Backtracking

(Algorithmic Problems)

Backtracking Algorithms

- **backtracking** is form of recursion
- general algorithm for finding all solutions to some computational problems
- these are called **constraint satisfaction problems**
- backtracking is also important when solving **combinatorial optimization problems** (travelling salesman problem etc.)
- it is often much faster than brute force enumeration of all complete candidates - because it can eliminate a large number of candidates with a single test
- **N-queens problem** or **Sudoku**

Backtracking Algorithms

- **brute-force approach:** we consider and evaluate all the possible solutions (or states)
- **backtracking:** we can discard several bad states with one iteration
- if partial candidate **A** cannot be completed to a valid solution then we abandon **A** as a solution
- we can represent most of these problems with a **tree structure** – it is called *game tree* or *potential search tree*

Backtracking Algorithms

- each partial candidate is the parent of the candidates that differ from it by a single extension step
- leaves of the tree are the partial candidates that cannot be extended any further
- the **backtracking** algorithm traverses this search tree recursively, from the root down – like *depth-first search*

BACKTRACKING IS CALLED DEPTH-FIRST SEARCH IF APPLIED ON TREES

Backtracking Algorithms

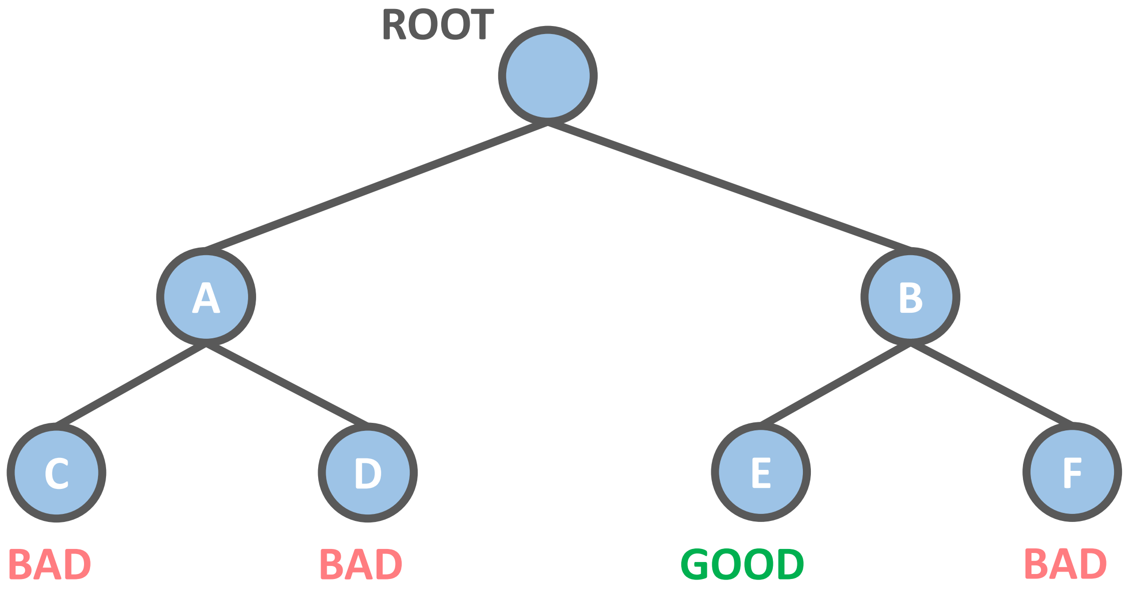
Backtracking is also called **depth-first search** (and vice versa)

1.) for every node the algorithm checks whether the given node can be completed to a valid solution

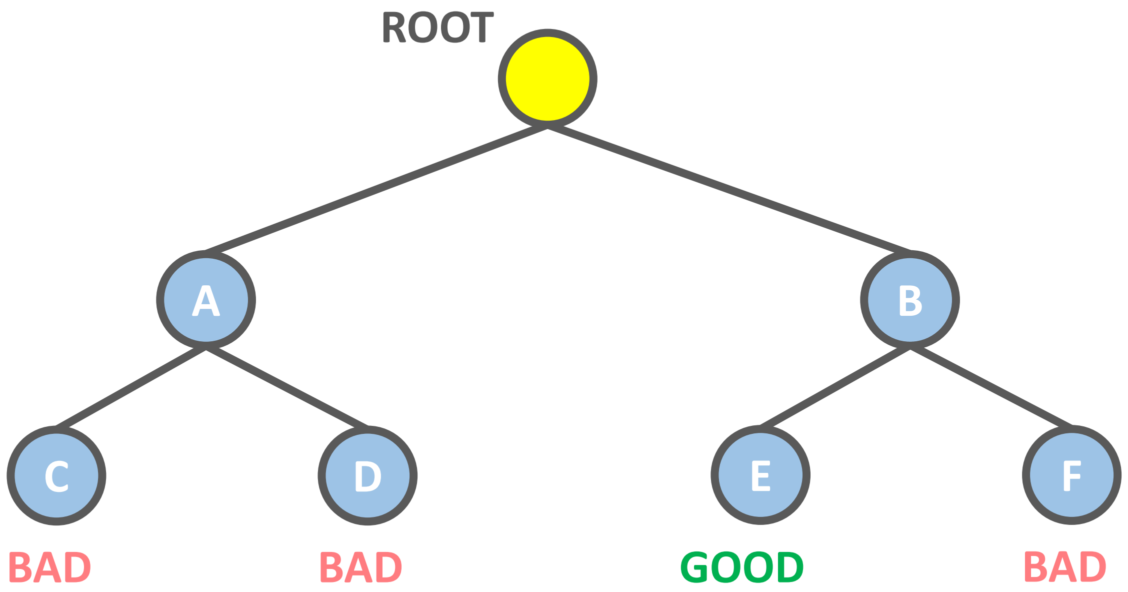
2.) if it can not then the whole subtree is skipped (this is the key advantage of backtracking)

3.) it recursively enumerates all subtree of the node

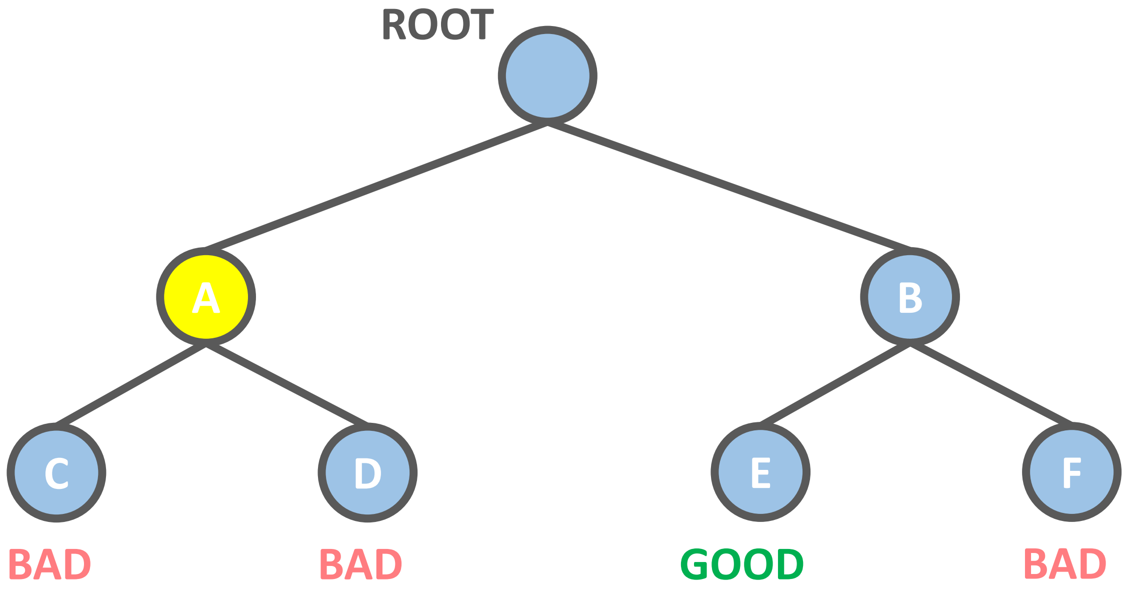
Backtracking Algorithms



Backtracking Algorithms

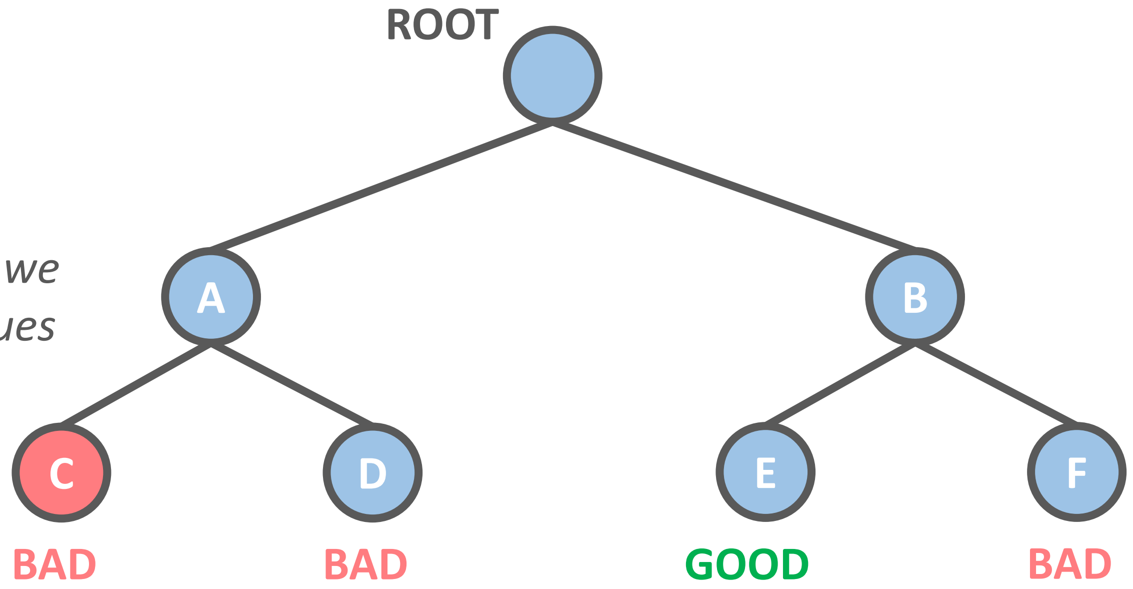


Backtracking Algorithms

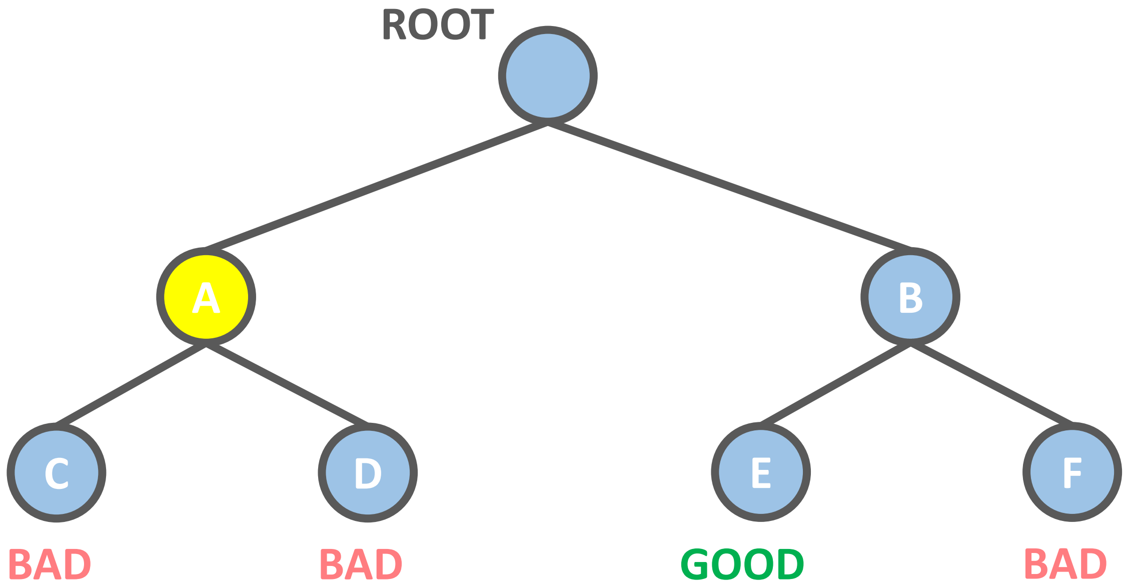


Backtracking Algorithms

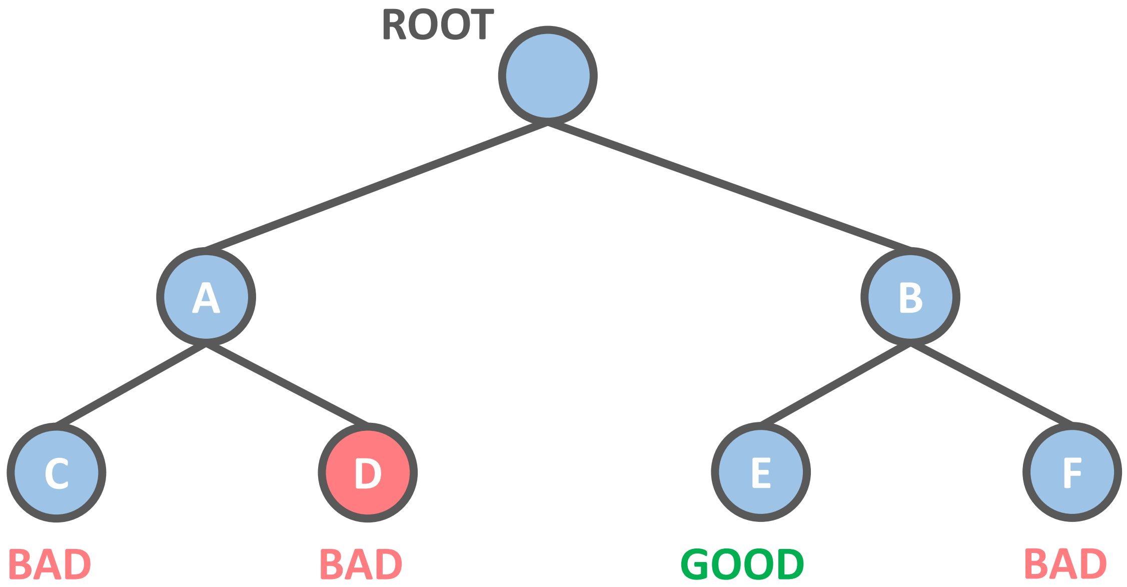
this is when we have to ***backtrack*** *which means that we have to update previous values (in previous iterations)*



