

Depth-first Search

Depth-first search treats the frontier as a stack: it always selects the last element added to the frontier.

select(Node, [Node|Frontier], Frontier).

add_to_frontier(Neighbors, Frontier₁, Frontier₂) ←

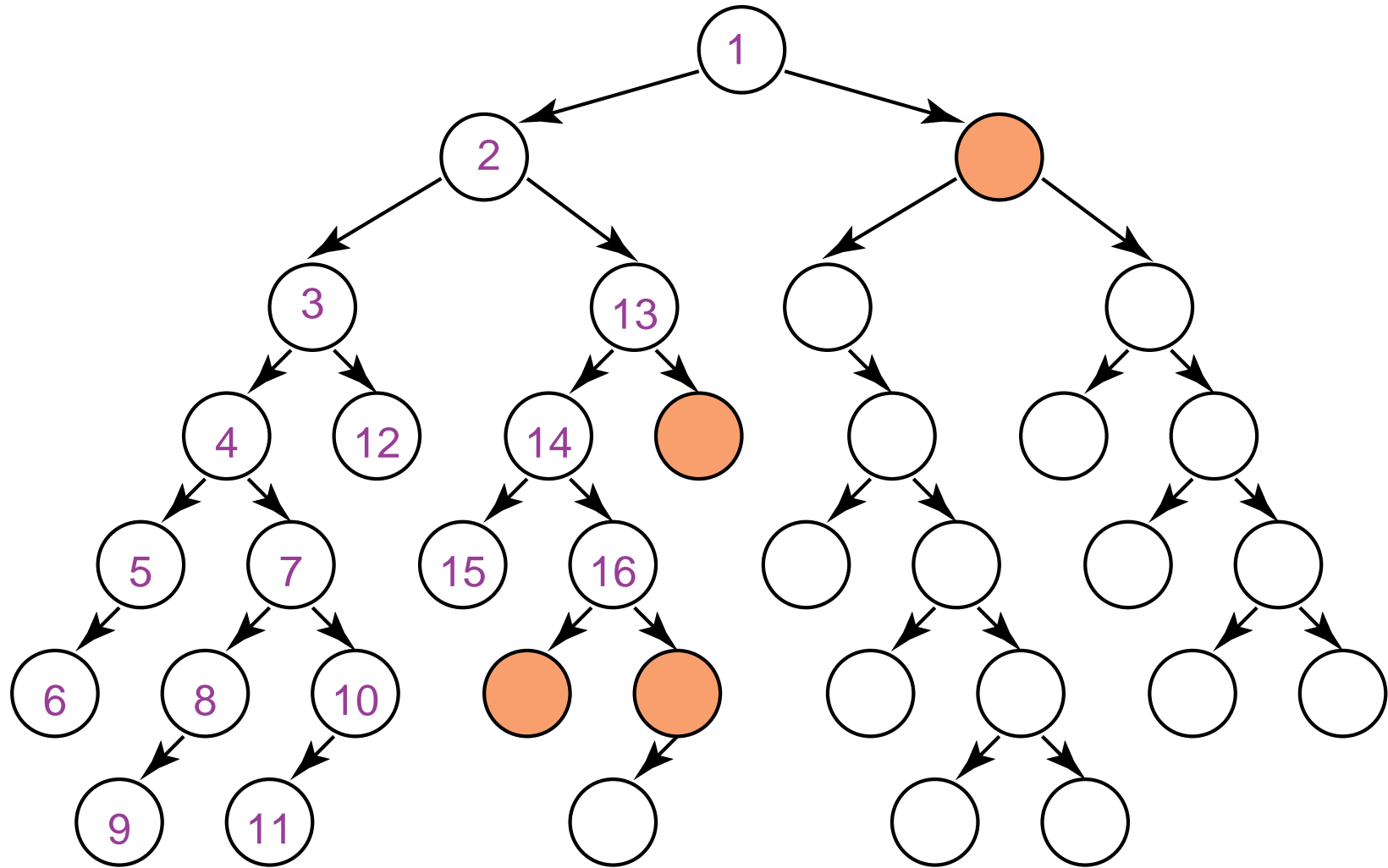
append(Neighbors, Frontier₁, Frontier₂).

Frontier: [e_1, e_2, \dots]

e_1 is selected. Its neighbors are added to the front of the stack.

e_2 is only selected when all paths from e_1 have been explored.

Illustrative Graph — Depth-first Search



Complexity of Depth-first Search

- Depth-first search isn't guaranteed to halt on infinite graphs or graphs with cycles.
- The space complexity is linear in the size of the path being explored.
- Search is unconstrained by the goal until it happens to stumble on the goal.



Breadth-first Search

Breadth-first search treats the frontier as a queue: it always selects the earliest element added to the frontier.

select(Node, [Node|Frontier], Frontier).

add_to_frontier(Neighbors, Frontier₁, Frontier₂) ←

append(Frontier₁, Neighbors, Frontier₂).

