Topological Sorting - GeeksforGeeks

Source: https://www.geeksforgeeks.org/topological-sorting/

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 Topological Sorting Last Updated : 20 Dec, 2025 Topological sorting for Directed Acyclic Graph (DAG) is a linear ordering of vertices such that for every directed edge u→v, vertex u comes before v in the ordering. There may be several topological orderings for a graph. Applies only to DAGs (Directed Acyclic Graphs), and is not possible for cyclic or undirected graphs. It represents dependency ordering between vertices which is different from normal DFS or BFS traversals. Used in problems like task scheduling, build order, course prerequisite ordering, etc. Example: Input: adj[][] = [[1], [2], [], [2, 4], []] Output: [0, 3, 1, 4, 2] Explanation: Since vertices 0 and 3 do not have any incoming edges from any other vertex, they appear first in the topological ordering. Next, vertex 1 depends only on vertex 0, so it comes after 0. Similarly, vertex 4 depends only on vertex 3, placing it after 3. Finally, vertex 2 depends on both vertices 1 and 3, so it appears after both of them in the ordering. Try it on GfG Practice Topological Sorting in Directed Acyclic Graphs (DAGs) DAGs are graphs where each edge is directed and no cycle exists in the graph. Before understanding why Topological sort only exists for DAGs, lets first answer two questions: Why Topological Sort is not possible for graphs with undirected edges or cycles? Undirected Graph: Every edge u–v can be seen as both u→v and v→u, so you can't decide which one should come first. Cyclic Graph: Let's say the graph contains the cycle, u→v→w→u, then you'd need u before v, v before w, and w before u — impossible to satisfy. Topological order may not be Unique: Topological sort represents all possible ordering satisfying the condition that if there is an edge between u→v, u comes before v in the ordering. Any ordering that satisfies this for all edges(u->v) is valid. Let's see all possible topological orderings for the below graph: For the above image, 4->5->3->1 and 5->4->3->1, both topological orders are valid. Topological Sorting Using DFS: The main idea is to perform a Depth First Search (DFS) on the Directed Acyclic Graph (DAG) and, for each vertex, push it onto a stack only after visiting all its adjacent vertices. This ensures that every vertex appears after all its neighboring vertices. Finally, reversing the stack (or popping elements from it) gives the topological ordering of the graph. Refer to this article to read more. Topological Sorting Using BFS: Topological Sort using BFS (Kahn's Algorithm) works by repeatedly selecting vertices with in-degree zero(no dependencies), adding them to the result, and reducing the in-degree of their adjacent vertices. This process continues until all vertices are processed, producing a valid linear ordering of a DAG. Refer to this article to read more. Advantages and Applications of Topological Sort: Task scheduling and project management. For example, course scheduling in universities. Detects cycles in a directed graph. For example, Deadlock detection in operating systems Efficient for solving problems with precedence constraints. In software deployment tools like Makefile or any other tool for dependency resolution. Disadvantages of Topological Sort: May not be unique, multiple valid topological orderings can exist. Related Articles: Kahn's algorithm for Topological Sorting All Topological Sorts of a Directed Acyclic Graph Topological sort using DFS Comment Article Tags: Article Tags: Graph DSA Microsoft Amazon Morgan Stanley Flipkart Samsung Accolite Moonfrog Labs OYO DFS Topological Sorting + 8 More