Backtracking Algorithms

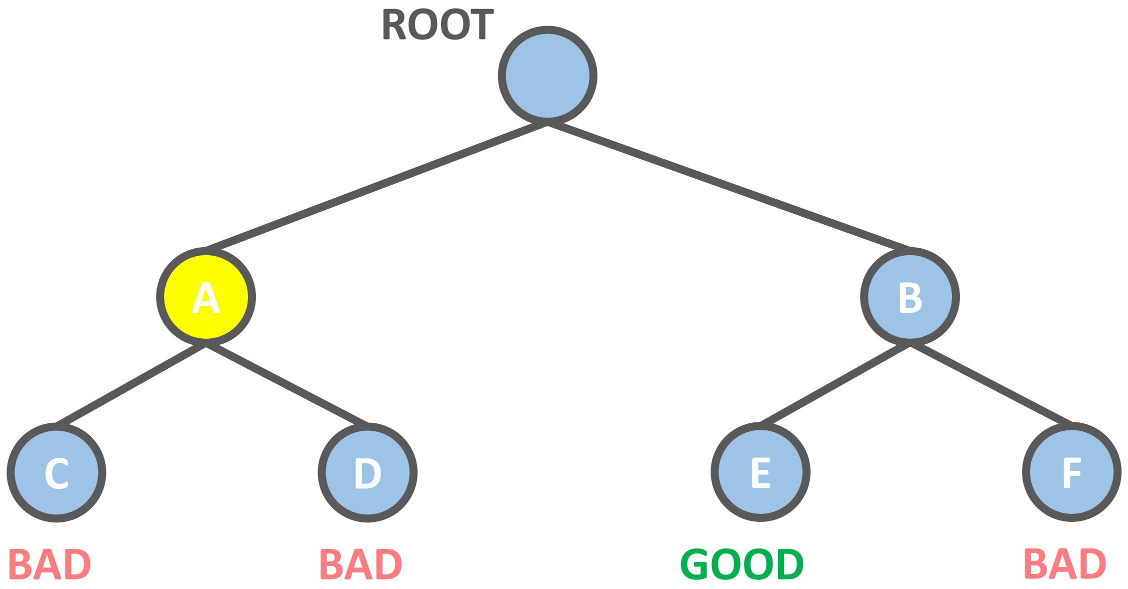


Backtracking Algorithms

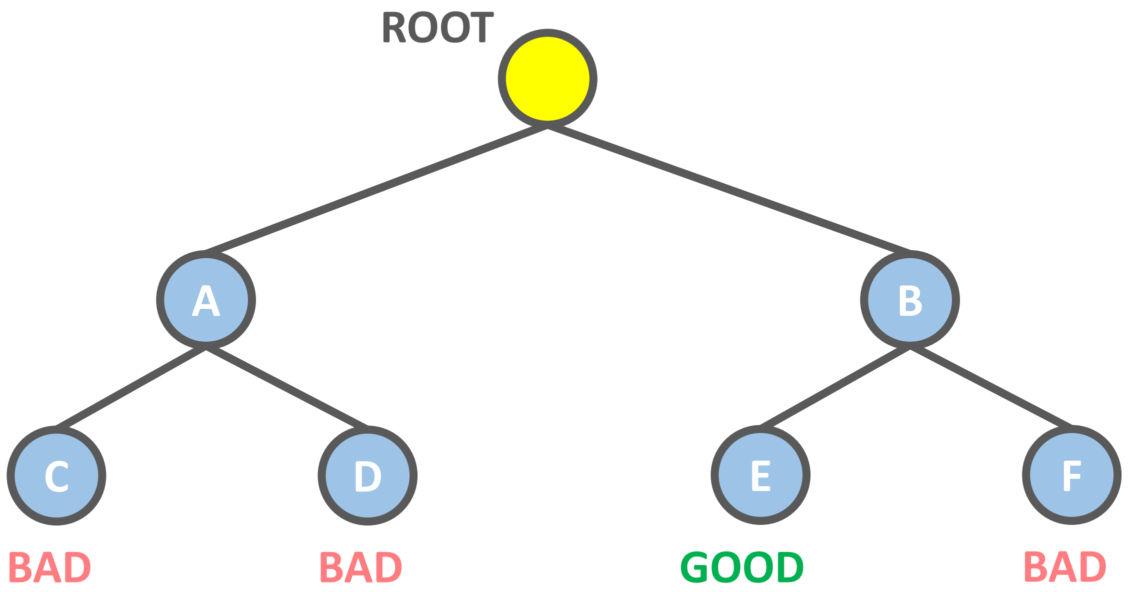


*this is when we have to **backtrack** which means that we have to update previous values (in previous iterations)*

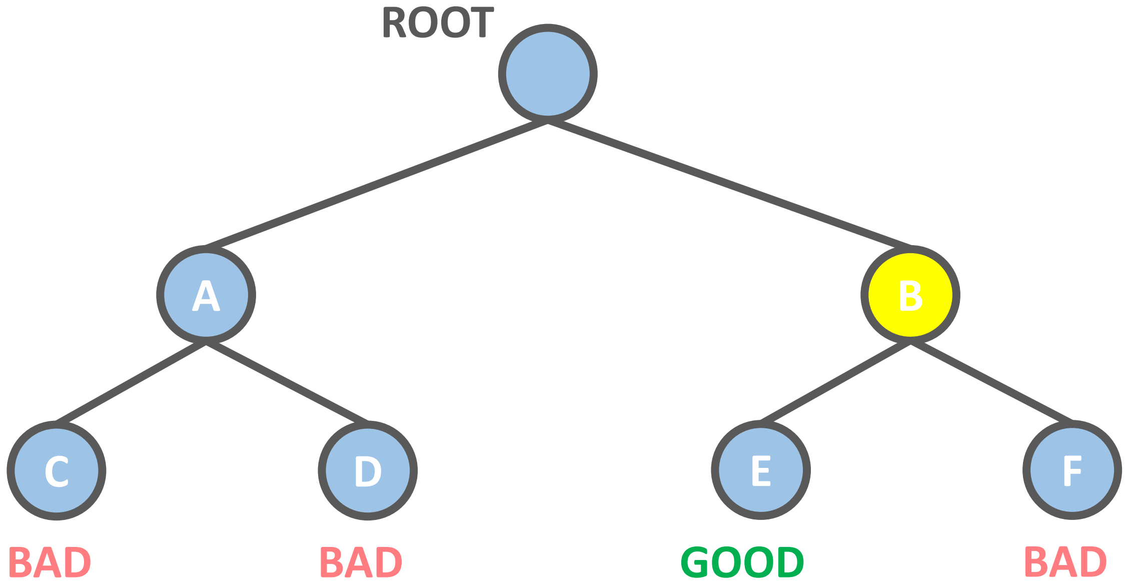
Backtracking Algorithms



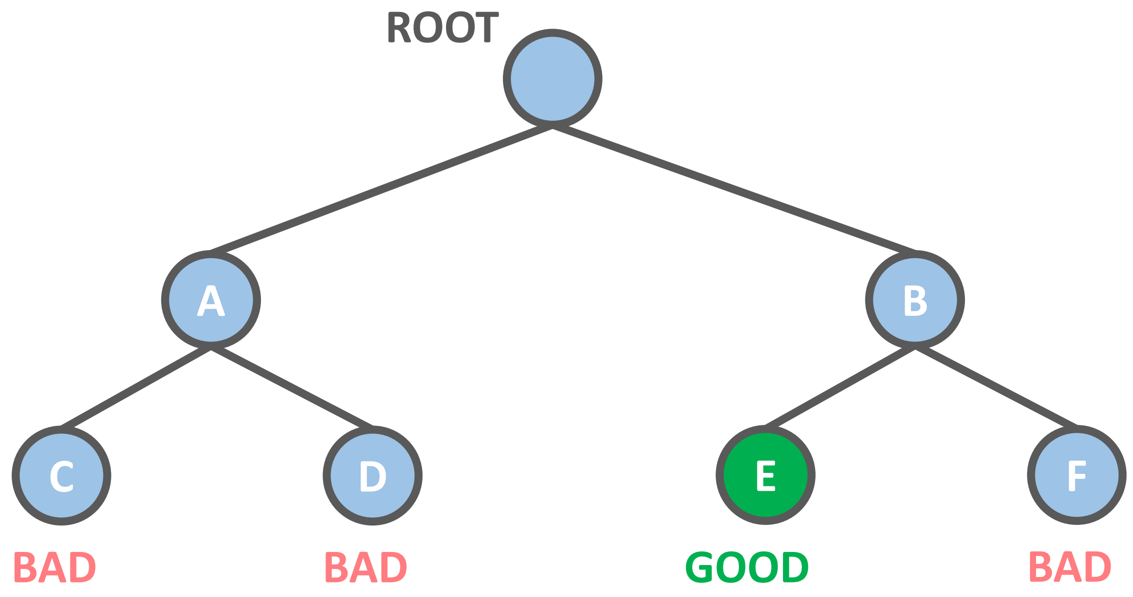
Backtracking Algorithms



Backtracking Algorithms



Backtracking Algorithms

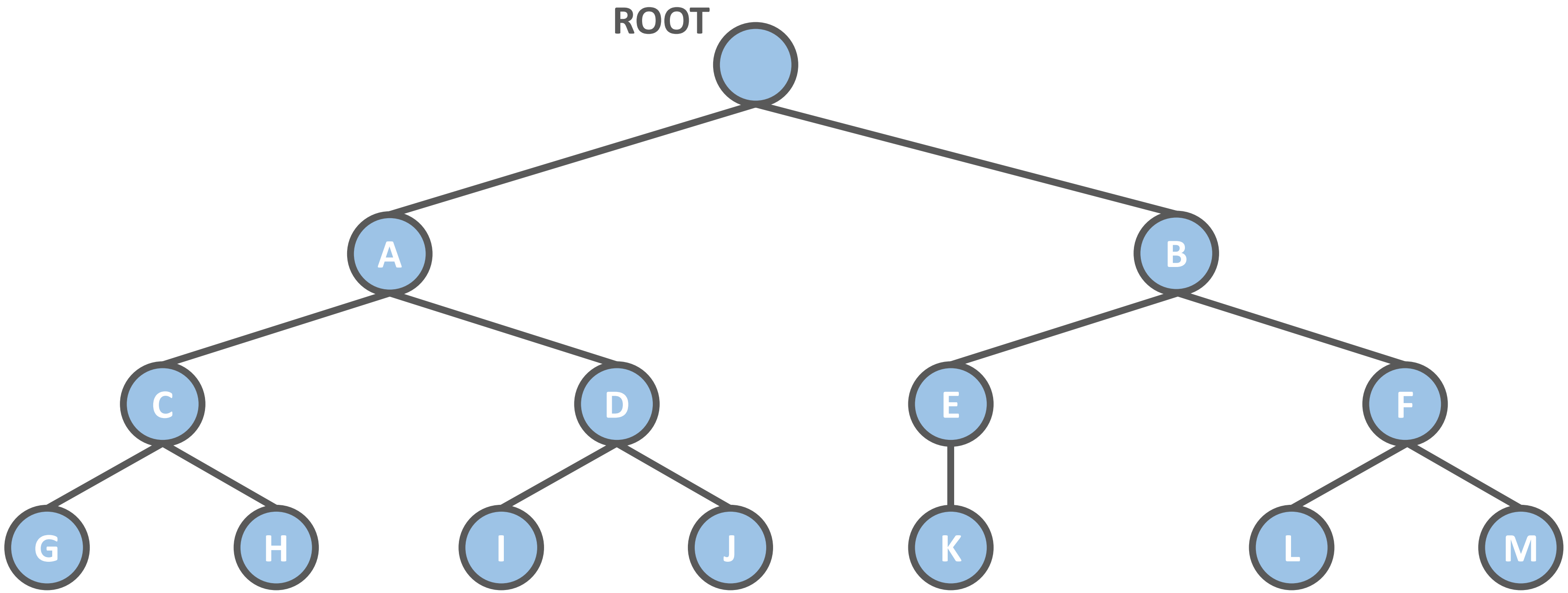


found the solution
we are looking for !!!

