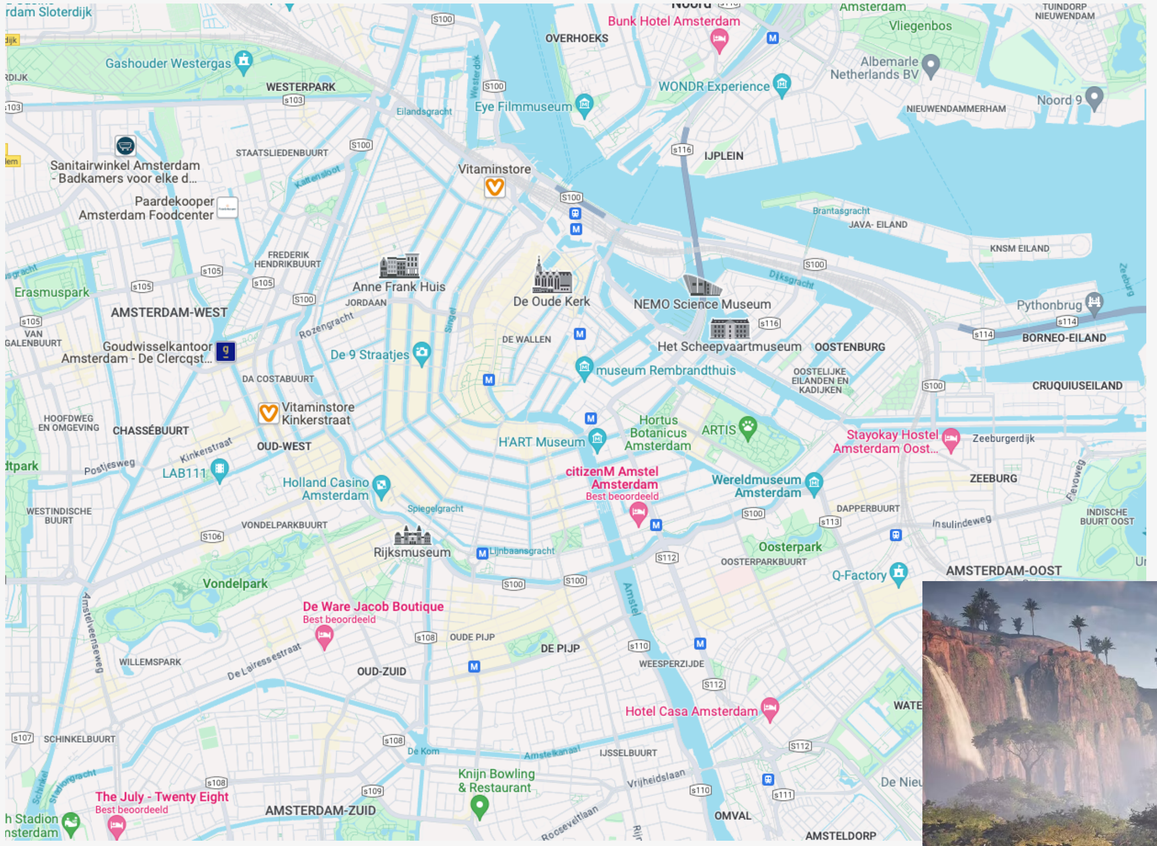


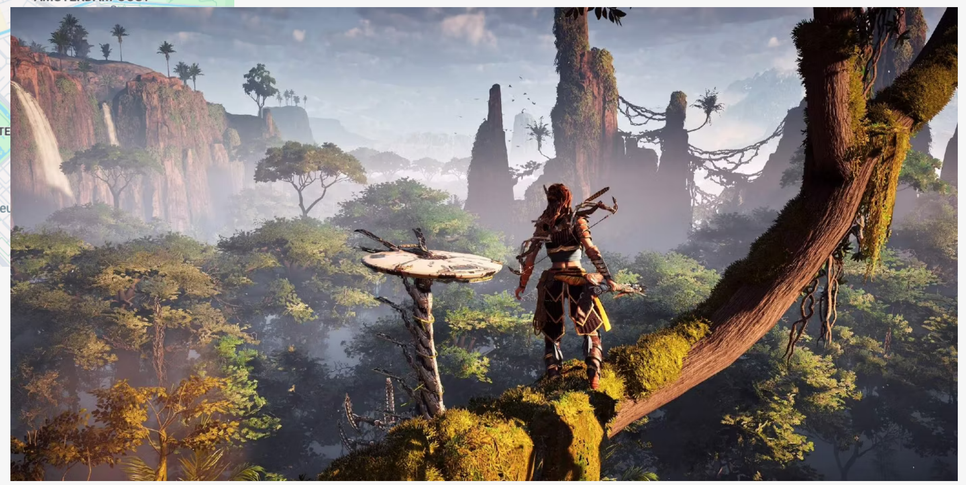
How do I get from E to D?

Path finding can be complex!

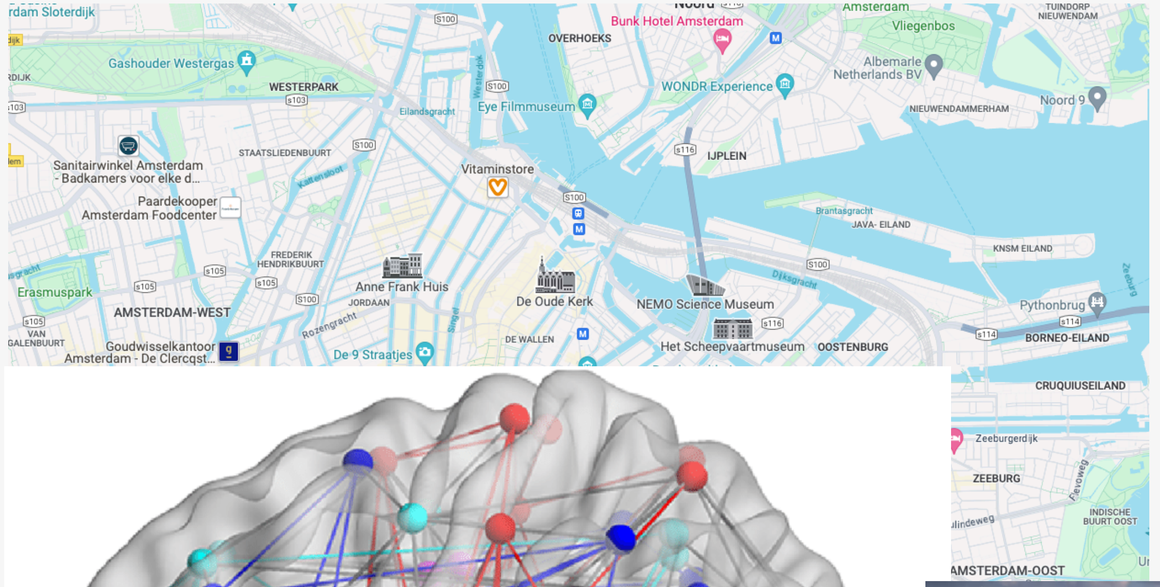


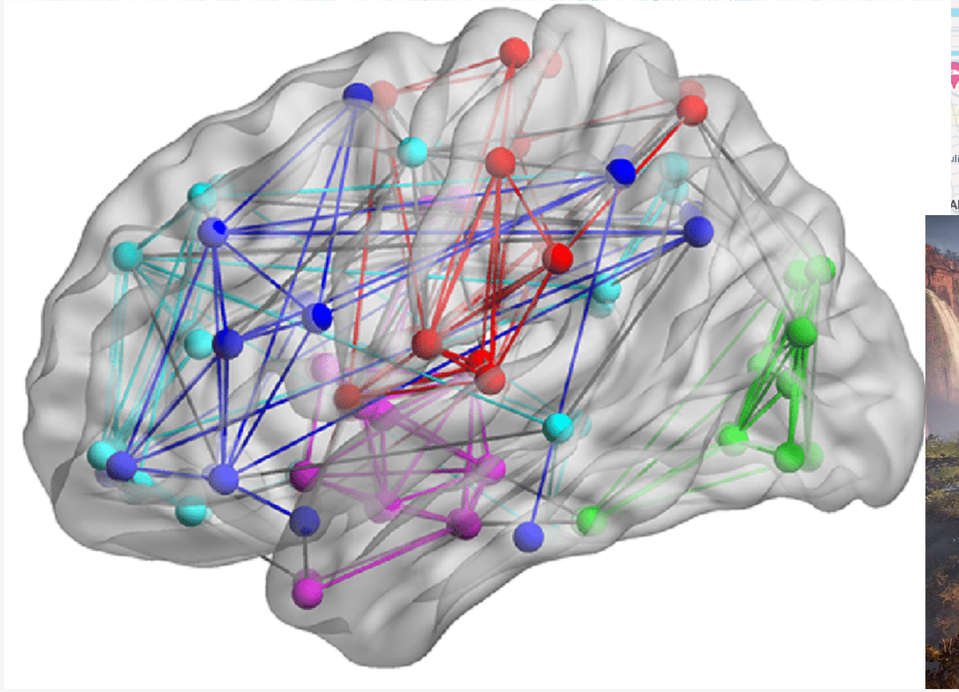
Path finding can be complex!



