Backtracking Algorithm - GeeksforGeeks

Source: https://www.geeksforgeeks.org/backtracking-algorithms/

Courses Tutorials Practice Jobs DSA Tutorial Interview Questions Quizzes Must Do Advanced DSA System Design Aptitude Puzzles Interview Corner DSA Python Technical Scripter 2026 Explore DSA Fundamentals Logic Building Problems Analysis of Algorithms Data Structures Array Data Structure String in Data Structure Hashing in Data Structure Linked List Data Structure Stack Data Structure Queue Data Structure Tree Data Structure Graph Data Structure Trie Data Structure Algorithms Searching Algorithms Sorting Algorithms Introduction to Recursion Greedy Algorithms Tutorial Graph Algorithms Dynamic Programming or DP Bitwise Algorithms Advanced Segment Tree Binary Indexed Tree or Fenwick Tree Square Root (Sqrt) Decomposition Algorithm Binary Lifting Geometry Interview Preparation Interview Corner GfG160 Practice Problem GeeksforGeeks Practice - Leading Online Coding Platform Problem of The Day - Develop the Habit of Coding DSA Course 90% Refund Backtracking Algorithm Last Updated : 18 Jan, 2026 Backtracking algorithms are like problem-solving strategies that help explore different options to find the solution. Work by trying out different paths and if one doesn't work, then backtrack and try another until he right one is found. It's like solving a puzzle by testing different pieces until they fit together perfectly. Useful for problems where you must generate all valid combinations, permutations, or subsets under constraints. Basics Introduction Backtracking vs Branch and Bound Standard Problems Permutations of a String The Knight’s tour problem Rat in a Maze N Queen Problem Word Addition Cryptographic Puzzle Subset Sum problem m Coloring Problem Hamiltonian Cycle Sudoku Magnet Puzzle Remove Invalid Parentheses Easy Problems Find all subsets Check for sum-string All paths between two vertices All distinct subsets Path of more than k length from a source All paths from a given source to a destination Medium Problems Tug of War 8 queen problem Combinational Sum Warnsdorff’s algorithm for Knight’s tour problem Paths from corner cell to middle cell in maze Maximum number possible by doing at-most K swaps Rat in a Maze with multiple jump allowed N Queen in O(n) space Hard Problems Power Set in Lexicographic order Word Break Problem using Backtracking Partition of a set into K subsets with equal sum Longest Possible Route in a Matrix with Hurdles Shortest safe route in a path with landmines All palindromic partitions of a string Printing all solutions in N-Queen Problem All longest common sub-sequences in lexicographical order Quick Links : DSA Tutorial Backtracking Interview Questions ‘Practice Problems’ on Backtracking ‘Quiz’ on Backtracking What is Backtracking Algorithm? Backtracking is a problem-solving algorithmic technique that involves finding a solution incrementally by trying different options and undoing them if they lead to a dead end . It is commonly used in situations where you need to explore multiple possibilities to solve a problem, like searching for a path in a maze or solving puzzles like Sudoku . When a dead end is reached, the algorithm backtracks to the previous decision point and explores a different path until a solution is found or all possibilities have been exhausted. How Does a Backtracking Algorithm Work? A backtracking algorithm works by recursively exploring all possible solutions to a problem. It starts by choosing an initial solution, and then it explores all possible extensions of that solution. If an extension leads to a solution, the algorithm returns that solution. If an extension does not lead to a solution, the algorithm backtracks to the previous solution and tries a different extension. The following is a general outline of how a backtracking algorithm works: Choose an initial solution. Explore all possible extensions of the current solution. If an extension leads to a solution, return that solution. If an extension does not lead to a solution, backtrack to the previous solution and try a different extension. Repeat steps 2-4 until all possible solutions have been explored. Example of Backtracking Algorithm Example: Finding the shortest path through a maze Input: A maze represented as a 2D array, where 0 represents an open space and 1 represents a wall. Algorithm: Start at the starting point. For each of the four possible directions (up, down, left, right), try moving in that direction. If moving in that direction leads to the ending point, return the path taken. If moving in that direction does not lead to the ending point, backtrack to the previous position and try a different direction. Repeat steps 2-4 until the ending point is reached or all possible paths have been explored. When to Use a Backtracking Algorithm? Backtracking algorithms are best used to solve problems that have the following characteristics: There are multiple possible solutions to the problem. The problem can be broken down into smaller subproblems. The subproblems can be solved independently. Applications of Backtracking Algorithm Backtracking algorithms are used in a wide variety of

applications, including: Solving puzzles (e.g., Sudoku, crossword puzzles) Finding the shortest path through a maze Scheduling problems Resource allocation problems Network optimization problems Combinatorial problems, such as generating permutations, combinations, or subsets. Comment Article Tags: Article Tags: Backtracking DSA Algorithms-Backtracking Algorithms-Recursion