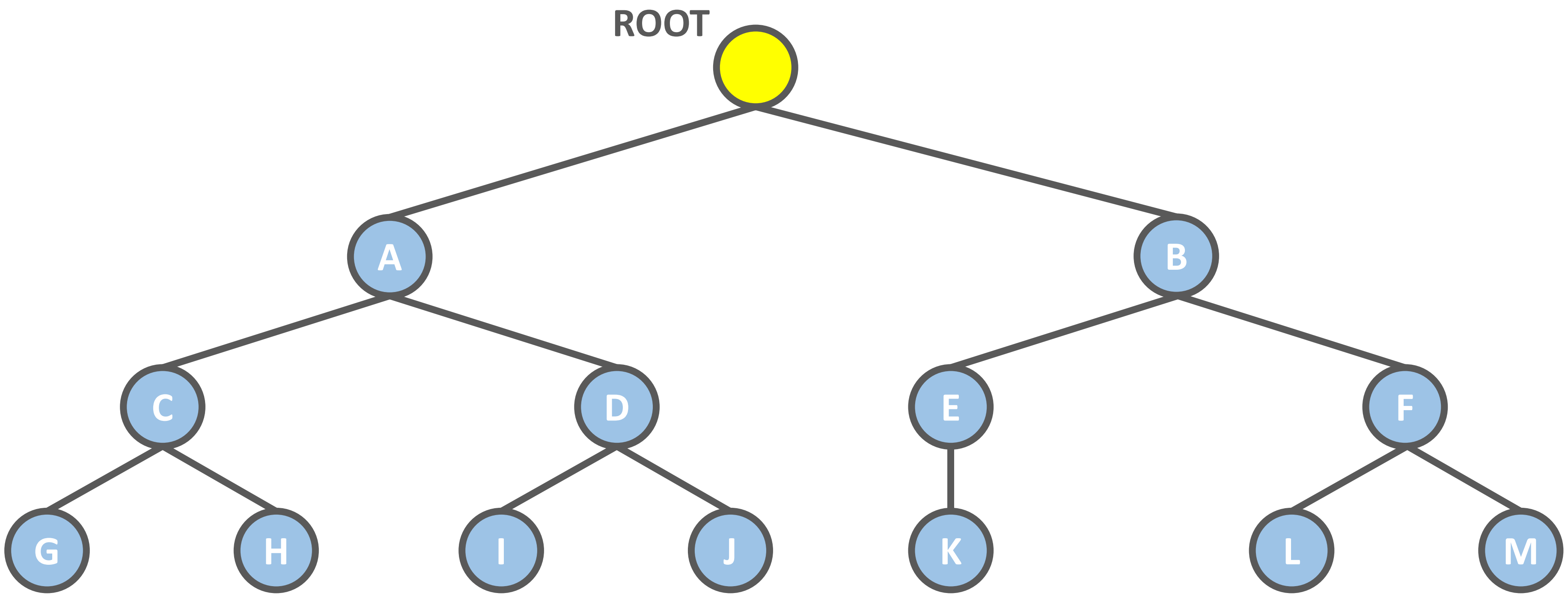
Advantage of Backtracking

(Algorithmic Problems)

