**Breadth-first search** (**BFS**) is an [algorithm](https://en.wikipedia.org/wiki/Algorithm) for searching a [tree](https://en.wikipedia.org/wiki/Tree_(data_structure)) data structure for a node that satisfies a given property. It starts at the [tree root](https://en.wikipedia.org/wiki/Tree_(data_structure)#Terminology) and explores all nodes at the present [depth](https://en.wikipedia.org/wiki/Tree_(data_structure)#Terminology) prior to moving on to the nodes at the next depth level. Extra memory, usually a [queue](https://en.wikipedia.org/wiki/Queue_(data_structure)), is needed to keep track of the child nodes that were encountered but not yet explored.

For example, in a [chess endgame](https://en.wikipedia.org/wiki/Chess_endgame) a [chess engine](https://en.wikipedia.org/wiki/Chess_engine) may build the [game tree](https://en.wikipedia.org/wiki/Game_tree) from the current position by applying all possible moves, and use breadth-first search to find a win position for white. Implicit trees (such as game trees or other problem-solving trees) may be of infinite size; breadth-first search is guaranteed to find a solution node[[1]](https://en.wikipedia.org/wiki/Breadth-first_search#cite_note-1) if one exists.

In contrast, (plain) [depth-first search](https://en.wikipedia.org/wiki/Depth-first_search), which explores the node branch as far as possible before backtracking and expanding other nodes,[[2]](https://en.wikipedia.org/wiki/Breadth-first_search#cite_note-2) may get lost in an infinite branch and never make it to the solution node. [Iterative deepening depth-first search](https://en.wikipedia.org/wiki/Iterative_deepening_depth-first_search) avoids the latter drawback at the price of exploring the tree's top parts over and over again. On the other hand, both depth-first algorithms get along without extra memory.

# FURTHER Queue (abstract data type)

# Routing (electronic design automation)

# Depth-first search

Maze Search: <https://www.youtube.com/watch?v=rop0W4QDOUI>

Dijkstra: <https://www.youtube.com/watch?v=GazC3A4OQTE>

A\* Search: https://www.youtube.com/watch?v=ySN5Wnu88nE