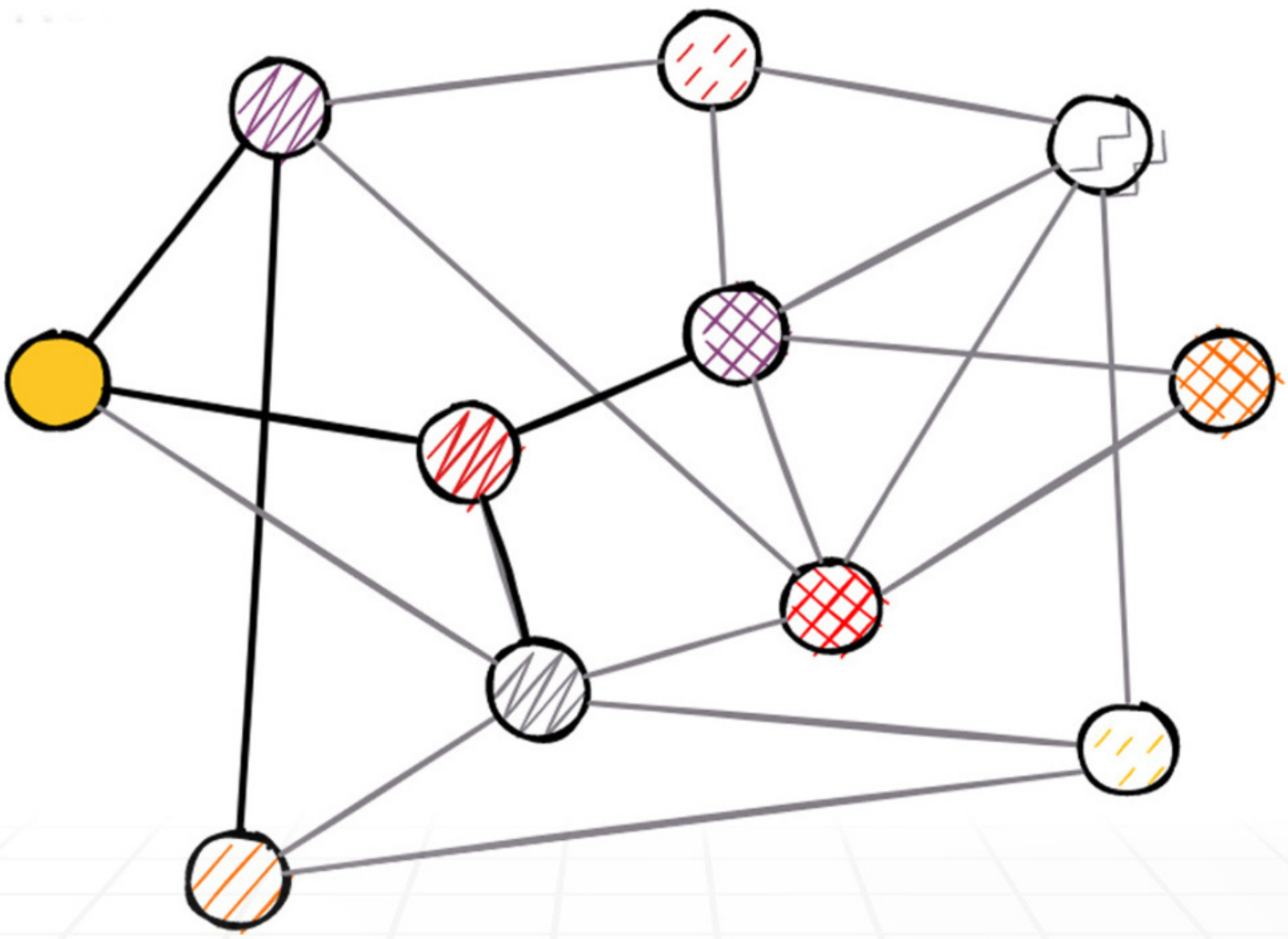
AlgoTutor

MASTER GRAPHS IN JUST 10 DAYS



AlgoTutor

Day 1

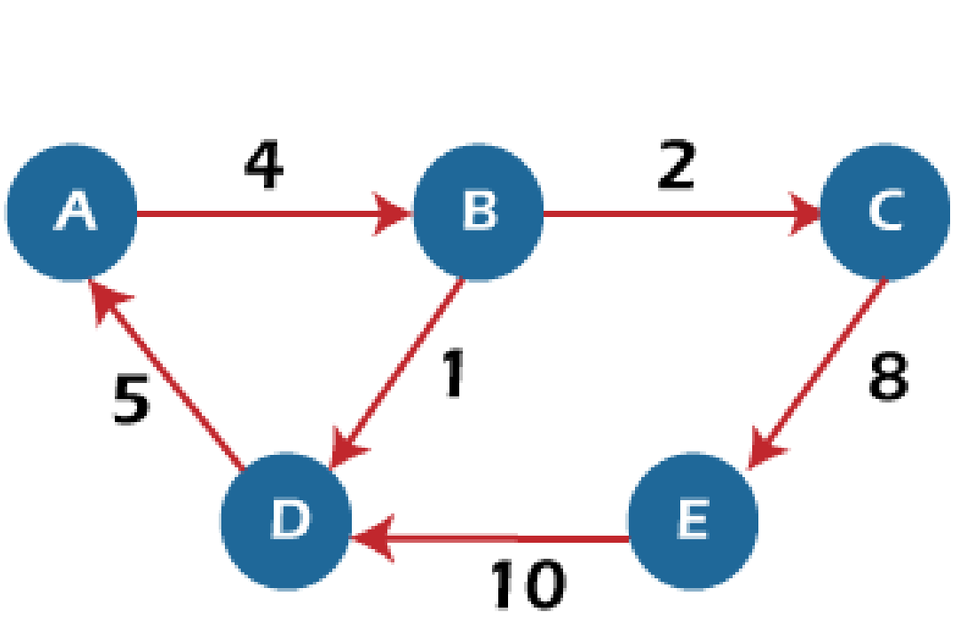
Graph Basics and Terminology

- Understand the fundamental concepts of graphs: vertices, edges, directed, and undirected.
- Learn basic graph terminology and classifications (e.g., connected, disconnected, acyclic, cyclic).

Day 2

Graph Representations

- Study various graph representations: adjacency matrix, adjacency list.
- Understand when to use each representation and their space complexities.



weighted Directed Graph

	A	B	C	D	E
A	0	4	0	0	0
B	0	0	2	1	0
C	0	0	0	0	8
D	5	0	0	0	0