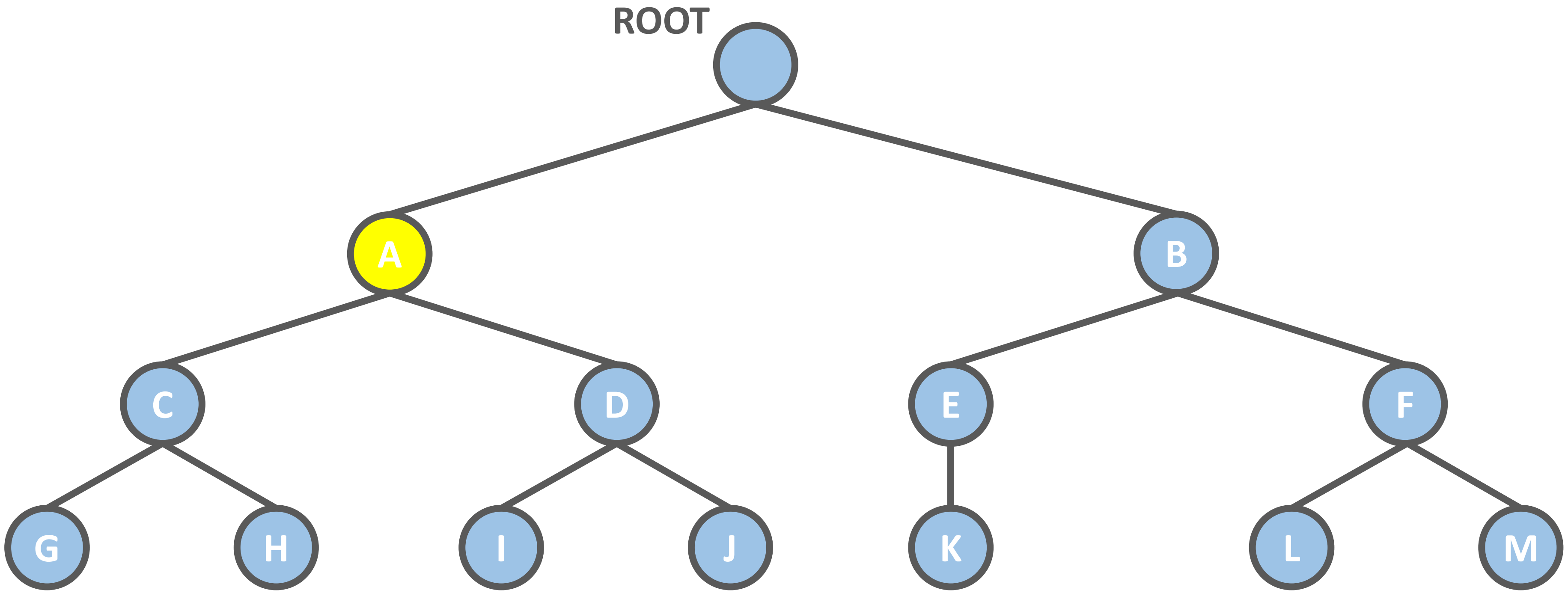
Brute-Force Search



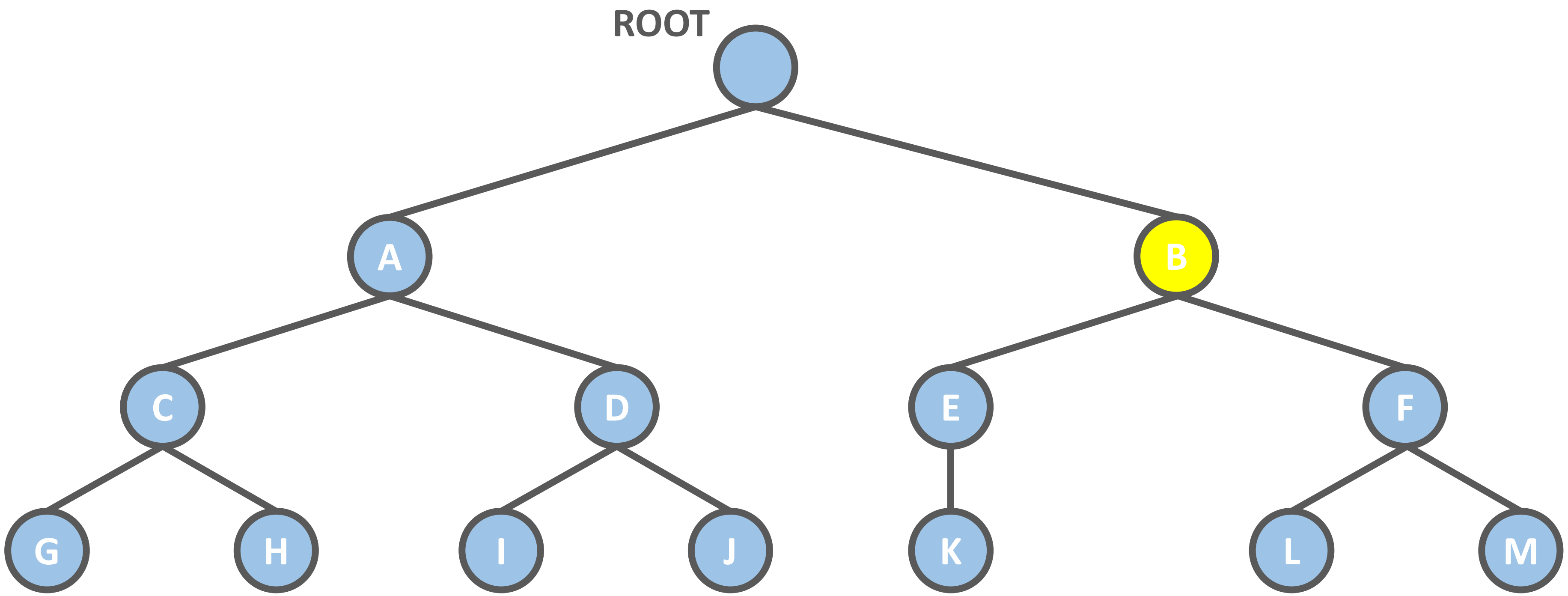
Brute-Force Search



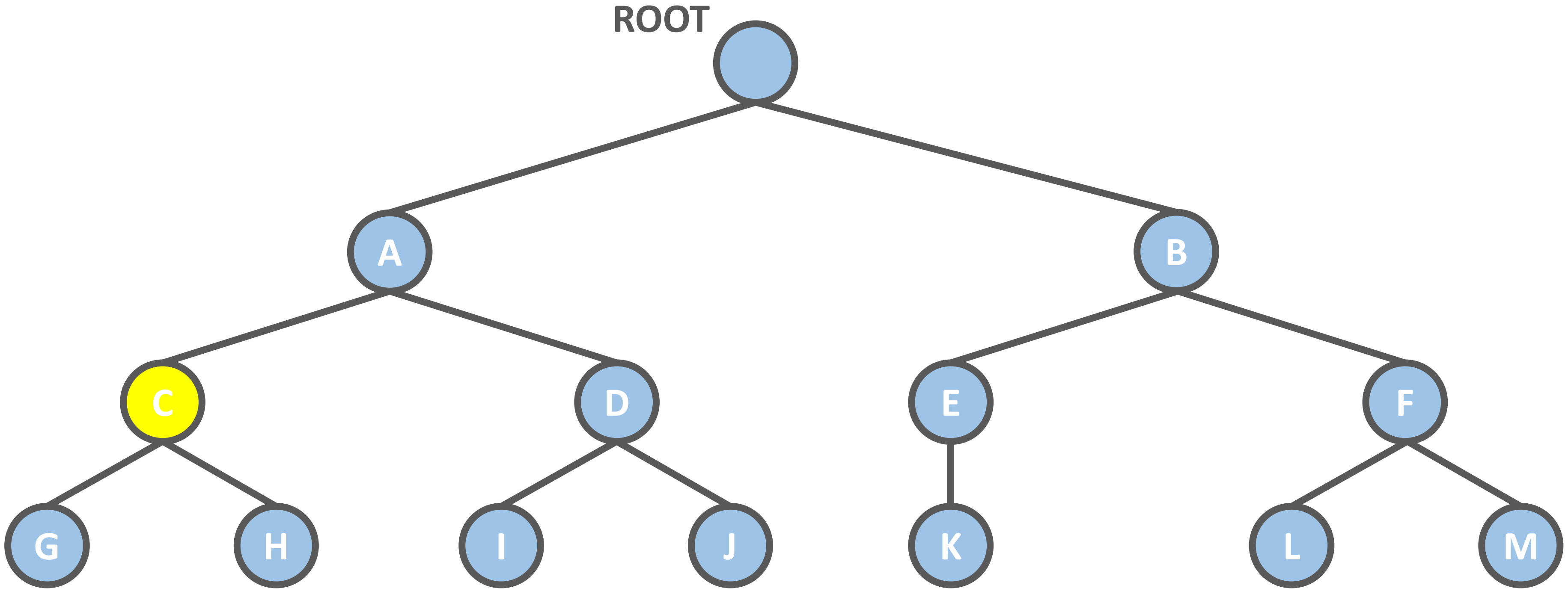
Brute-Force Search



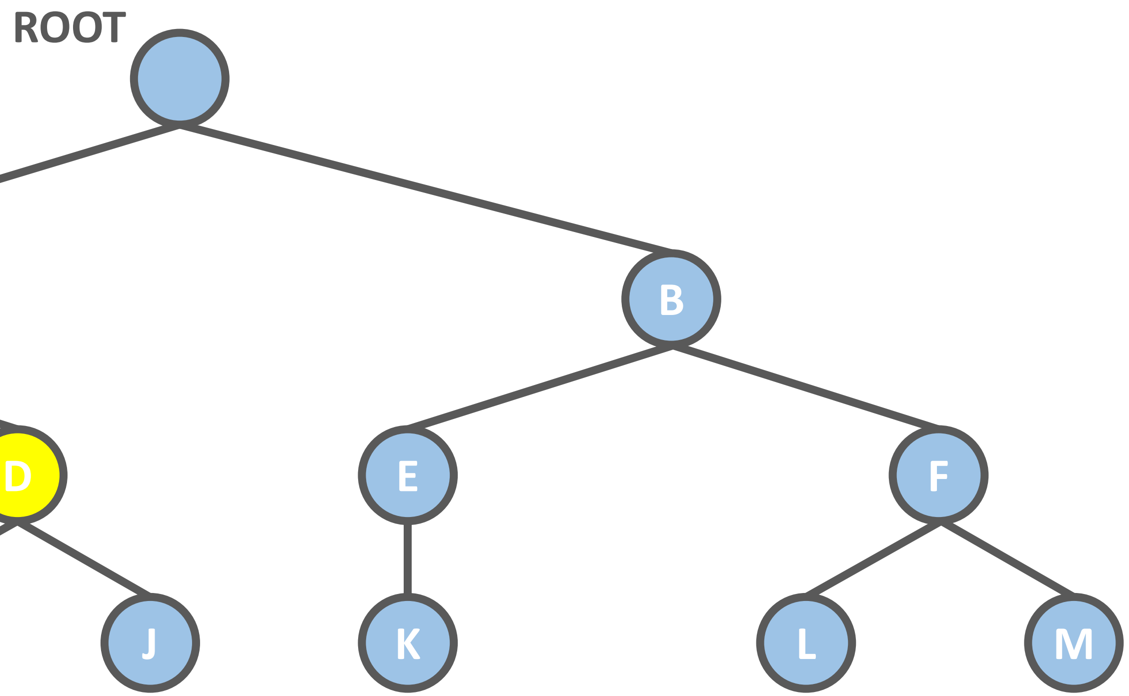
Brute-Force Search



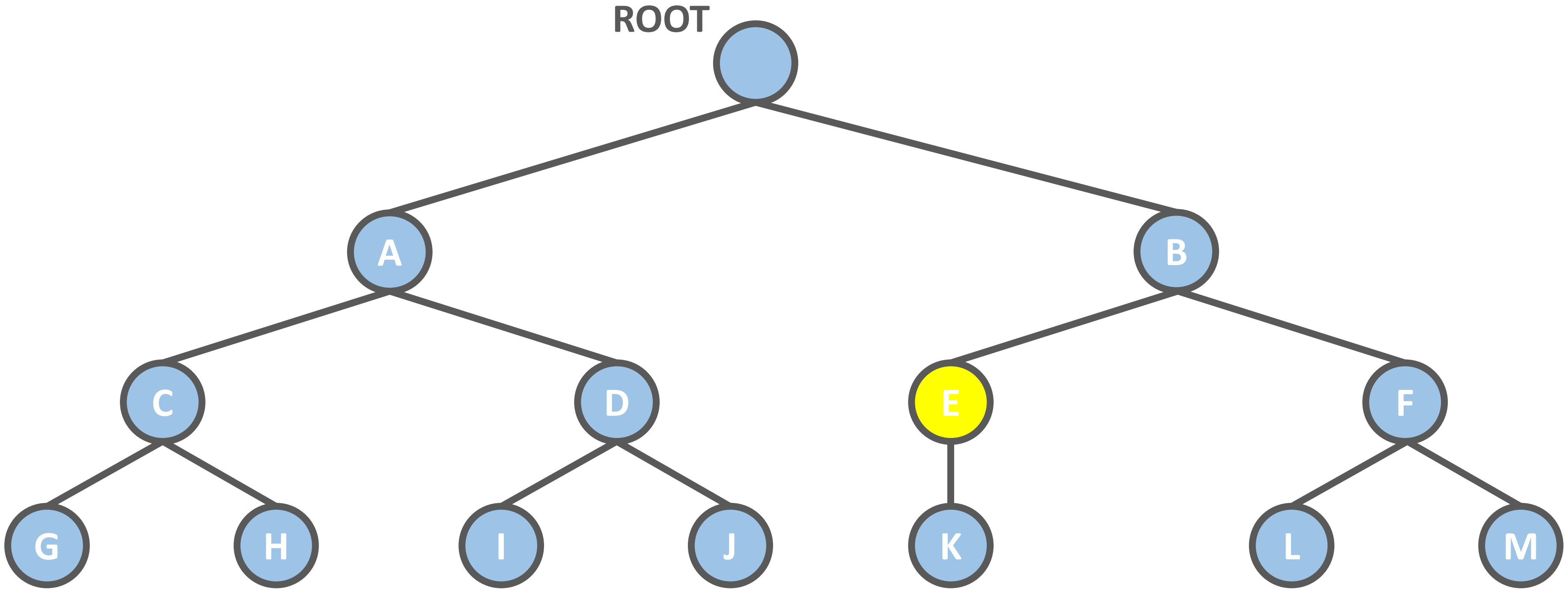
Brute-Force Search



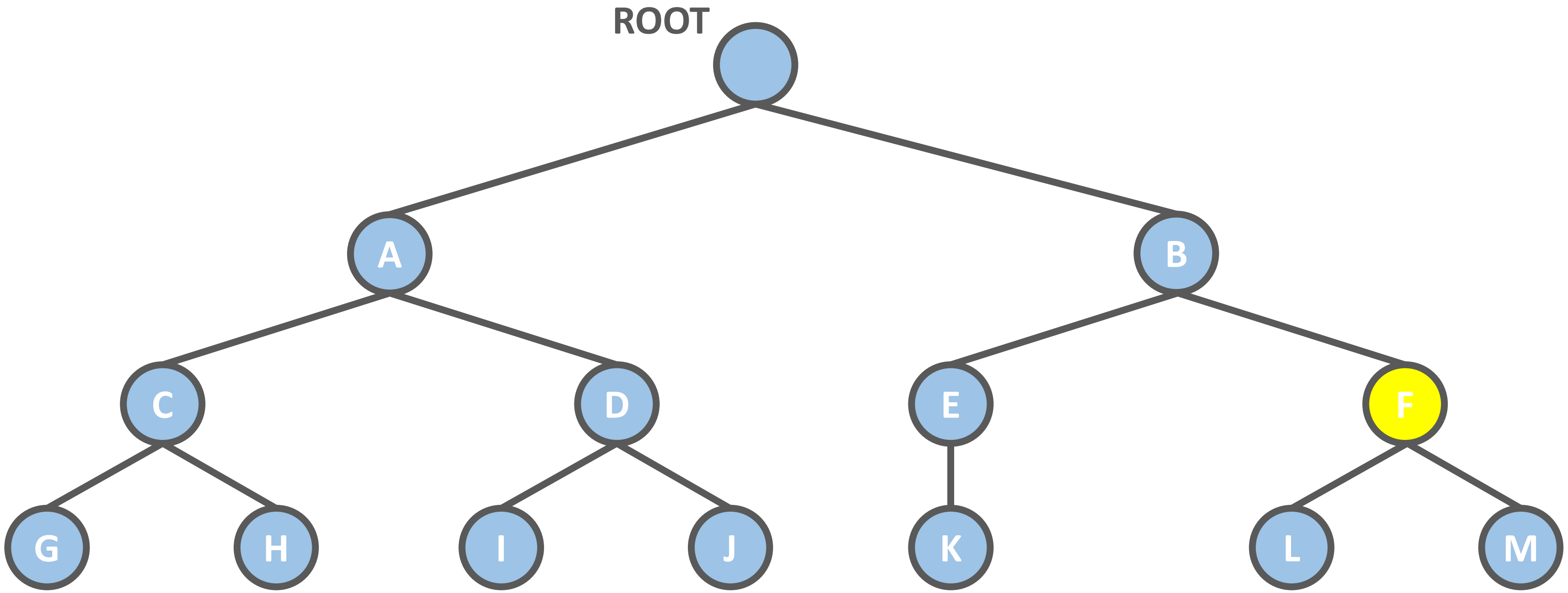
Brute-Force Search