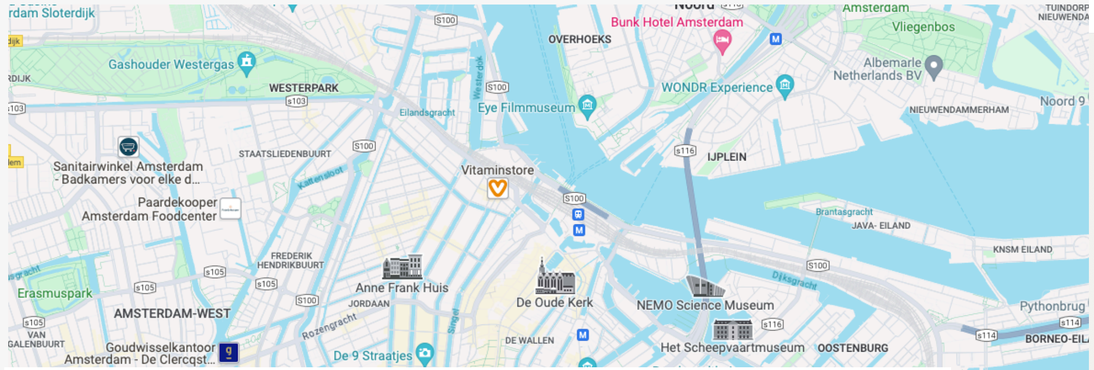


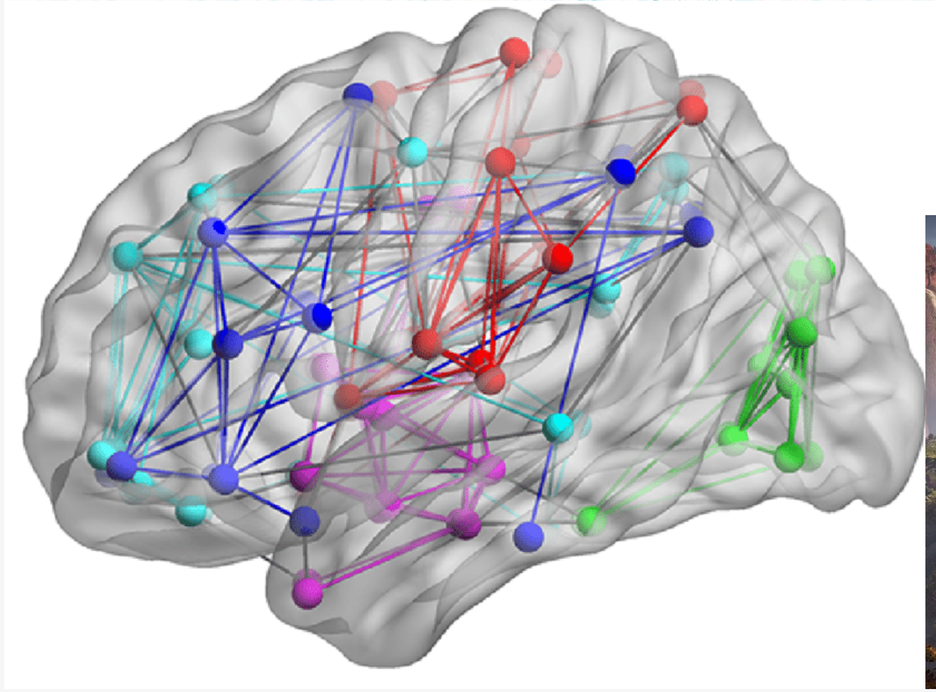
Path finding can be complex!

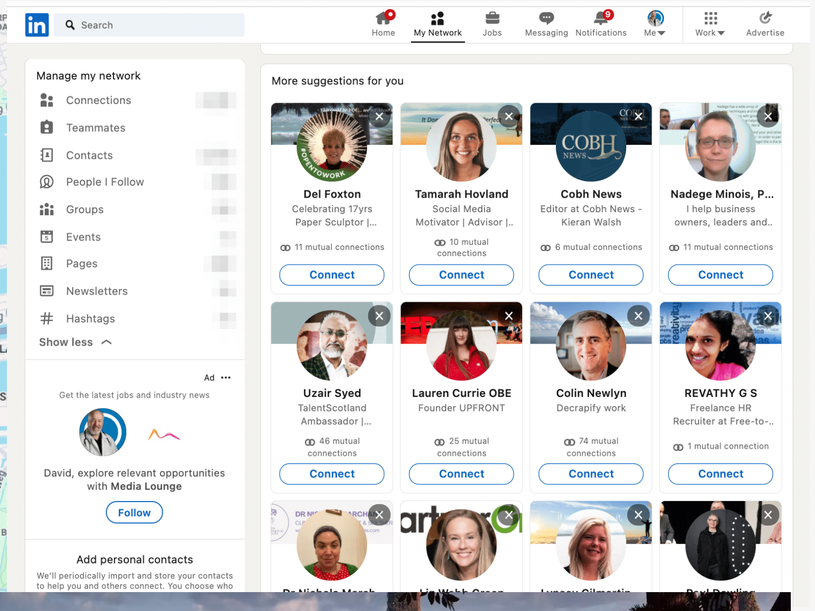


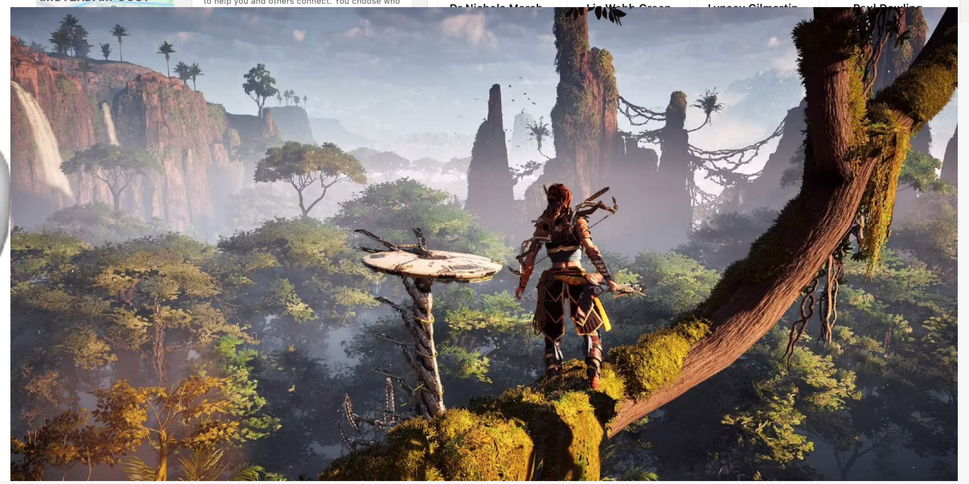


Path finding can be complex!



