Breadth-first search

- Start at a node in the graph.
- Explore all nodes adjacent to the current one.
- Move on to the next closest unvisited node.

Example:

- Initial node is A.
- Explore all nodes adjacent to A: B, C, D.
- Move on to the closest unvisited node: B.
- Explore all nodes adjacent to B: E, F.
- Move on to the closest unvisited node: E.
- Explore all nodes adjacent to E: G, H.
- Move on to the closest unvisited node: G.
- Explore all nodes adjacent to G: I, J.
- Move on to the closest unvisited node: I.
- Explore all nodes adjacent to I: K, L.
- Move on to the closest unvisited node: J.
- Explore all nodes adjacent to J: M, N.
- Move on to the closest unvisited node: M.
- Explore all nodes adjacent to M: O, P.
- Move on to the closest unvisited node: O.

- The breadth-first search tree is:
  - A → B → E → G → I → K → M → O
  - A → B → C → D → H → J → L → N
  - A → B → C → D → G → I → K → M

- Time complexity: \( O(n + m) \)
- Space complexity: \( O(n) \)