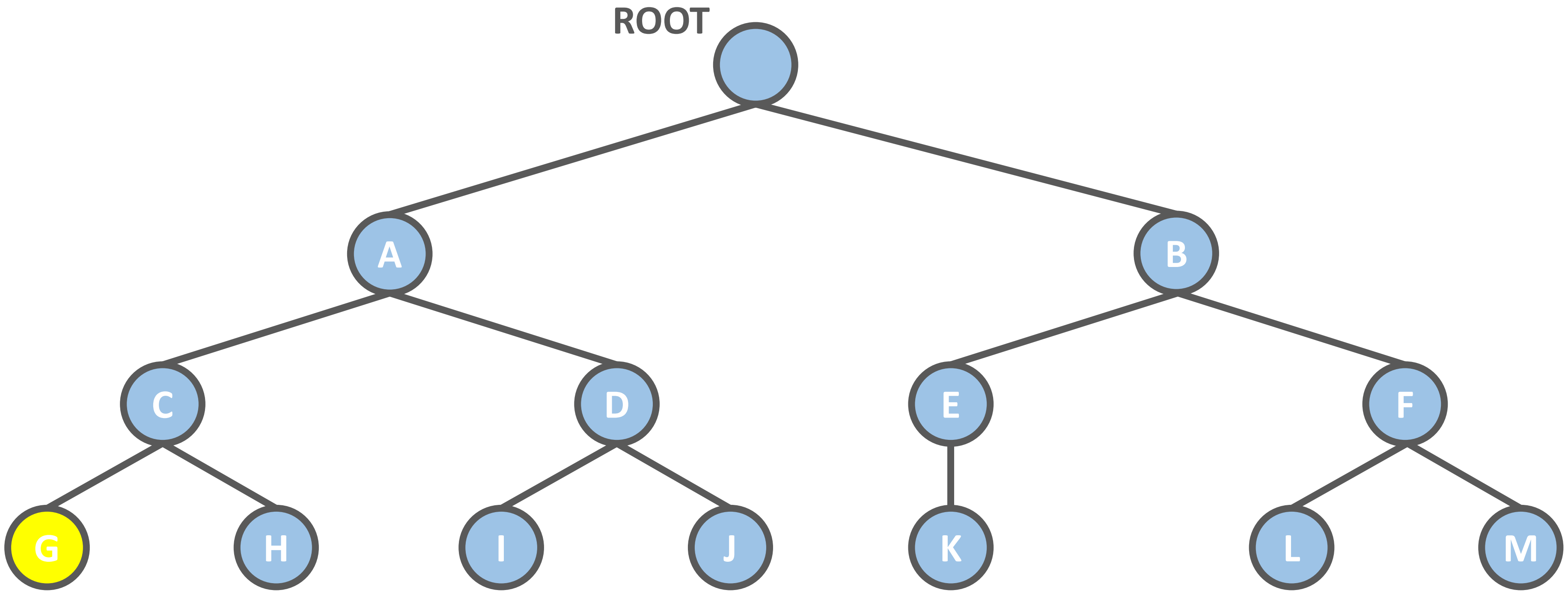
Brute-Force Search



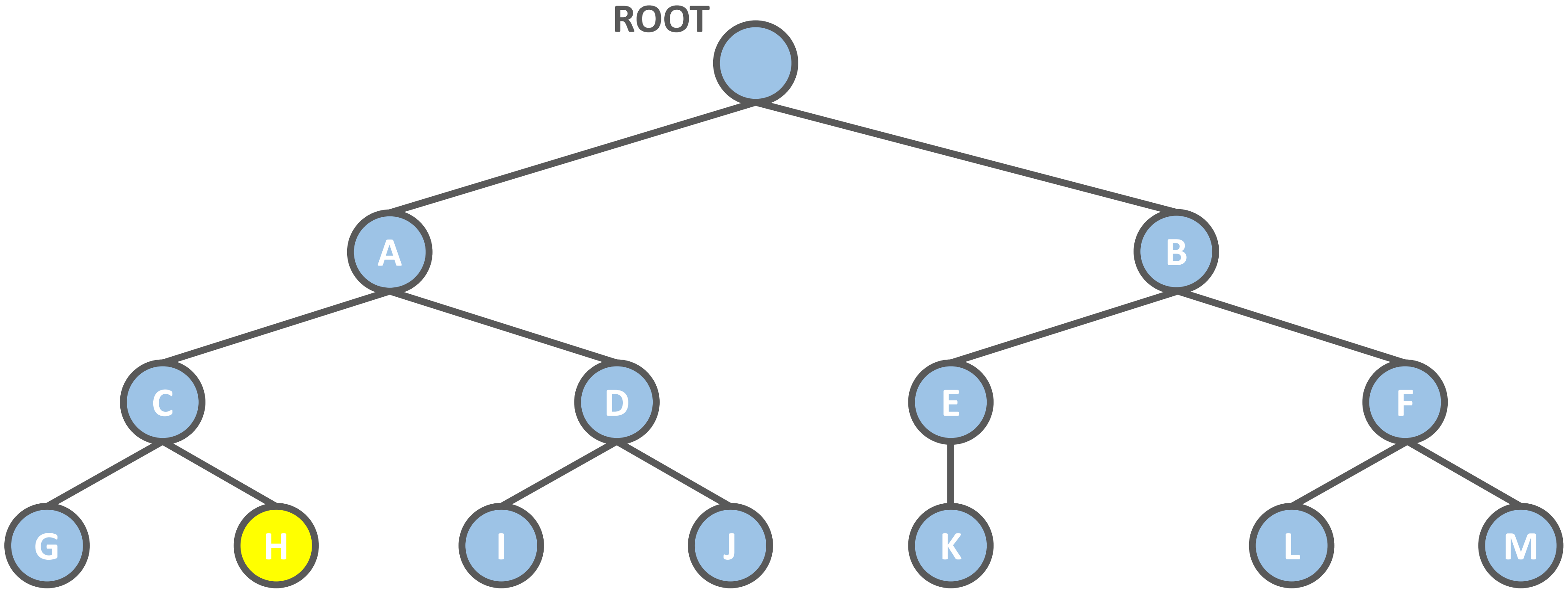
Brute-Force Search



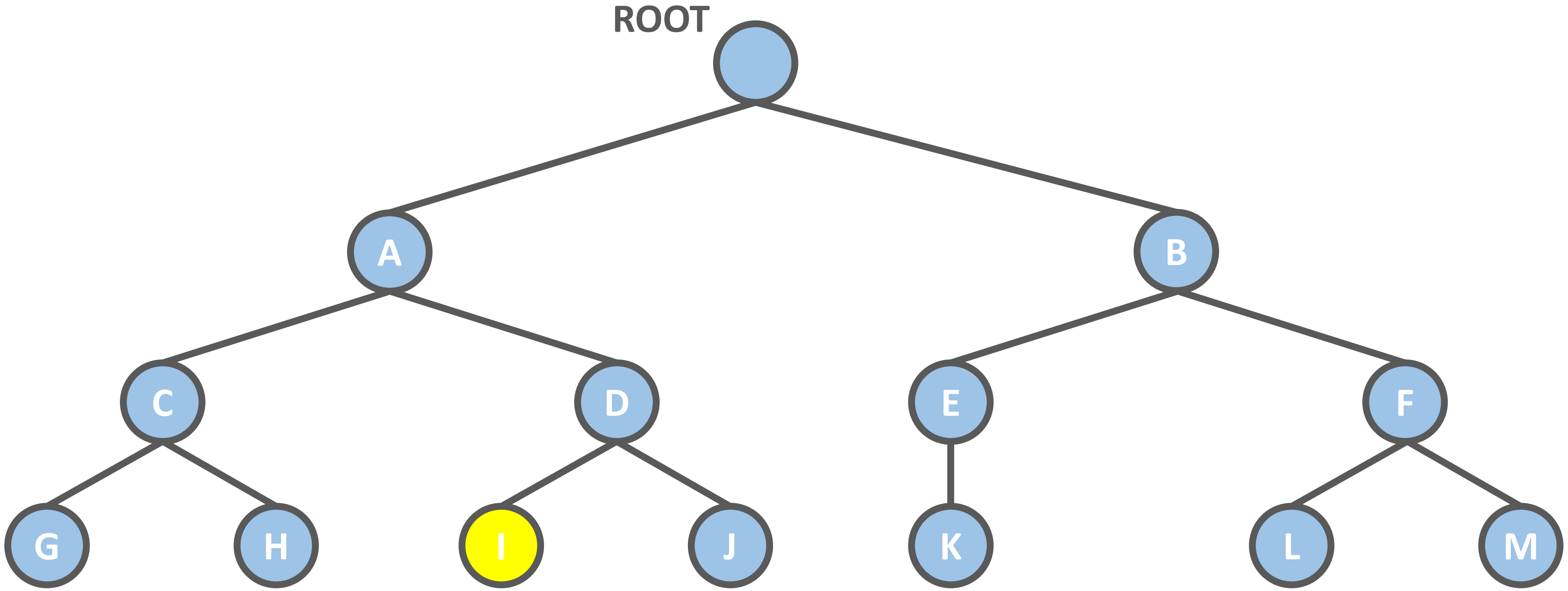
Brute-Force Search



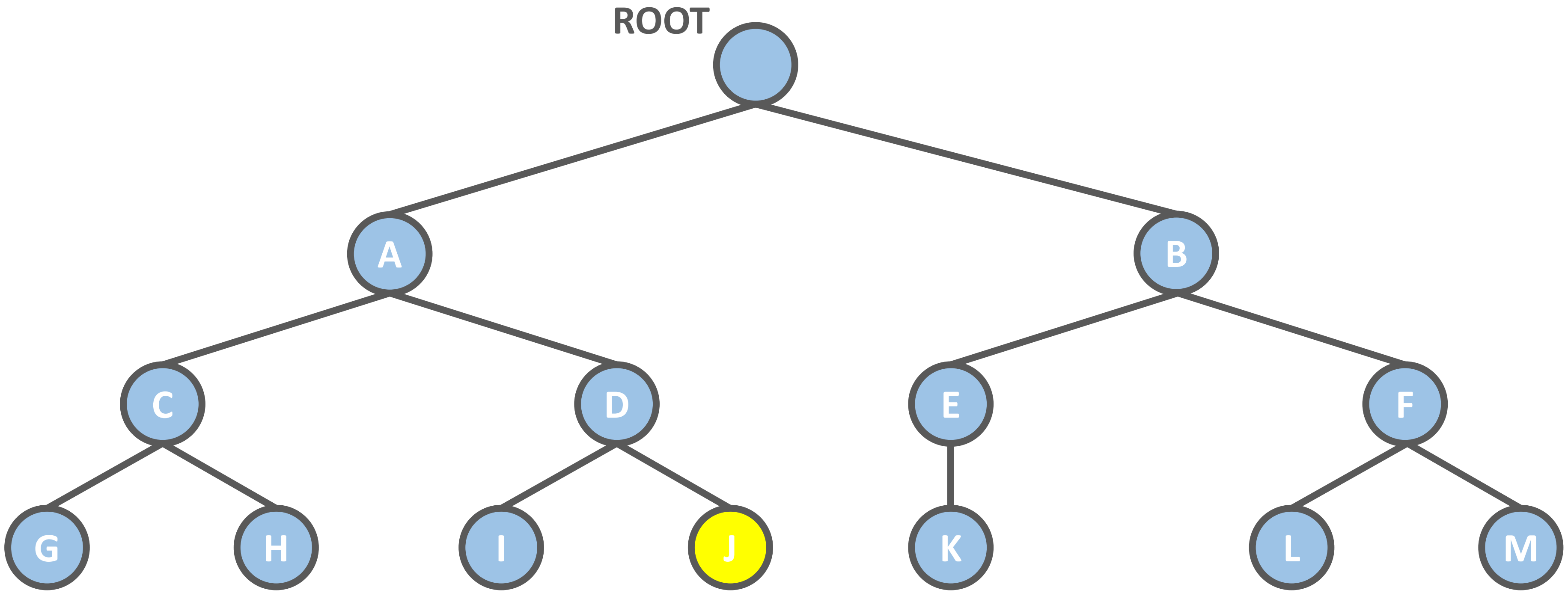
Brute-Force Search



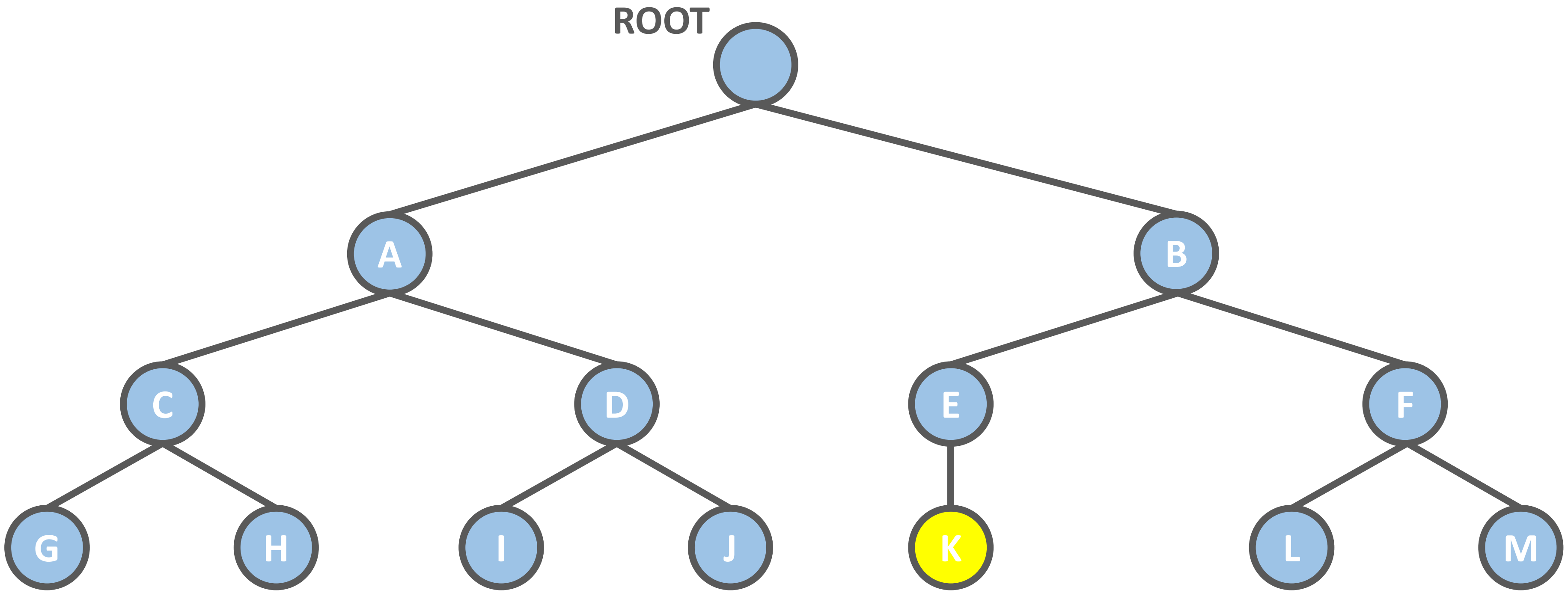
Brute-Force Search



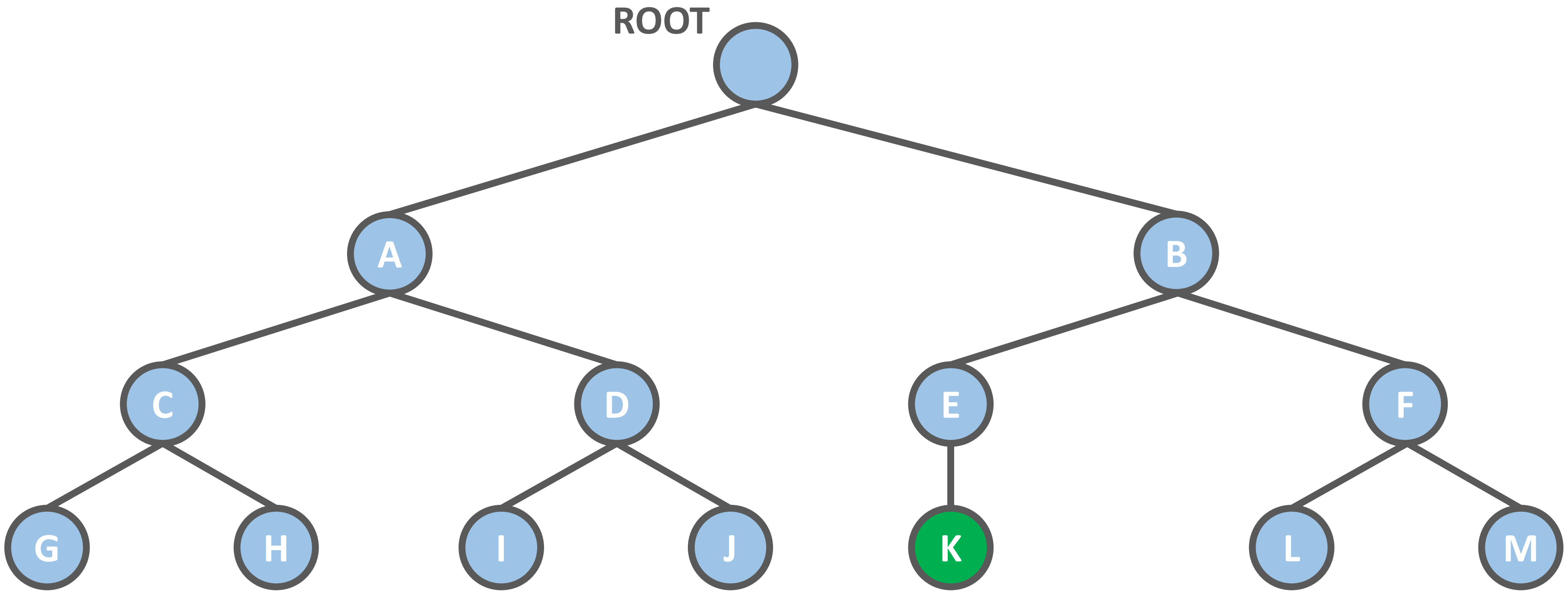
Brute-Force Search