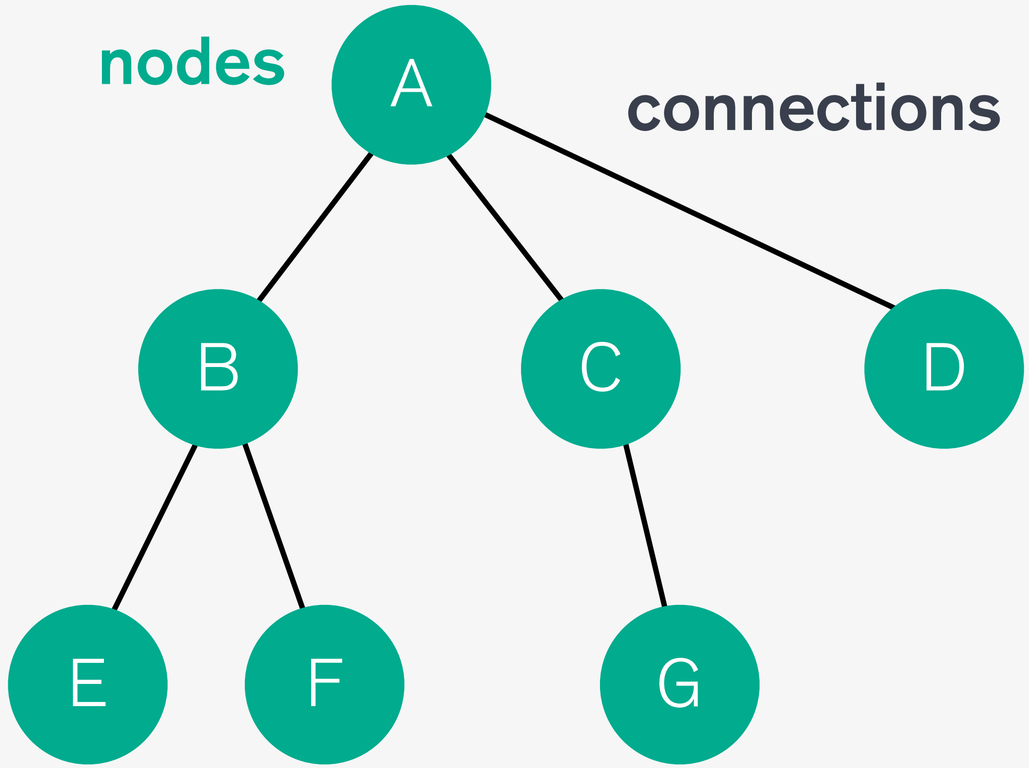




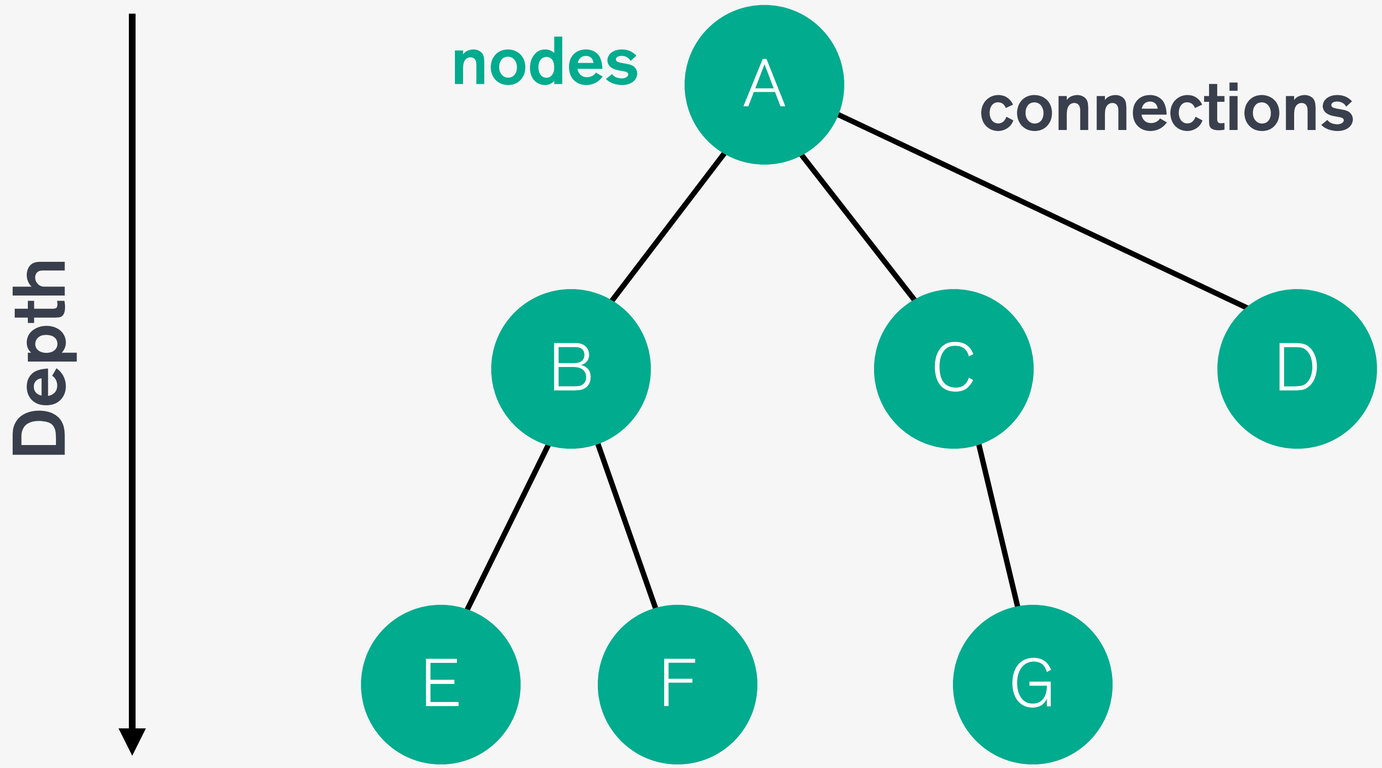


Search algorithms to the rescue!

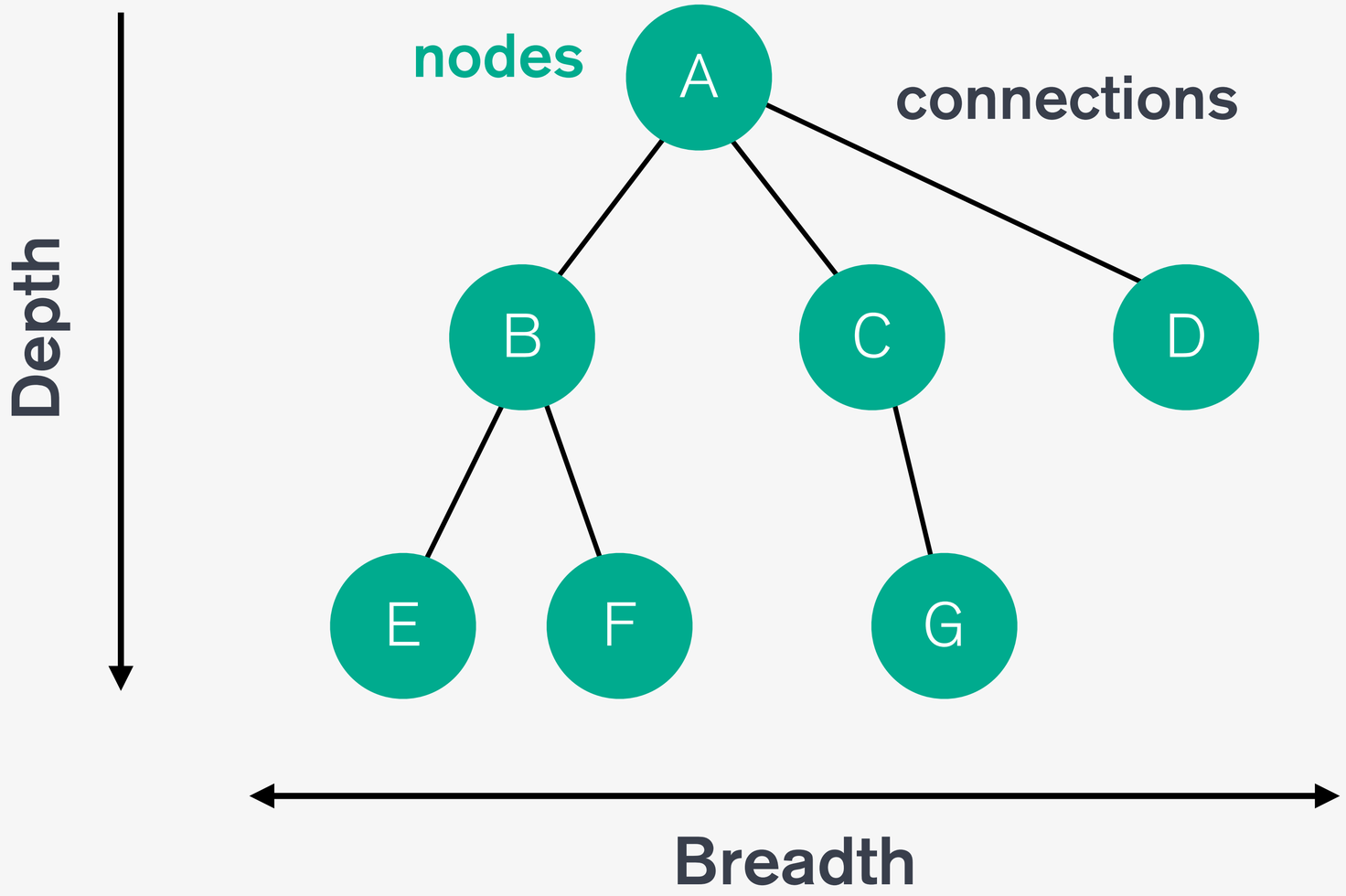
Two dimensions:



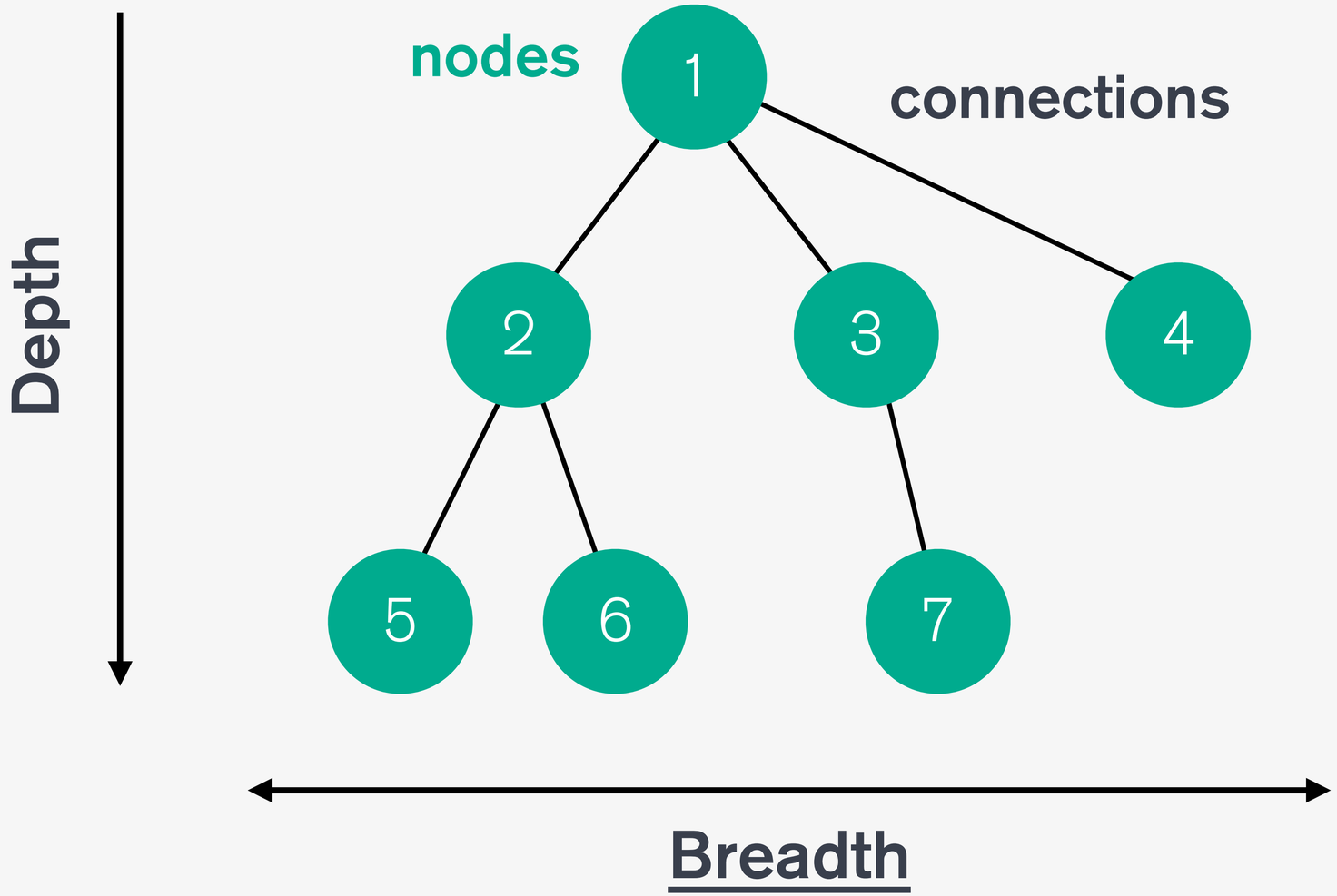
Two dimensions:



Two dimensions:

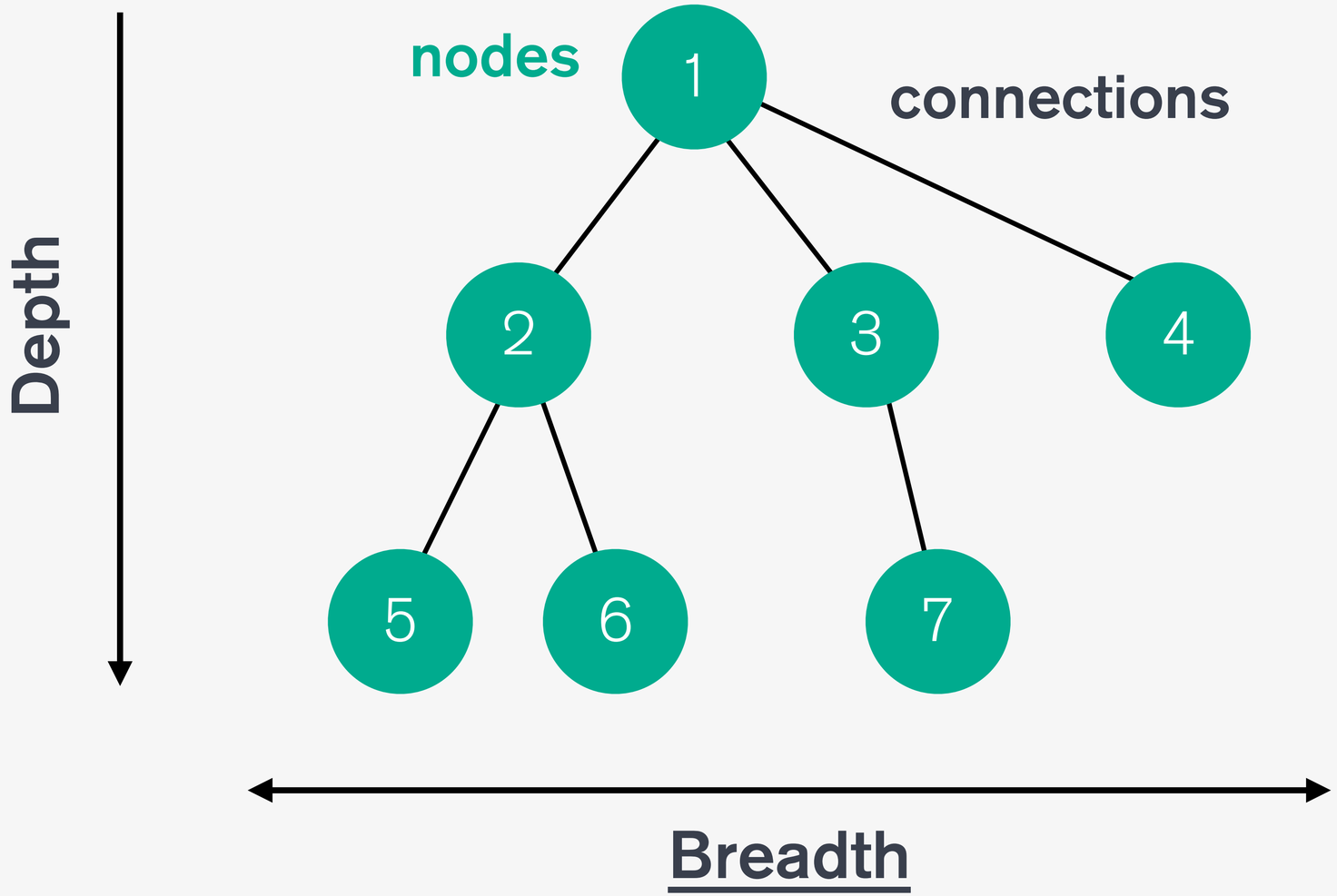


Breadth-first search

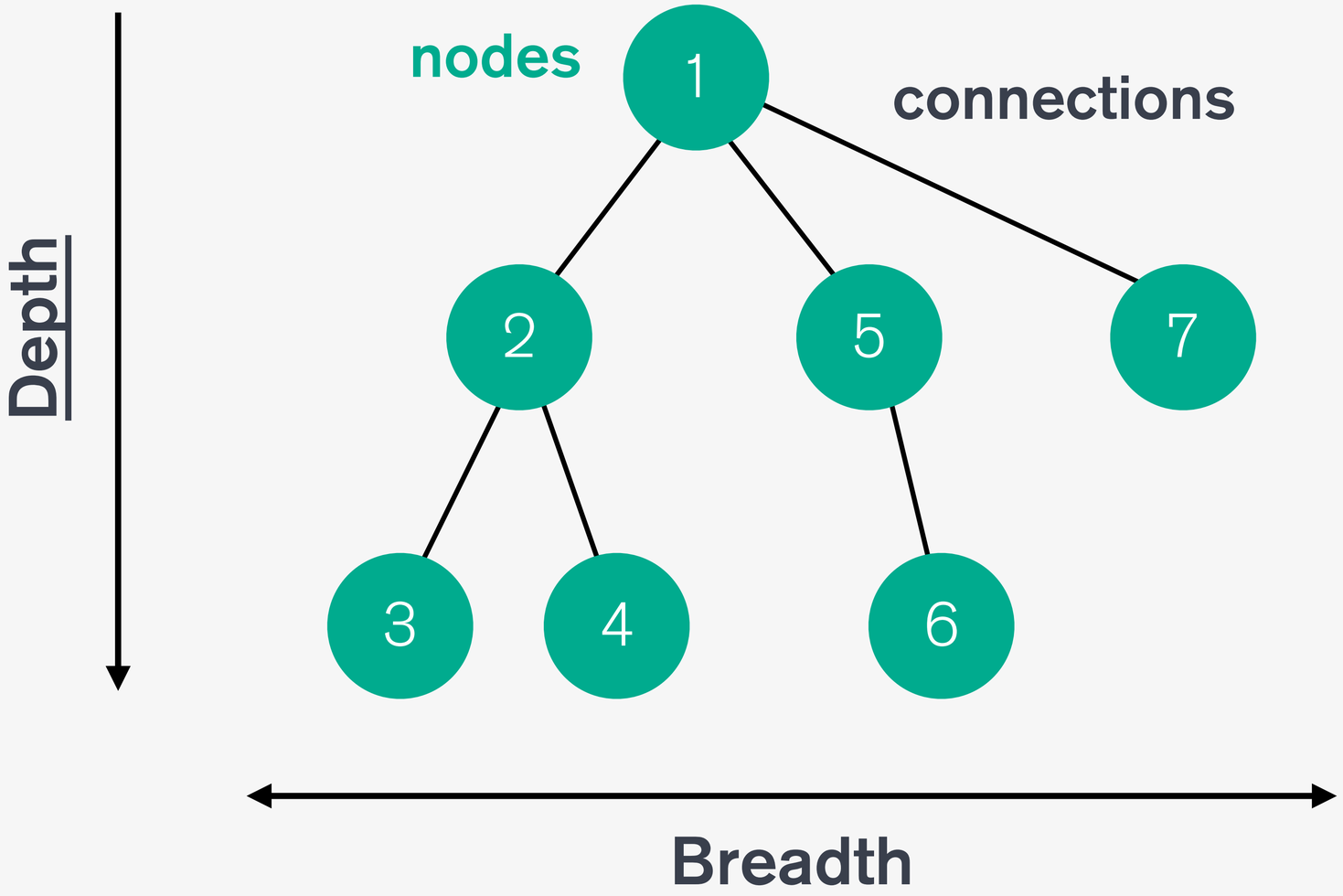


Breadth-first search

Explore all connections, before moving to the next level

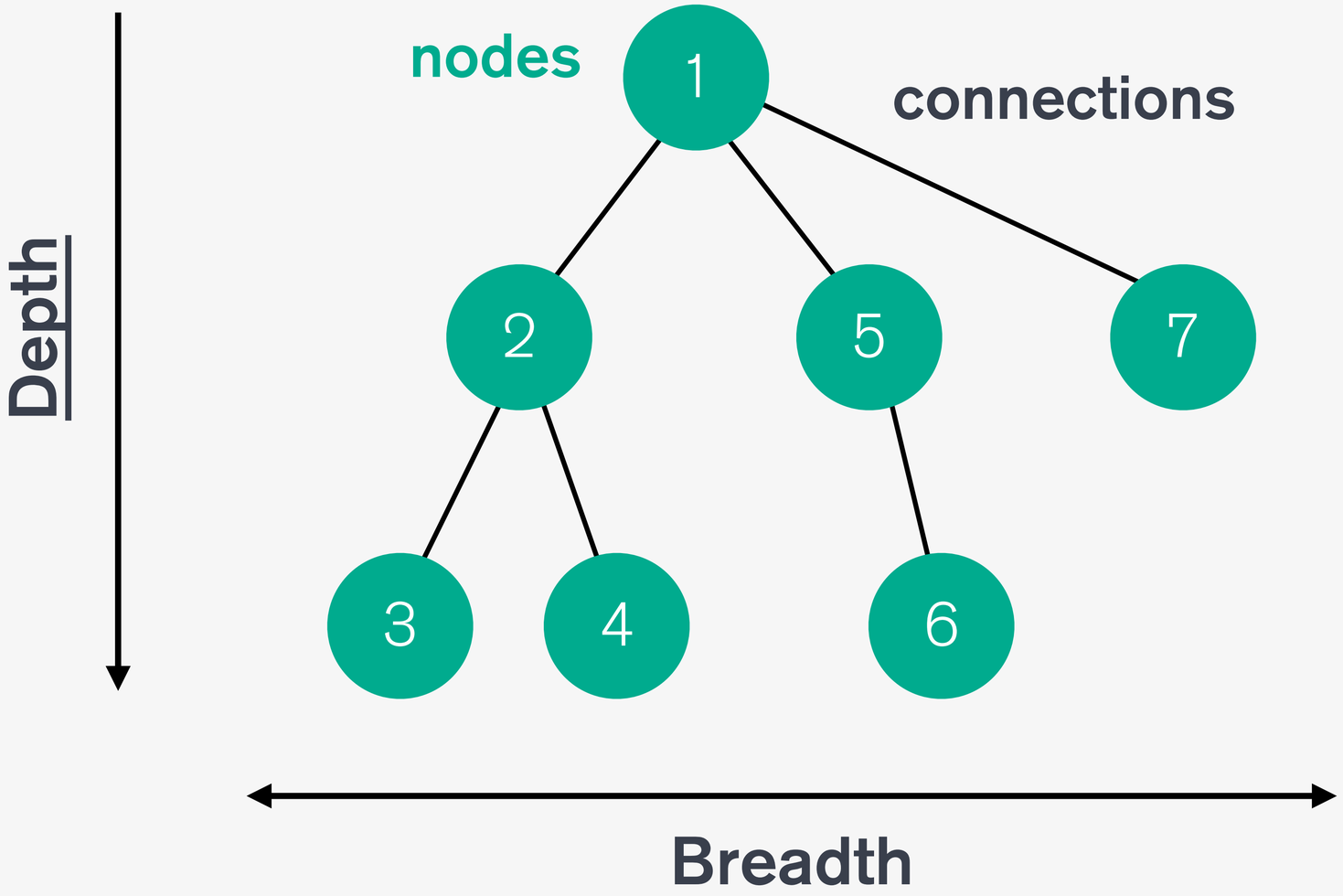


Depth-first search

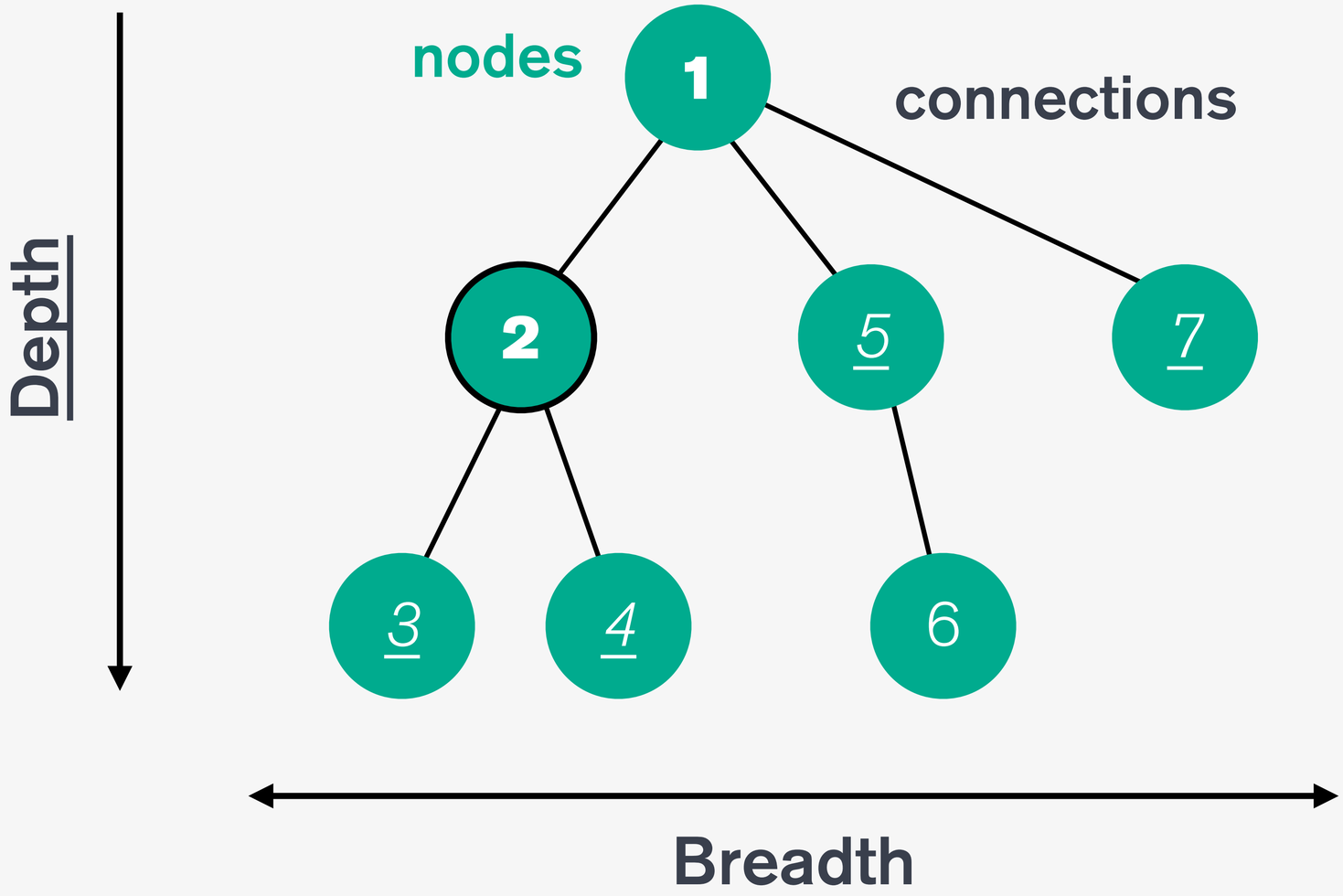


