Backtracking Algorithm

Last Updated : 16 Nov, 2024

Backtracking algorithms are like problem-solving strategies that help explore different options to find the best solution. They work by trying out different paths and if one doesn't work, they backtrack and try another until they find the right one. It's like solving a puzzle by testing different pieces until they fit together perfectly.

Introduction to
Backtracking

Backtracking

Basic of Backtracking Algorithm:

1. Introduction to Backtracking
2. Backtracking vs Recursion
3. Backtracking vs Branch and Bound

Standard Problems

- Permutations of a String
- The Knight's tour problem
- Rat in a Maze
- N Queen Problem | Backtracking-3
- Subset Sum problem

- m Coloring Problem
- Hamiltonian Cycle
- Sudoku | Backtracking-7
- Magnet Puzzle
- Remove Invalid Parentheses
- A backtracking approach to generate n bit Gray Codes

Easy Problem:

- Backtracking to find all subsets

DSA Practice Backtracking Interview Problems on Backtracking MCQs on Backtracking Tutorial on Backtrackin

- Count all possible paths between two vertices
- Find all distinct subsets of a given set
- Find if there is a path of more than k length from a source
- Print all paths from a given source to a destination
- Print all possible strings that can be made by placing spaces

Medium Problems:

- Tug of War
- 8 queen problem
- Combinational Sum
- Warnsdorff's algorithm for Knight's tour problem
- Find paths from corner cell to middle cell in maze
- Find Maximum number possible by doing at-most K swaps
- Rat in a Maze with multiple steps or 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
- Find shortest safe route in a path with landmines
- Print all palindromic partitions of a string
- Printing all solutions in N-Queen Problem
- Print all longest common sub-sequences in lexicographical order

Quick Links :

- Learn Data Structure and Algorithms | DSA Tutorial
- Top 20 Backtracking Algorithm Interview Questions
- 'Practice Problems' on Backtracking
- 'Quiz' on Backtracking
- 'Videos' 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:

1. Choose an initial solution.
2. Explore all possible extensions of the current solution.
3. If an extension leads to a solution, return that solution.
4. If an extension does not lead to a solution, backtrack to the previous solution and try a different extension.
5. 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:

1. Start at the starting point.
2. For each of the four possible directions (up, down, left, right), try moving in that direction.
3. If moving in that direction leads to the ending point, return the path taken.
4. If moving in that direction does not lead to the ending point, backtrack to the previous position and try a different direction.
5. 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.

H hare... 7

Next Article

Introduction to Backtracking

Similar Reads

Backtracking Algorithm in Python

Backtracking is a problem-solving algorithmic technique that involves finding a solution incrementally by trying different options and undoing them if they lead...

4 min read

Top 20 Backtracking Algorithm Interview Questions

N Queens ProblemWarnsdorff's AlgorithmWord Break ProblemRemove Invalid ParenthesisMatch a pattern and string using regular expressionFind Path from...

1 min read

Algorithms | Backtracking | Question 1

Which of the following is not a backtracking algorithm? (A) Knight tour problem (B) N queen problem (C) Tower of hanoi (D) M coloring problem Answer: (C)...

1 min read

Print the DFS traversal step-wise (Backtracking also)

Given a graph, the task is to print the DFS traversal of a graph which includes every step including the backtracking. 1st step:- 0 -> 1 2nd step:- 1 -> 5 3rd...

10 min read

Rat in a Maze | Backtracking using Stack

Prerequisites - Recursion, Backtracking and Stack Data Structure. A Maze is given as N*M binary matrix of blocks and there is a rat initially at (0, 0) ie. maze[0][0]...

15+ min read

Difference between Backtracking and Branch-N-Bound technique

Algorithms are the methodical sequence of steps which are defined to solve complex problems. In this article, we will see the difference between two such...

4 min read

Maximum size subset with given sum using Backtracking

Given an array arr[] consisting of N integers and an integer K, the task is to find the length of the longest subsequence with a sum equal to K.Examples: Input:...

9 min read

Generate all distinct subsequences of array using backtracking

Given an array arr[] consisting of N positive integers, the task is to generate all distinct subsequences of the array. Examples: Input: arr[] = {1, 2, 2}Output: {} {1}...

7 min read

What is the difference between Backtracking and Recursion?

What is Recursion? The process in which a function calls itself directly or indirectly is called recursion and the corresponding function is called a recursive...

2 min read

Backtracking meaning in DSA

Backtracking can be defined as a general algorithmic technique that considers searching every possible combination in order to solve a computational problem...

3 min read

Article Tags : Backtracking DSA Algorithms-Recursion Algorithms-Backtracking

Practice Tags : Backtracking

Corporate & Communications Address:-
A-143, 9th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305) | Registered Address:- K 061,

Tower K, Gulshan Vivante Apartment,
Sector 137, Noida, Gautam Buddh
Nagar, Uttar Pradesh, 201305

Company

About Us
Legal
In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program
GeeksforGeeks Community

Languages

Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial
Tutorials Archive

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design

DevOps

Git
Linux
AWS
Docker
Kubernetes

Engineering Maths
Software Development
Software Testing

Azure
GCP
DevOps Roadmap

System Design
High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

Inteview Preparation
Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

School Subjects
Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

GeeksforGeeks Videos
DSA
Python
Java
C++
Web Development
Data Science
CS Subjects