Brute-Force Search

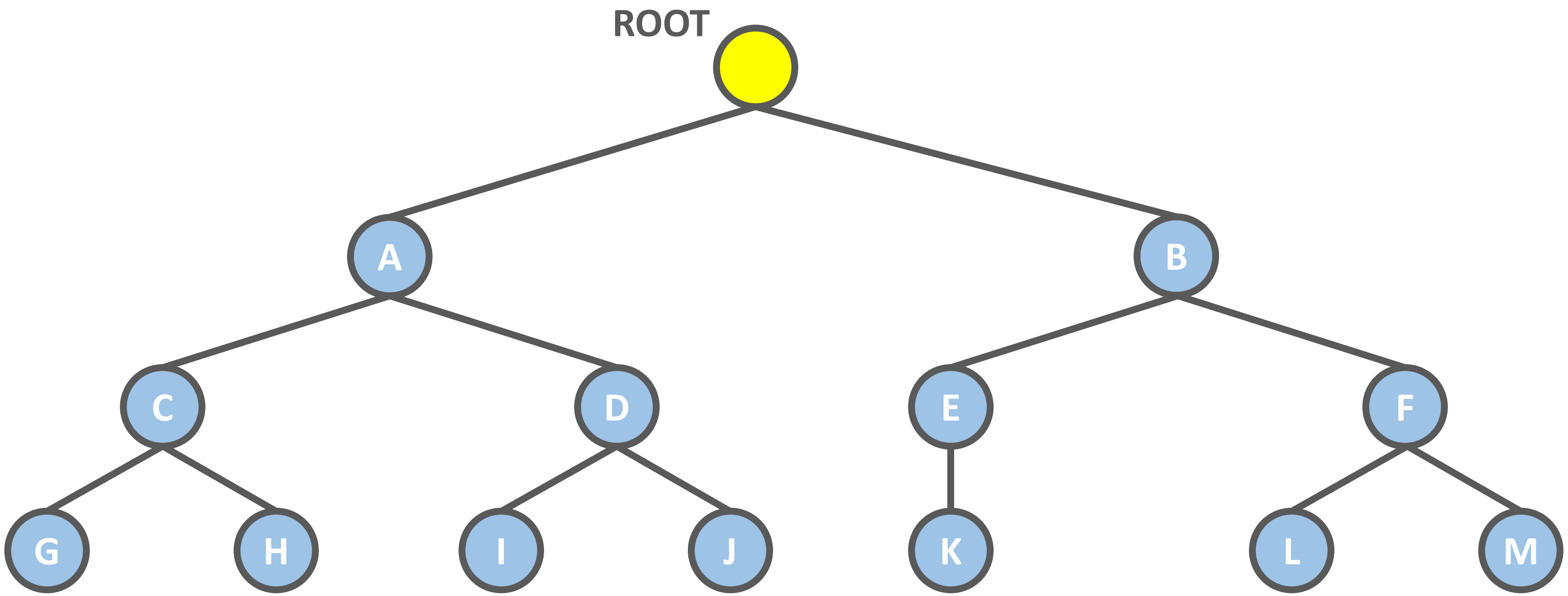


Brute-Force Search

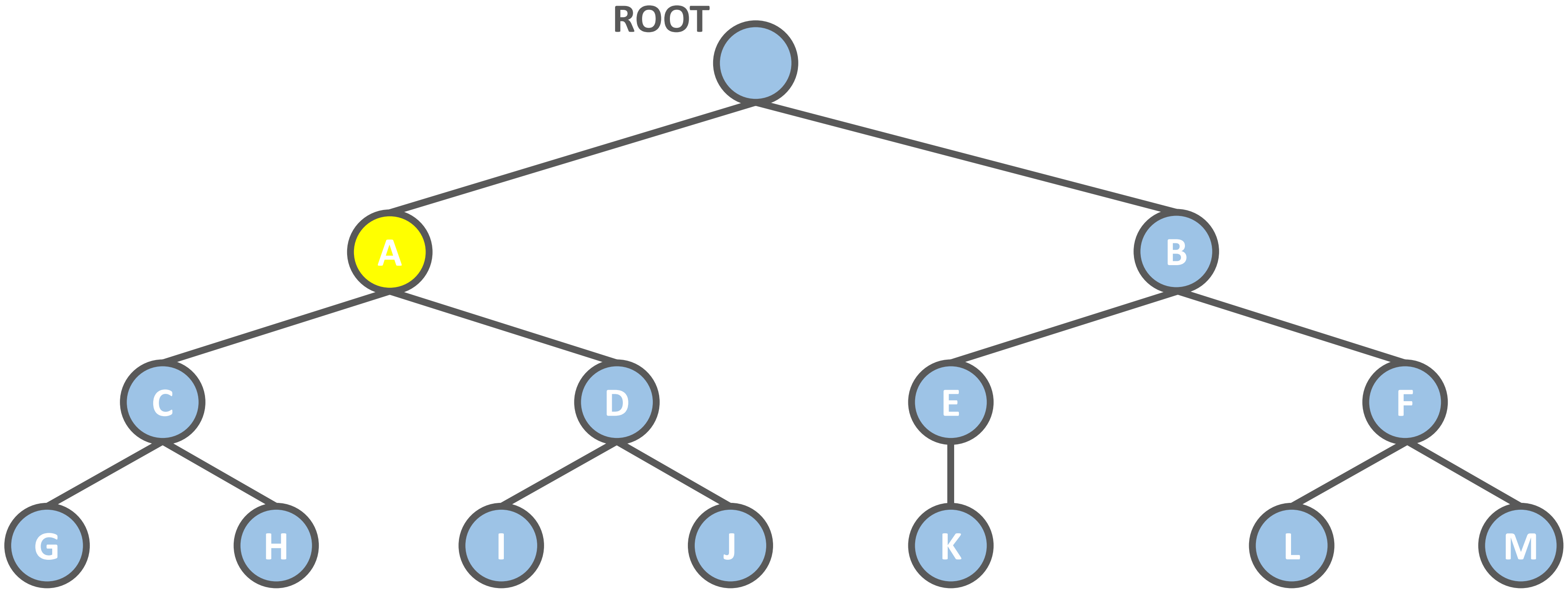


as you can see it takes
***12** steps with brute-force search*
to find the solution

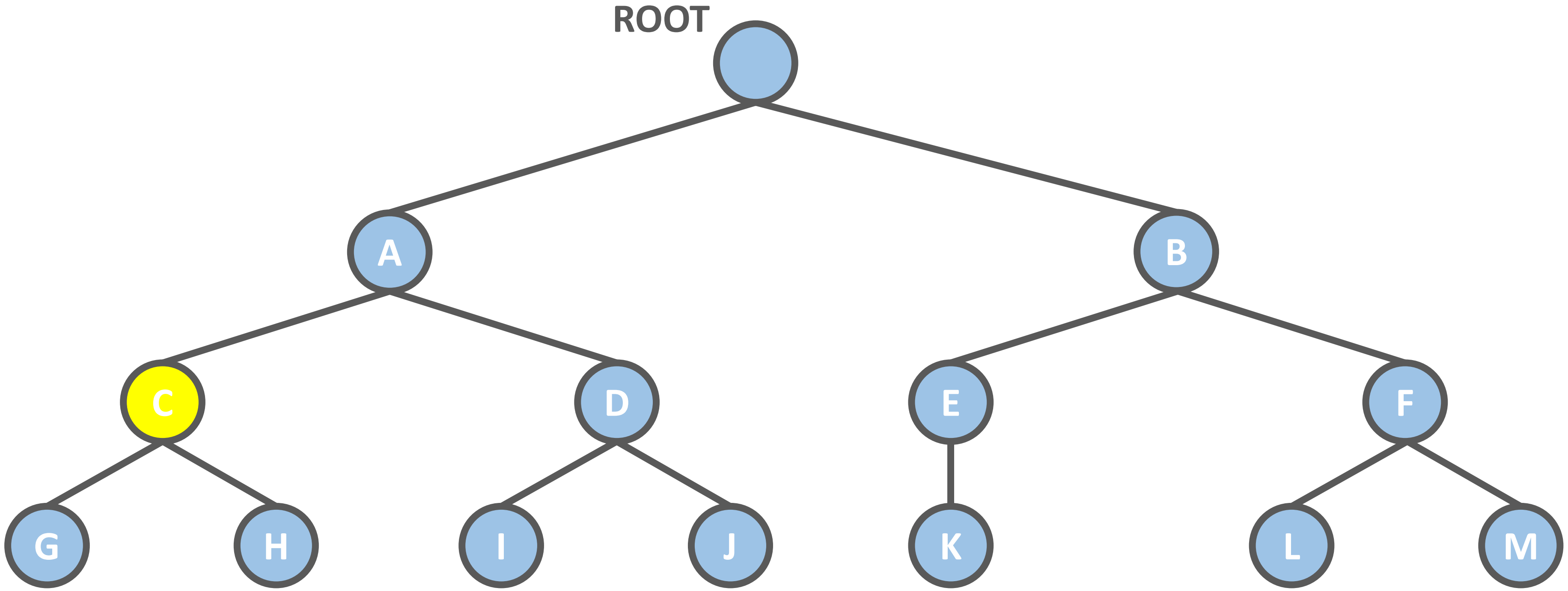
Backtracking (Depth-First Search)



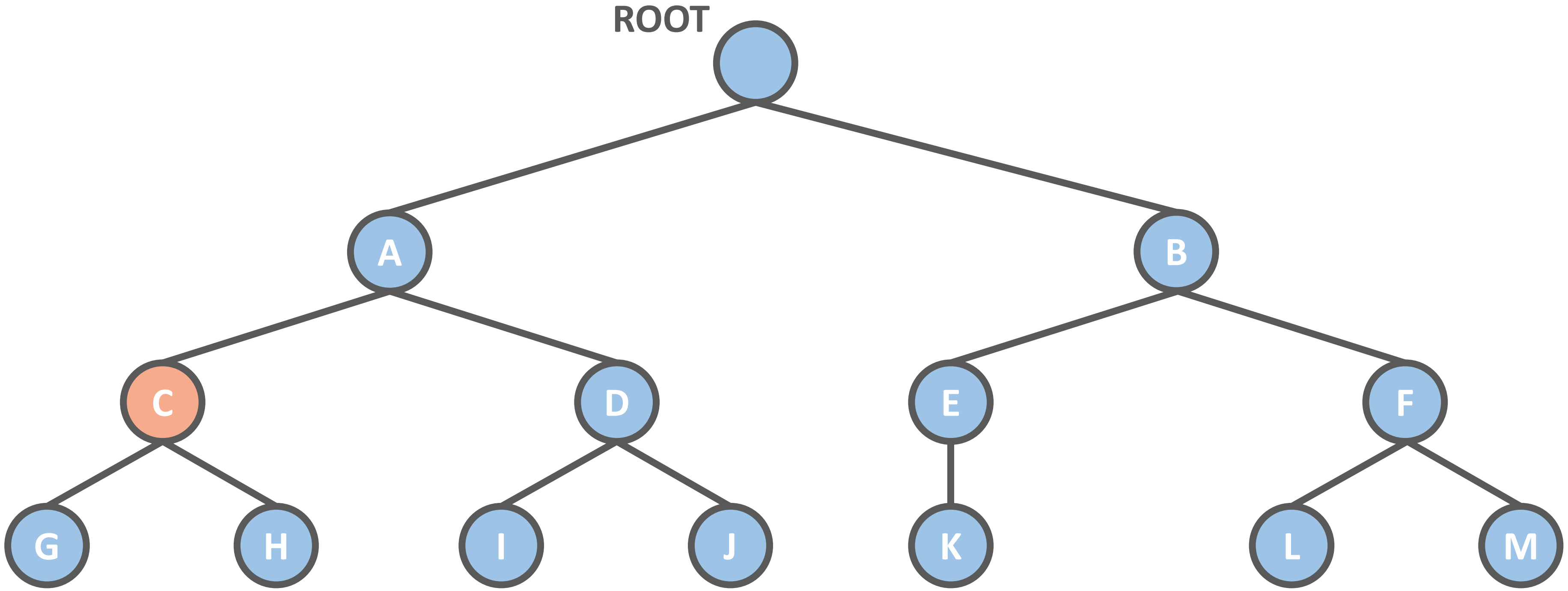
Backtracking (Depth-First Search)



