Advanced Data Structures - GeeksforGeeks

Source: https://www.geeksforgeeks.org/advanced-data-structures/

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 Advanced Data Structures Last Updated : 23 Jul, 2025 Advanced Data Structures refer to complex and specialized arrangements of data that enable efficient storage, retrieval, and manipulation of information in computer science and programming. These structures go beyond basic data types like arrays and lists , offering sophisticated ways to organize and manage data for optimal performance in various algorithms and applications. Self Balancing BSTs: AVL Tree: Introduction to AVL Tree Insertion in AVL Tree Deletion in an AVL Tree AVL with duplicate keys Splay Tree: Introduction to Splay tree data structure Searching in Splay Tree Insertion in Splay Tree B-Tree: Introduction of B-Tree Insert Operation in B-Tree Delete Operation in B-Tree Red-Black Tree: Introduction to Red-Black Tree Insertion in Red-Black Tree Deletion in Red-Black Tree Scape Goat Tree and Treap: ScapeGoat Tree | (Introduction and Insertion) Treap (A Randomized Binary Search Tree) Implementation of Search, Insert and Delete in Treap Disjoint Set: Introduction Disjoint Set Union on Trees Union By Rank and Path Compression in Union-Find Algorithm Disjoint Set Data Structure Problems Trie: Introduction Insert and Search Delete Pattern Searching using a Trie of all Suffixes Longest Common Prefix Implement a Phone Directory Weighted Prefix Search Boggle Palindrome pair in an array of words (or strings) How to Implement Reverse DNS Look Up Cache? Segment Tree: Introduction Range minimum query Lazy Propagation Persistent Segment Tree Range Minimum Query (Square Root Decomposition and Sparse Table) Min-Max Range queries in array LCA in a binary tree using RMQ Introduction to Heavy Light Decomposition Reconstructing Segment Tree Longest Common Extension / LCE using Segment Tree Binary Indexed Tree: Binary Indexed Tree or Fenwick Tree Two Dimensional Binary Indexed Tree or Fenwick Tree Binary Indexed Tree : Range Updates and Point Queries Binary Indexed Tree : Range Update and Range Queries proto van Emde Boas Trees | Background and Introduction Suffix Array and Suffix Tree Suffix Array Introduction Suffix Tree Introduction Ukkonen’s Suffix Tree Construction – Part 1 Generalized Suffix Tree Suffix Tree Application 1 – Substring Check K-Dimensional Tree: Search and Insertion in K Dimensional tree Find minimum in K Dimensional Tree Deletion in K Dimensional Tree Some other Data Structure: Palindromic Tree | Introduction & Implementation Ternary Search Tree Interval Tree BK-Tree | Introduction & Implementation Cartesian Tree Sparse Set Gomory-Hu Tree Persistent Data Structure Advanced Lists: Generic Linked List in C Memory efficient Doubly Linked List XOR Linked List | Set 1 XOR Linked List | Set 2 Skip List Self-Organizing List Unrolled Linked List n-ary Tree: Generic Trees (N-ary Tree) Mirror of n-ary Tree Diameter of an N-ary tree Depth of an N-Ary tree Height of n-ary tree if parent array is given Number of ways to traverse an N-ary tree Number of siblings of a given Node in n-ary Tree Next Larger element in n-ary tree Serialize and Deserialize an N-ary Tree DFS for a n-ary tree (acyclic graph) represented as adjacency list Quick Links: Trie Practice Problems Segment Tree Practice Problems Recommended: Learn Data Structure and Algorithms | DSA Tutorial Comment Article Tags: Article Tags: DSA