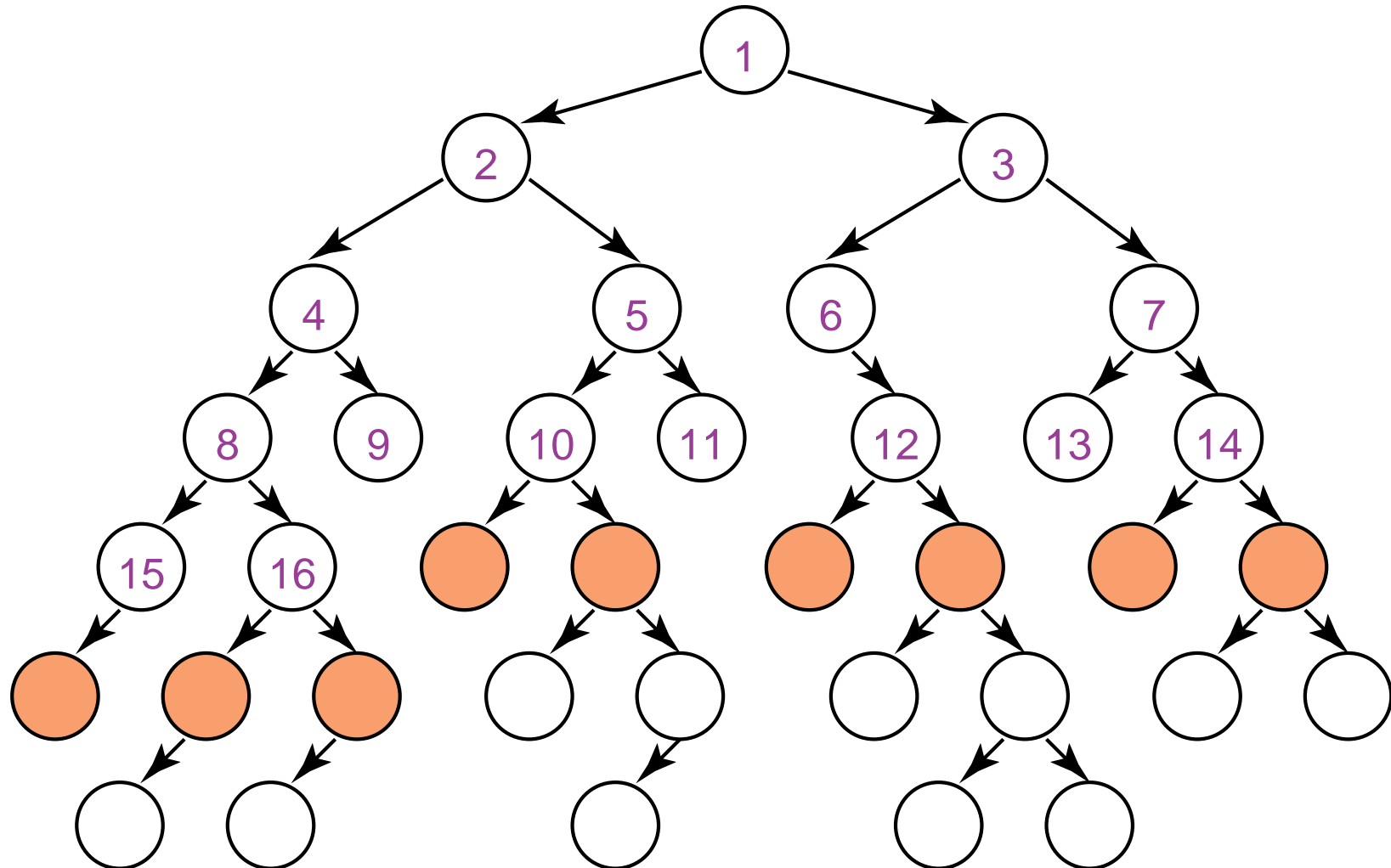
Frontier: [e_1, e_2, \dots]

e_1 is selected. Its neighbors are added to the end of the queue.

e_2 is selected next.



Illustrative Graph — Breadth-first Search



Complexity of Breadth-first Search

- The **branching factor** of a node is the number of its neighbors.
- If the branching factor for all nodes is finite, breadth-first search is guaranteed to find a solution if one exists. It is guaranteed to find the path with fewest arcs.
- Time complexity is exponential in the path length: b^n , where b is branching factor, n is path length.
- The space complexity is exponential in path length: b^n .
- Search is unconstrained by the goal.



Lowest-cost-first Search

- Sometimes there are **costs** associated with arcs. The cost of a path is the sum of the costs of its arcs.
- Lowest-cost-first search finds the shortest path to a goal node.
- At each stage, it selects the shortest path on the frontier.
- The frontier is implemented as a priority queue ordered by path length.
- When arc costs are equal \implies breadth-first search.