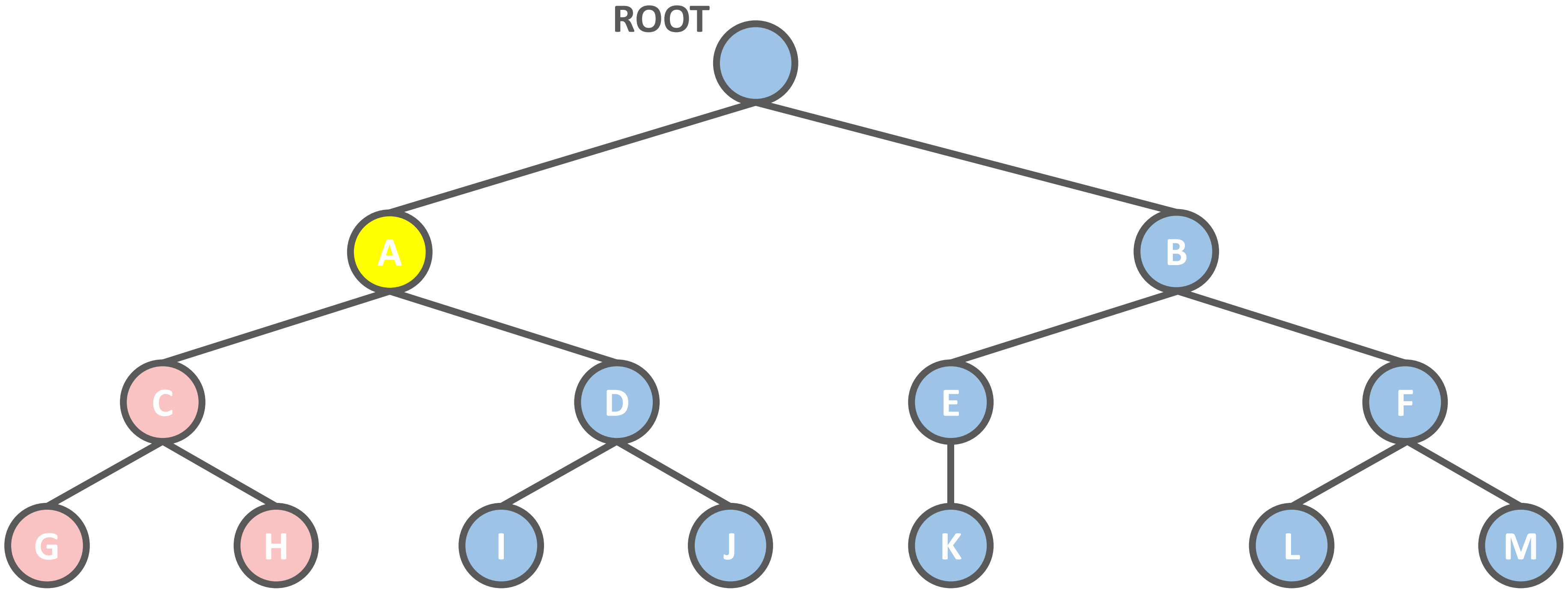
Backtracking (Depth-First Search)



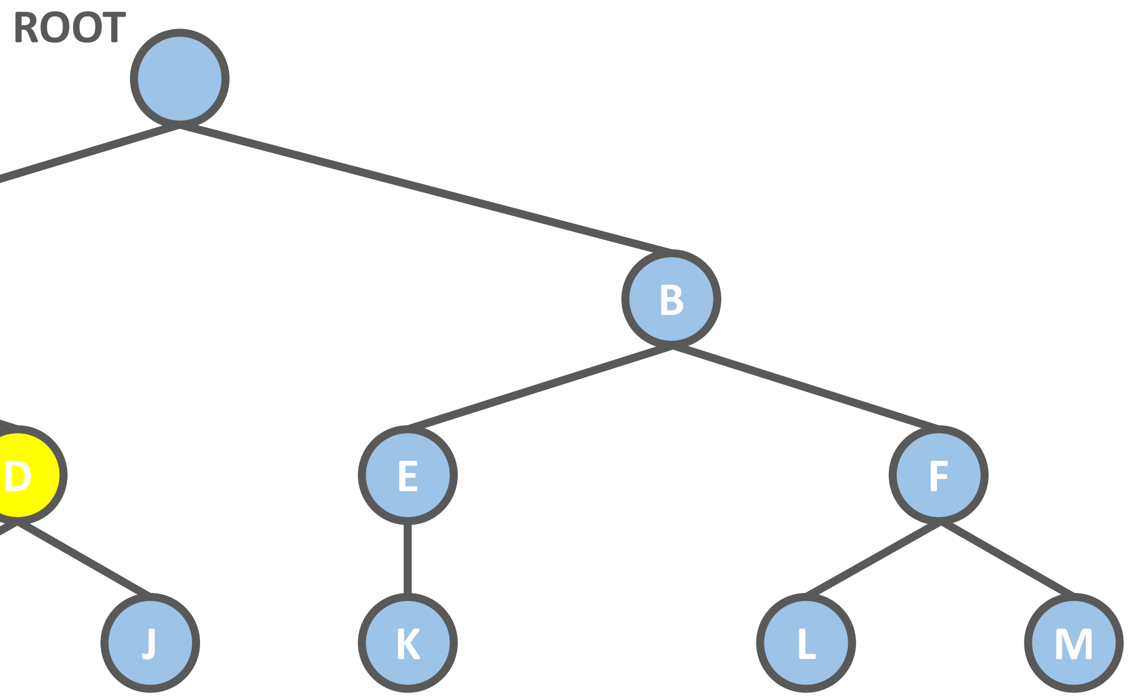
Backtracking (Depth-First Search)



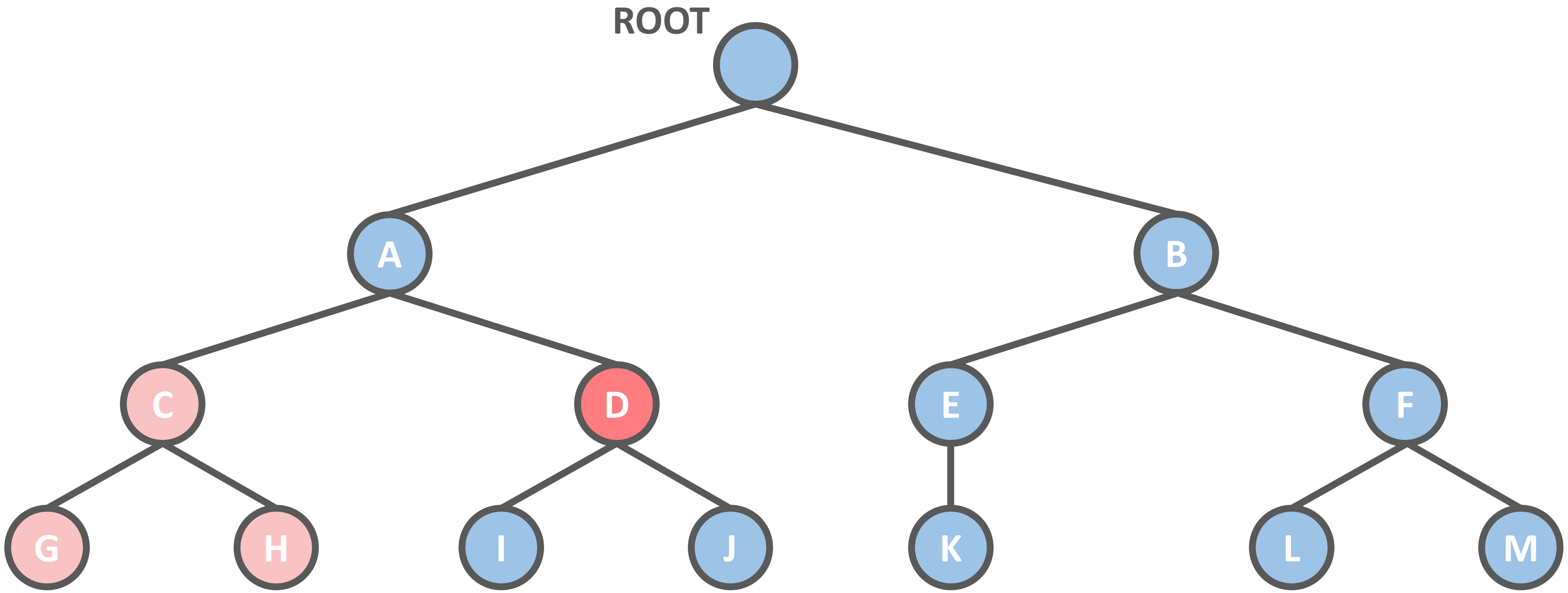
Backtracking (Depth-First Search)



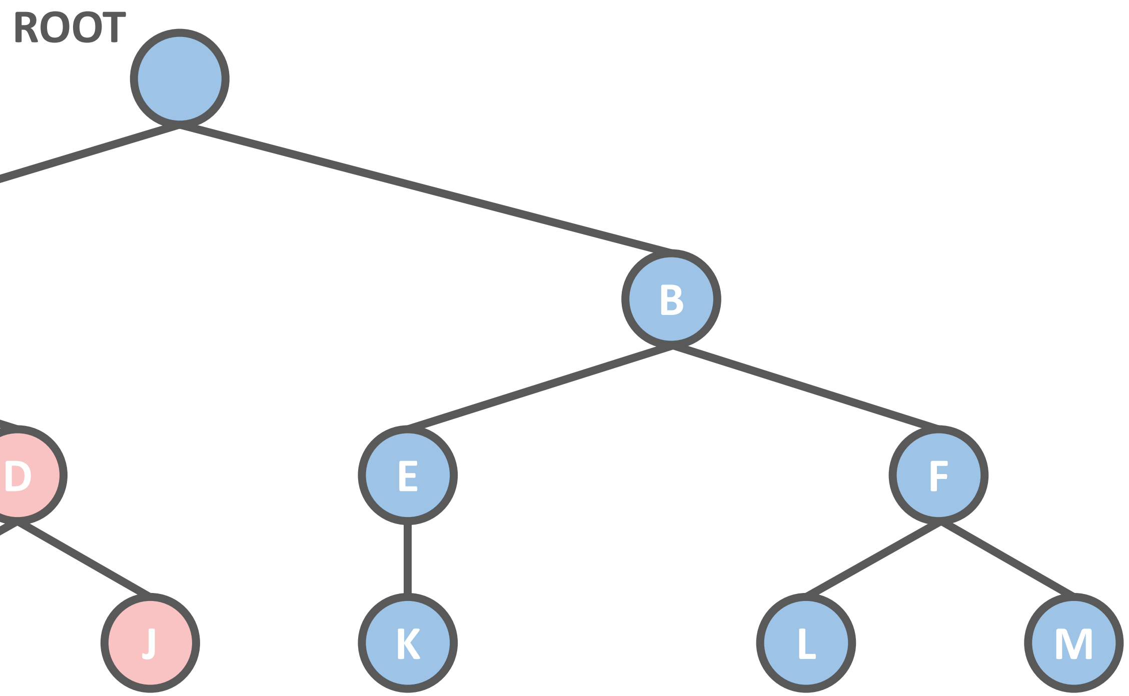
Backtracking (Depth-First Search)



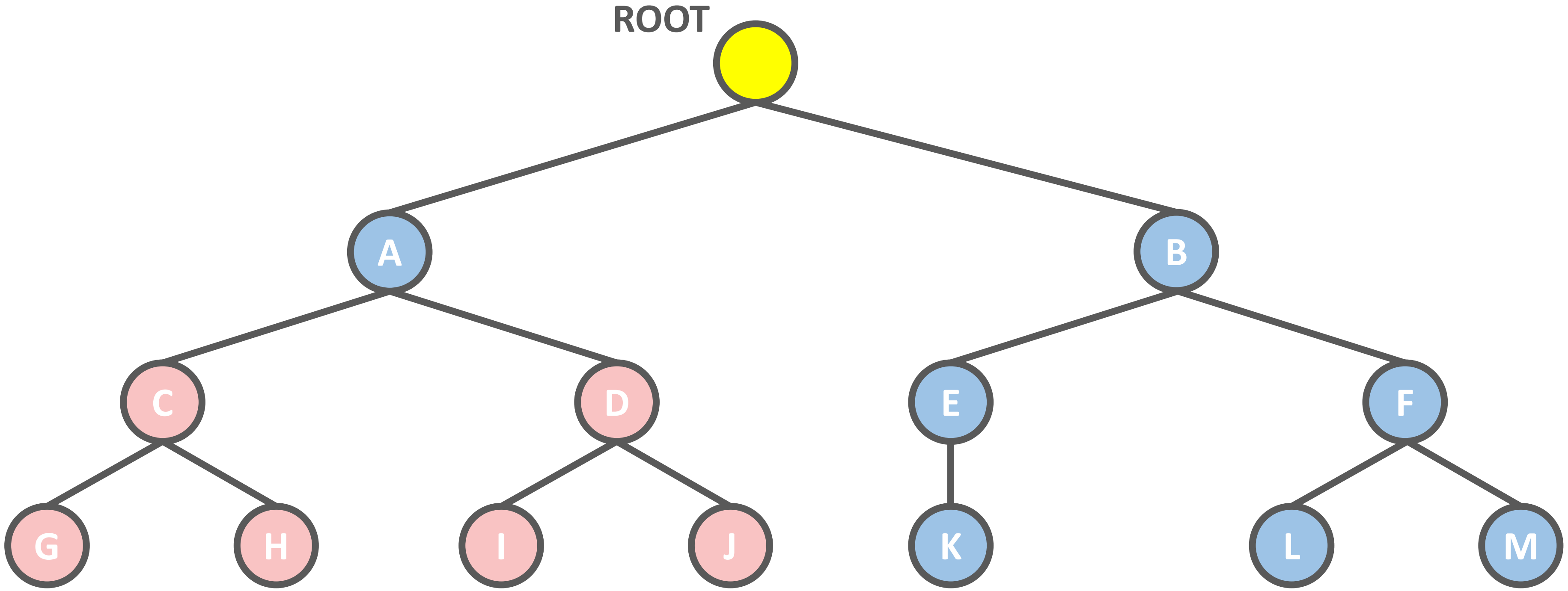
Backtracking (Depth-First Search)