Depth-first search

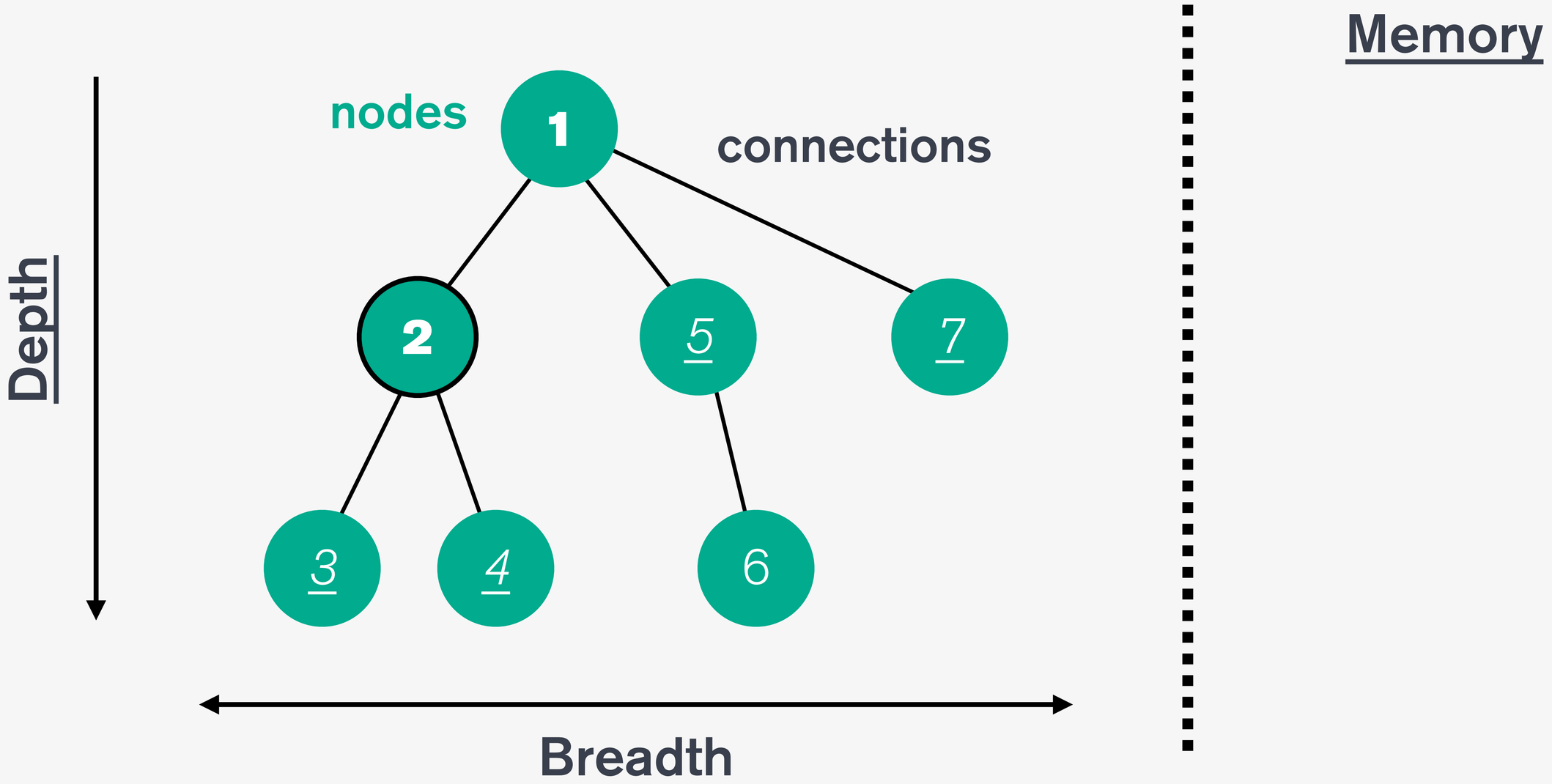
Explore as far as possible, before backtracking



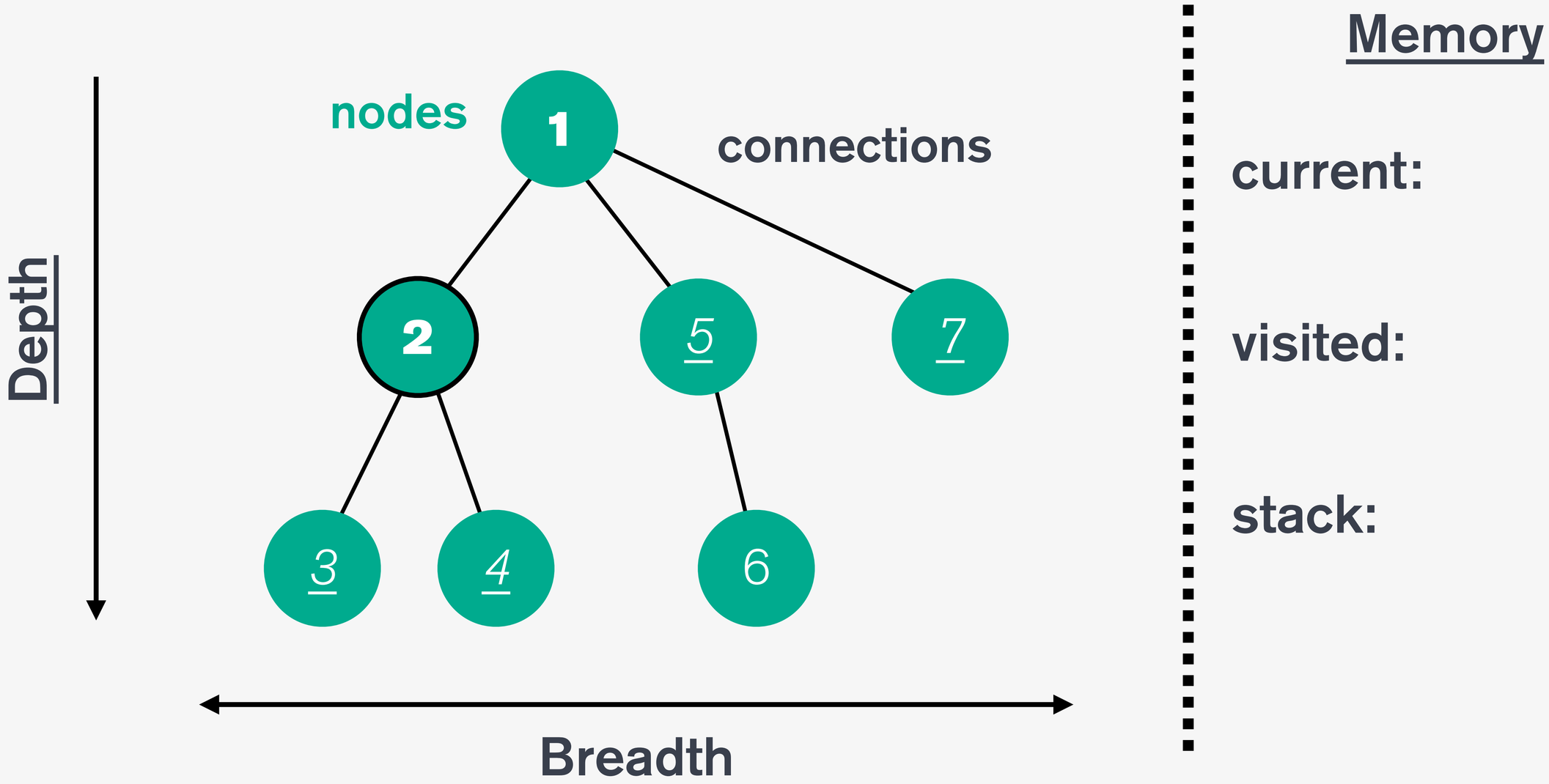
Depth-first search (DFS) in action:



Depth-first search (DFS) in action:



Depth-first search (DFS) in action:



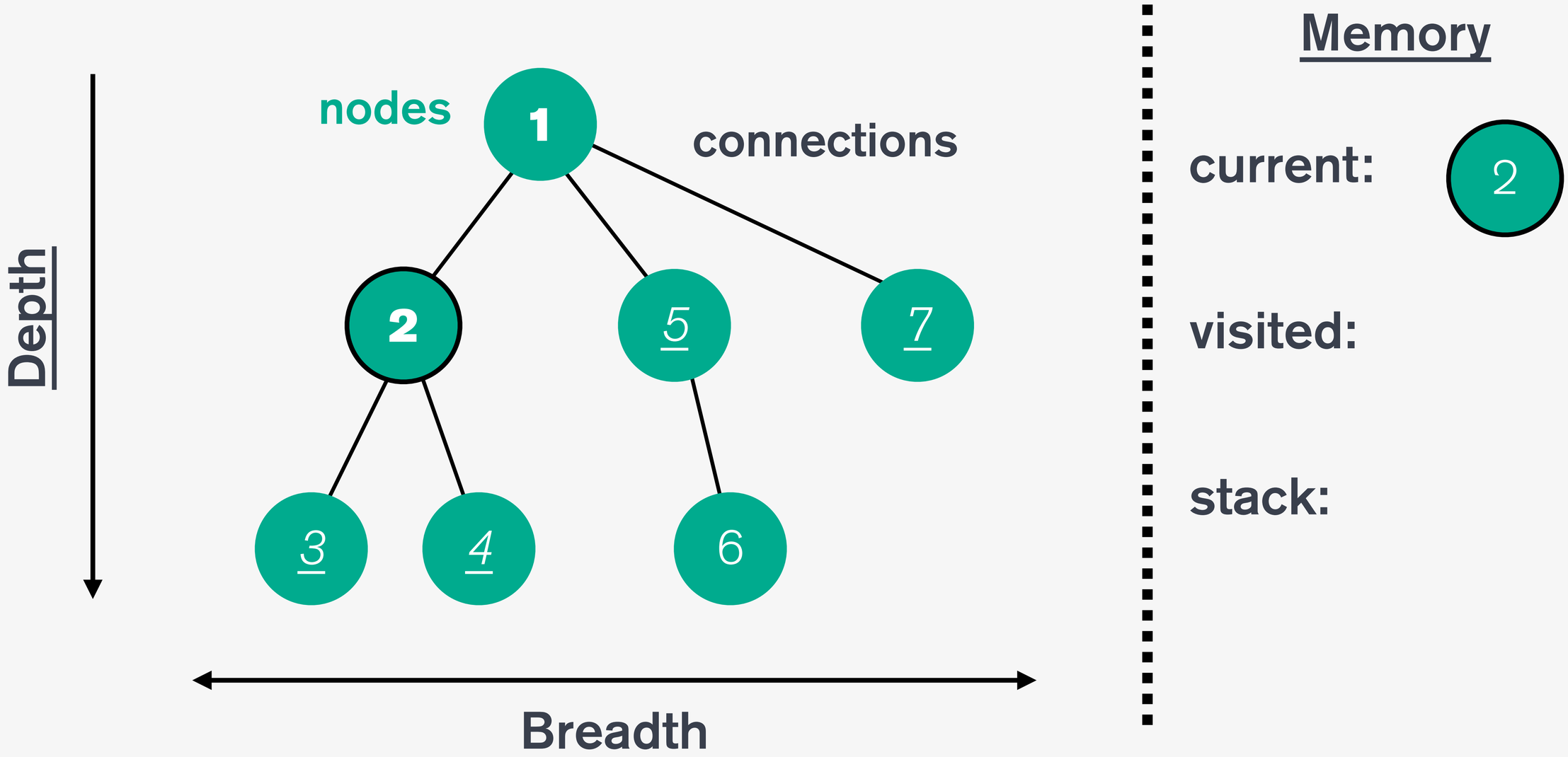
Memory

current:

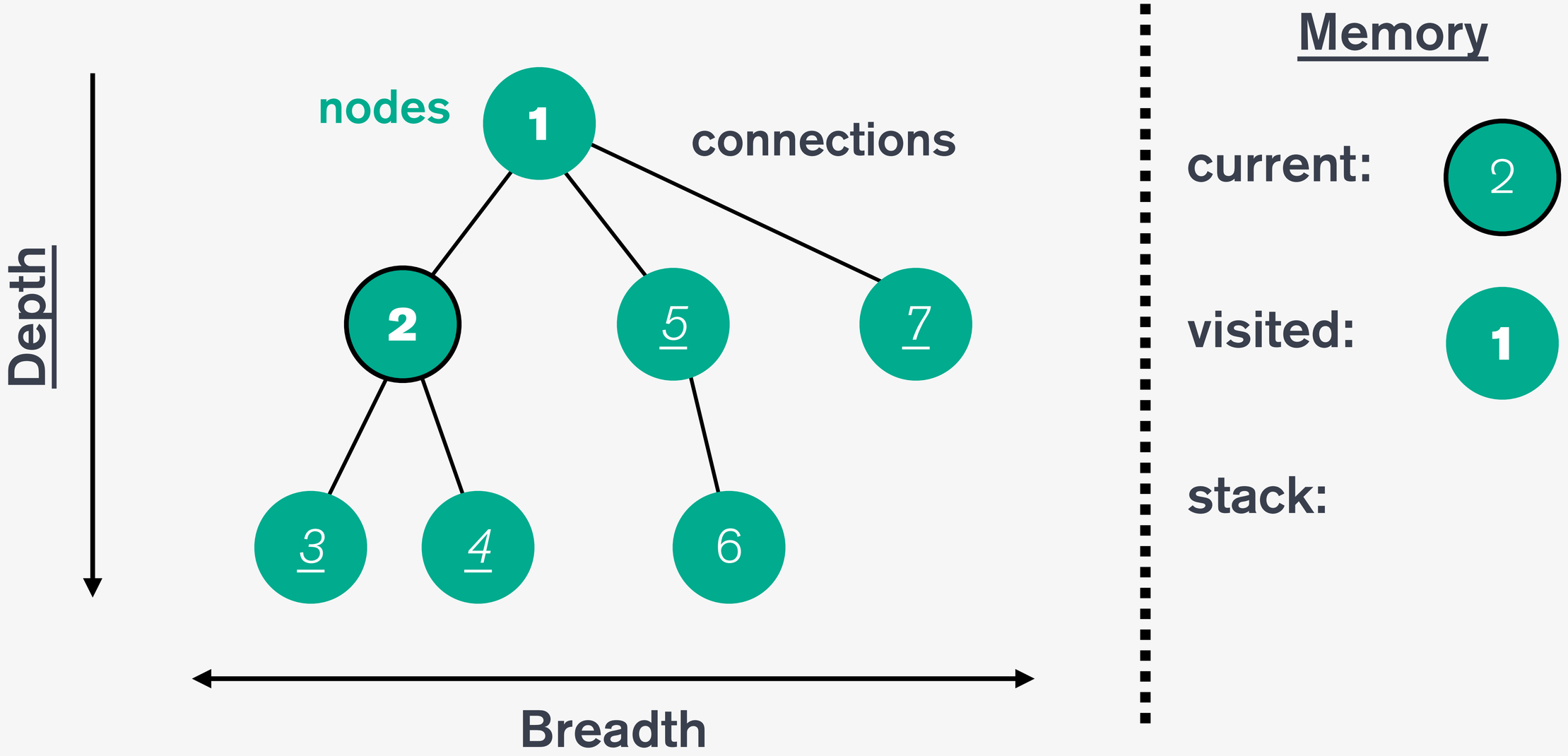
visited:

stack:

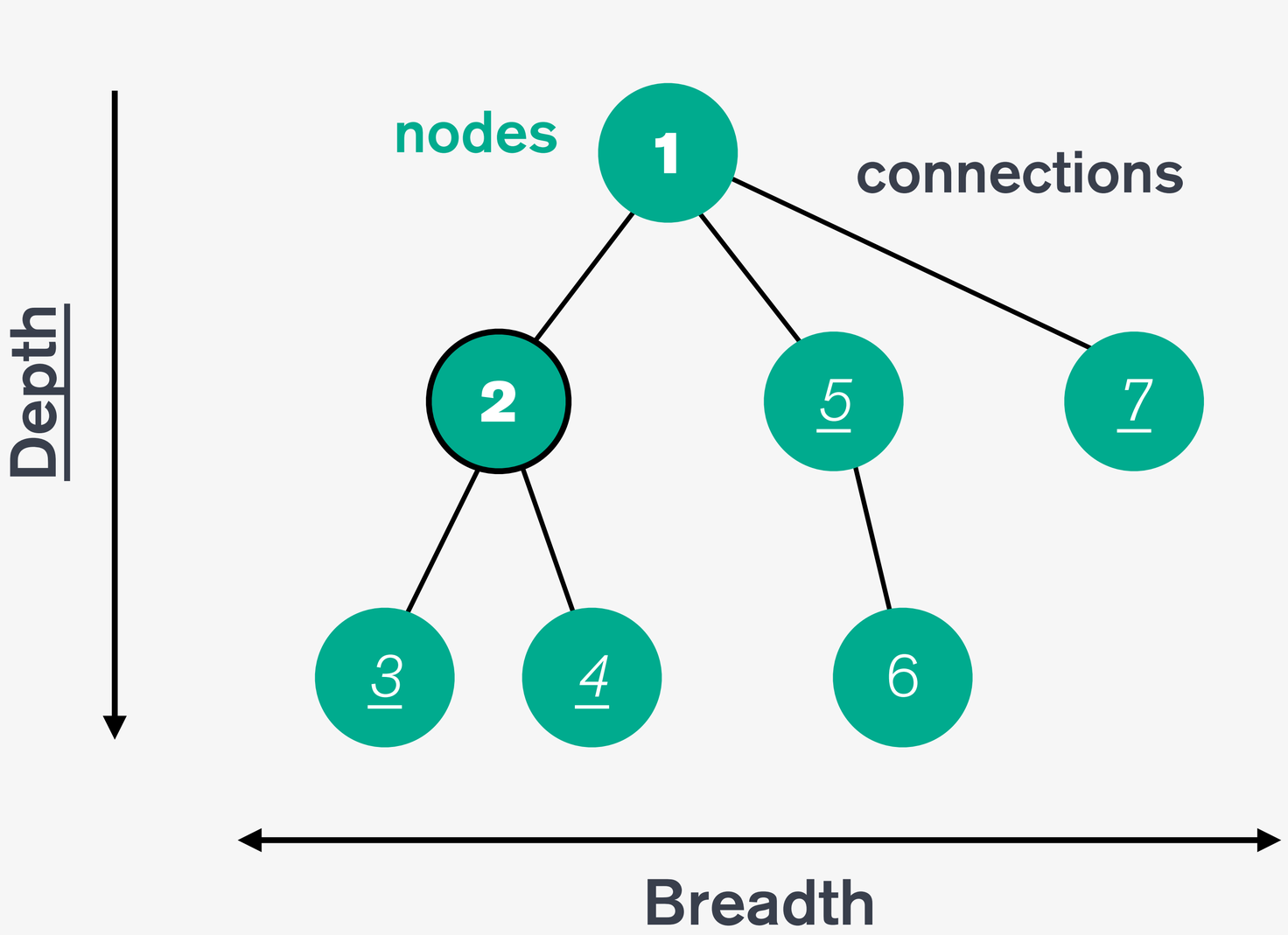
Depth-first search (DFS) in action:



Depth-first search (DFS) in action:



Depth-first search (DFS) in action:



Memory

current: 2

visited: 1

stack: 3 4 5 7

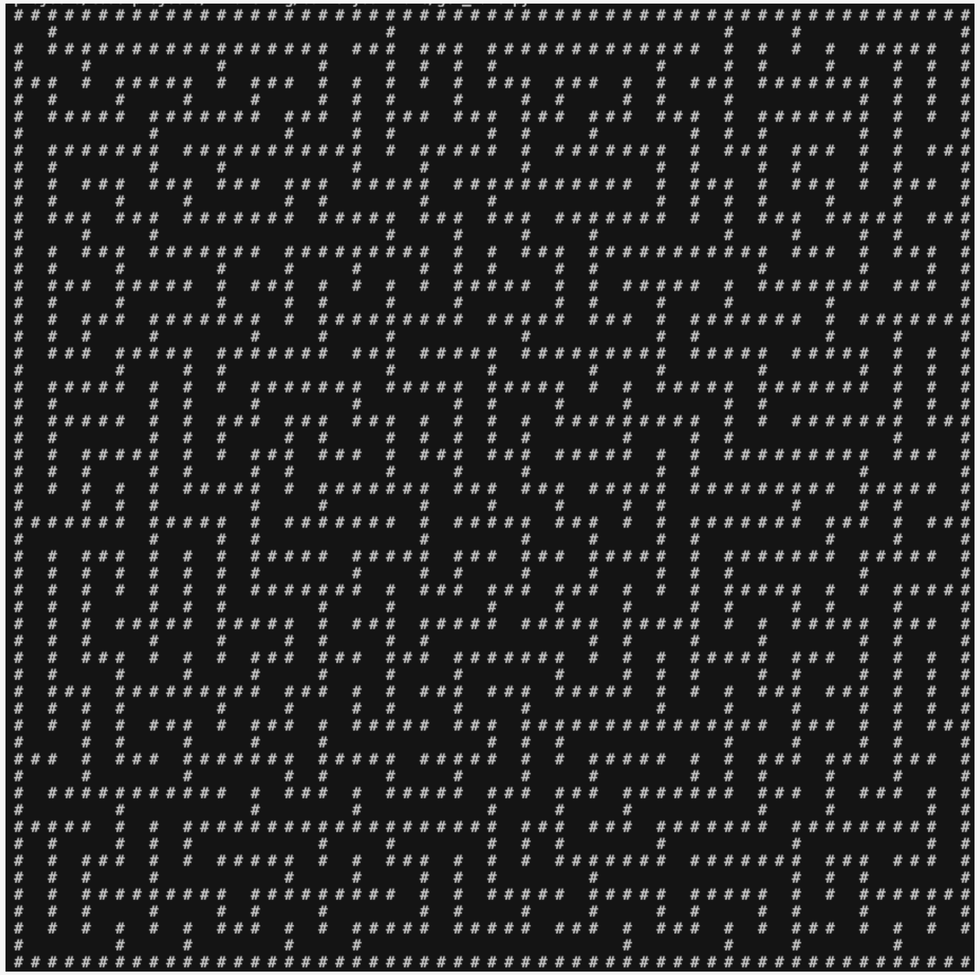
How could we program this?

DFS-pseudo.txt

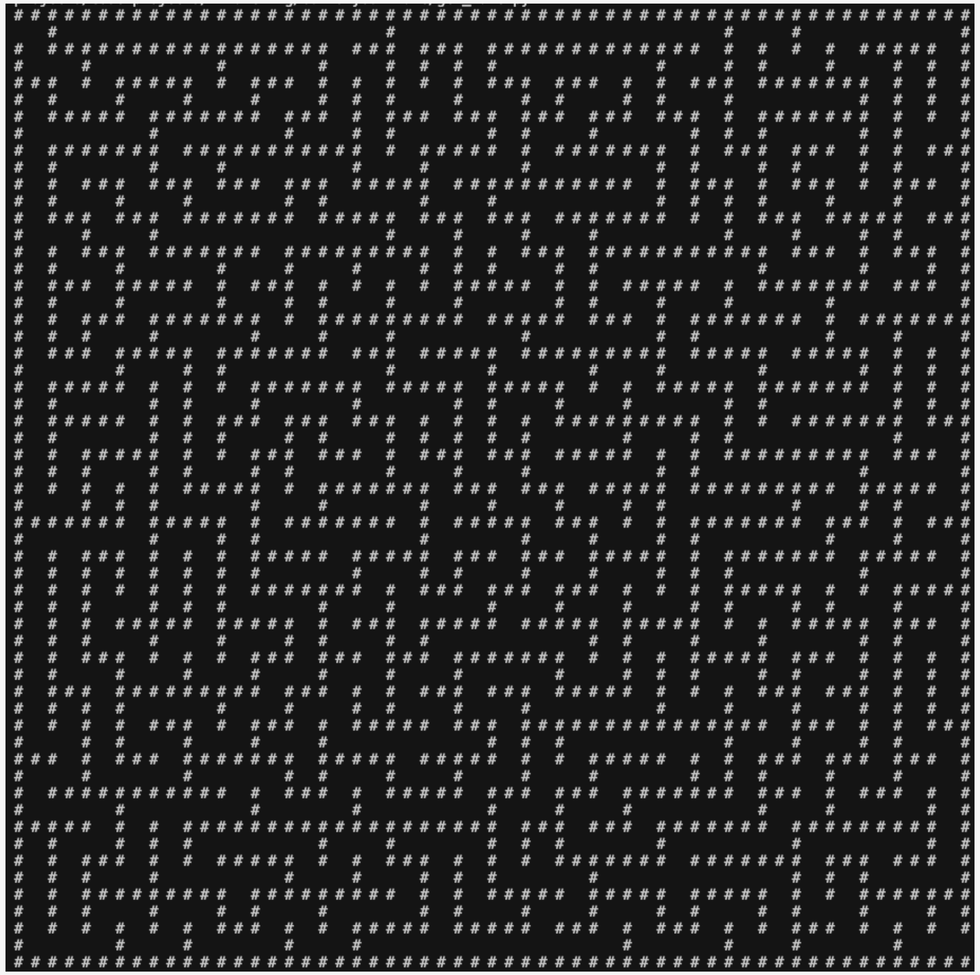
```
1   procedure DFS:
2
3       create 'stack'
4       create 'visited'
5       stack.append(start)
6
7       while stack is not empty:
8           current = stack.pop()
9
10          if current not yet visited:
11              label current as visited
12
13              for all connected to current,
14                      that are not visited:
15                  stack.append(connection)
```

Knowing this, let’s ask a slightly different question:

How would you *build* this maze?



Let’s check out the *solution* together!



A-maze-ing, right?