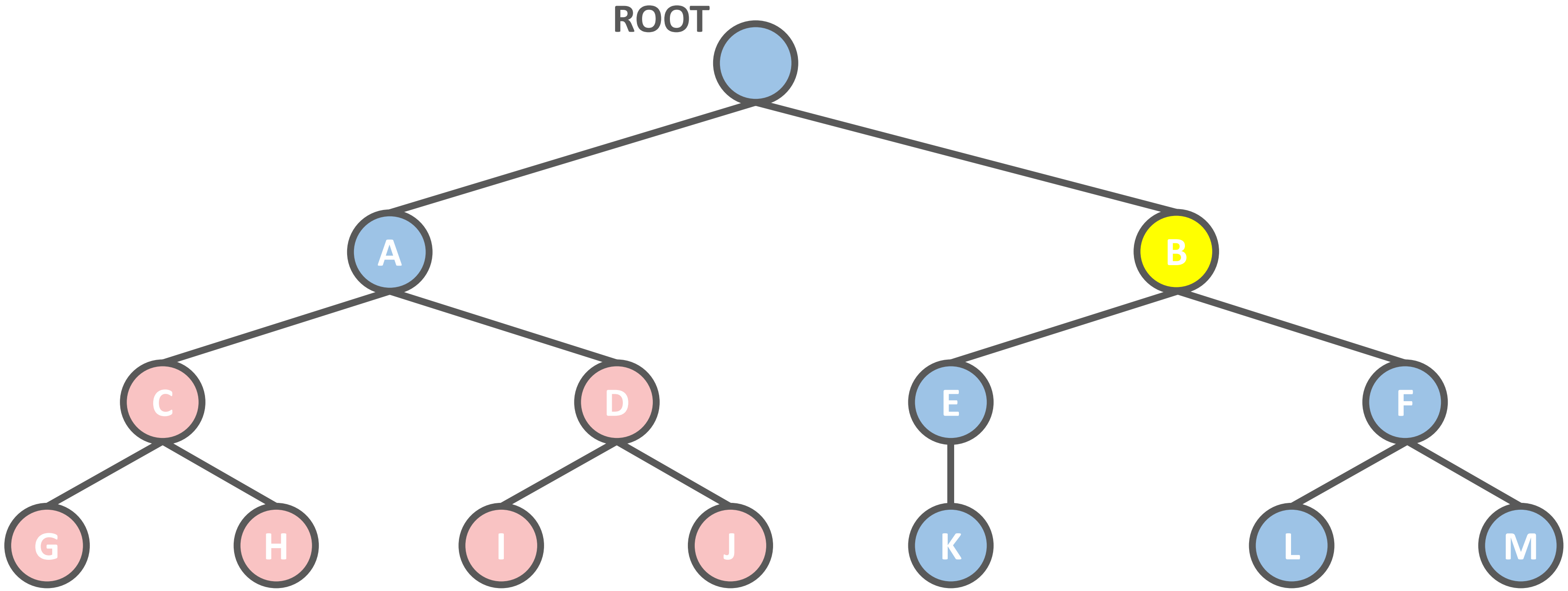
Backtracking (Depth-First Search)



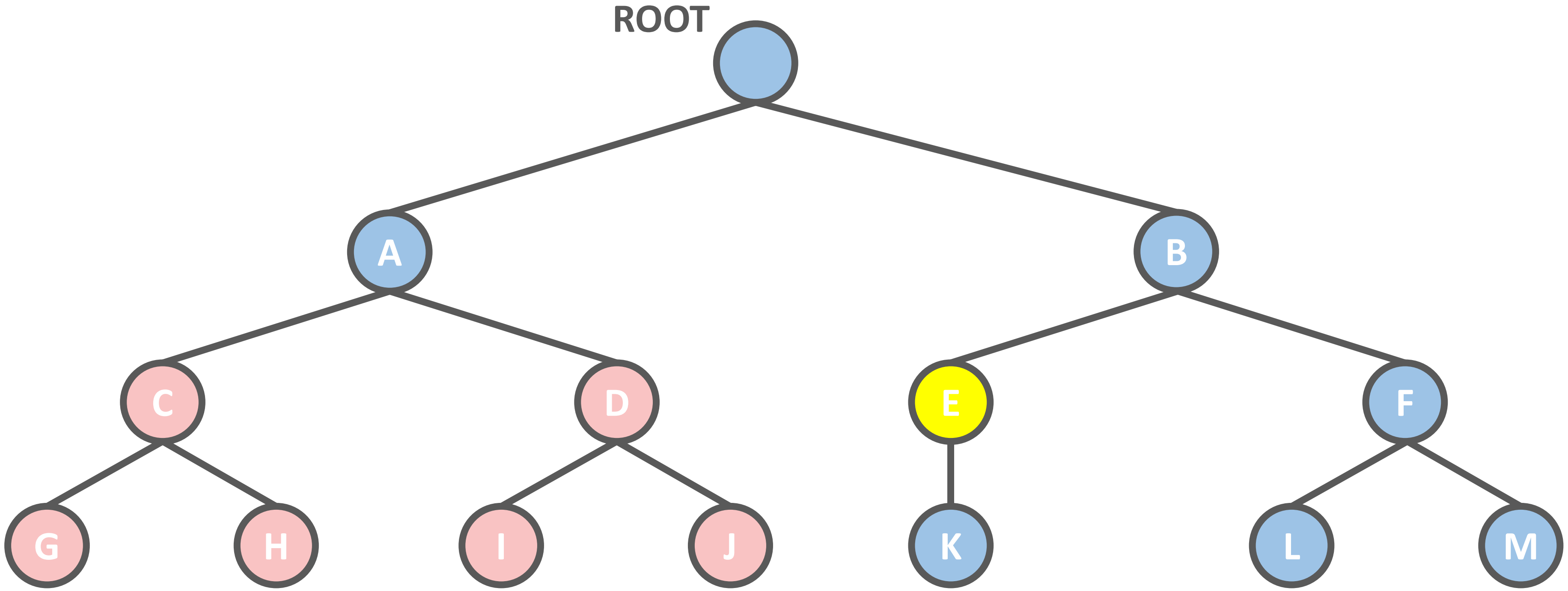
Backtracking (Depth-First Search)



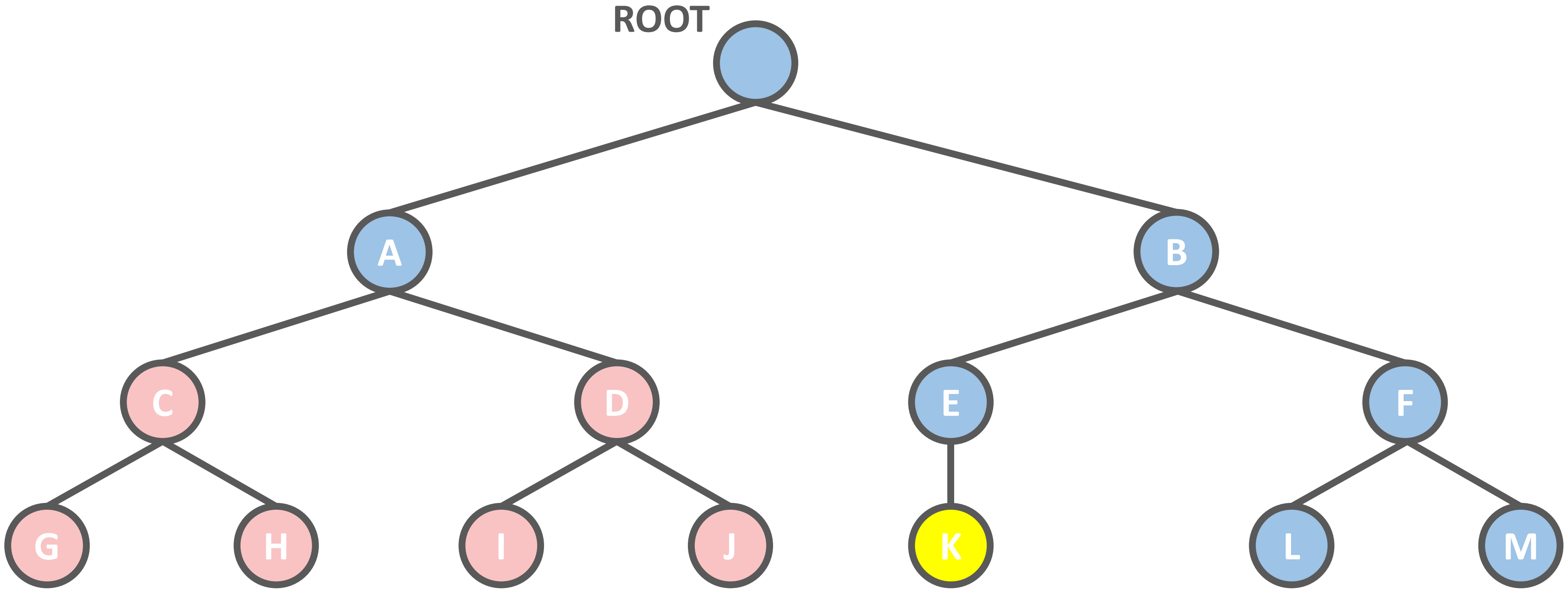
Backtracking (Depth-First Search)



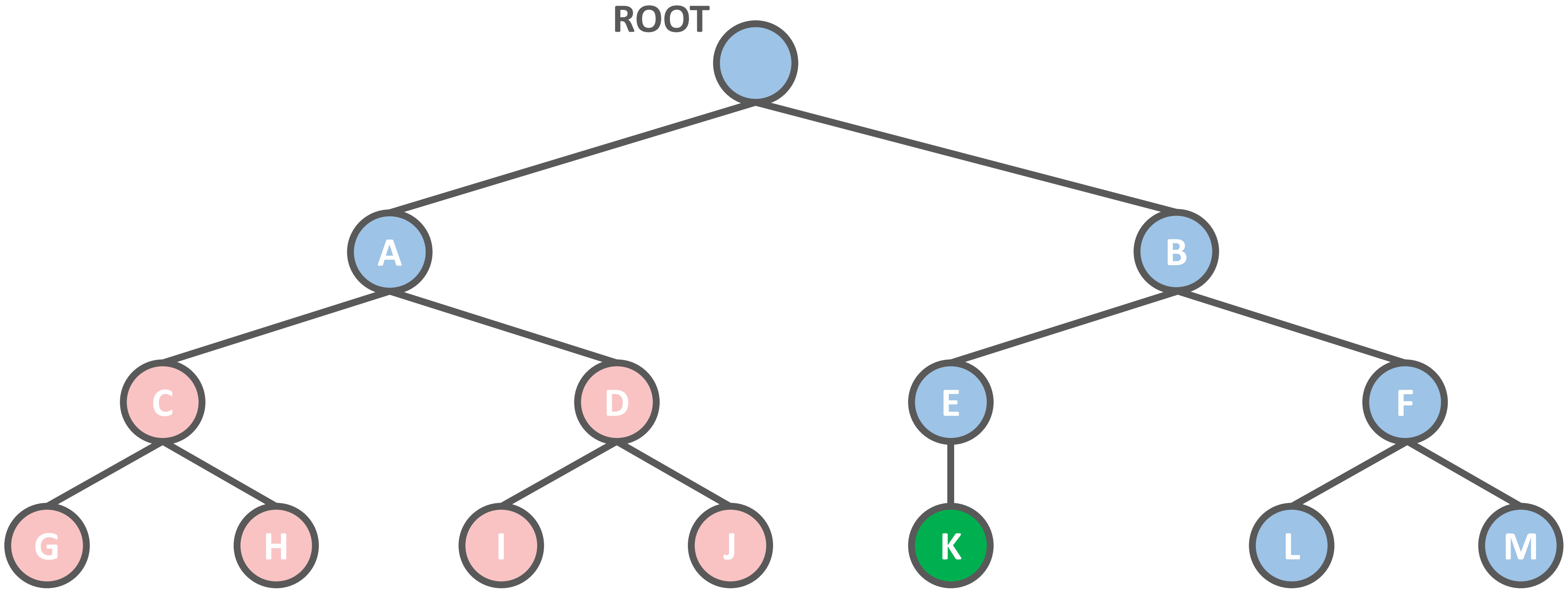
Backtracking (Depth-First Search)



Backtracking (Depth-First Search)



Backtracking (Depth-First Search)



as you can see it takes
***10** steps with backtracking*
to find the solution