E	0	0	0	10	0

Adjacency Matrix

AlgoTutor

Day 3

Depth-First Search (DFS)

- Learn about depth-first search (DFS) and its recursive and iterative implementations.
- Practice DFS on simple graphs and trees.

Day 4

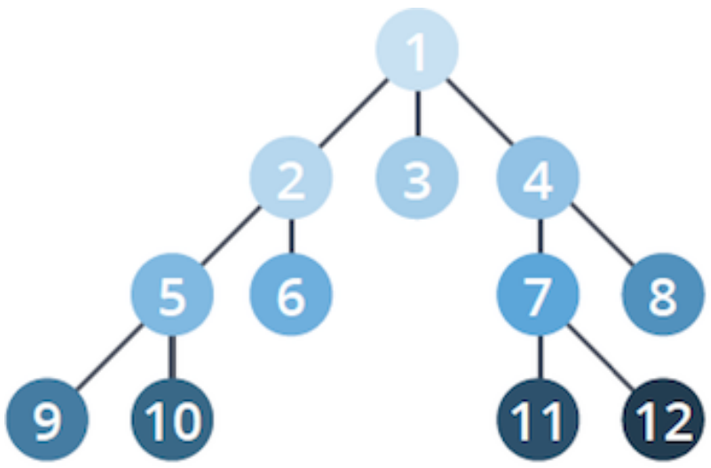
Breadth-First Search (BFS)

- Study breadth-first search (BFS) and its implementation.
- Practice BFS on graphs to explore shortest path problems.

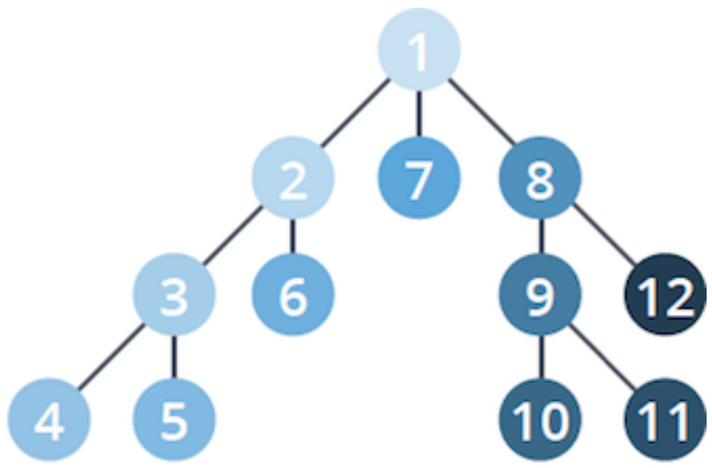
Day 5

Graph Traversal Techniques

- Explore traversal techniques like topological sorting and strongly connected components.
- Understand their applications in directed graphs.



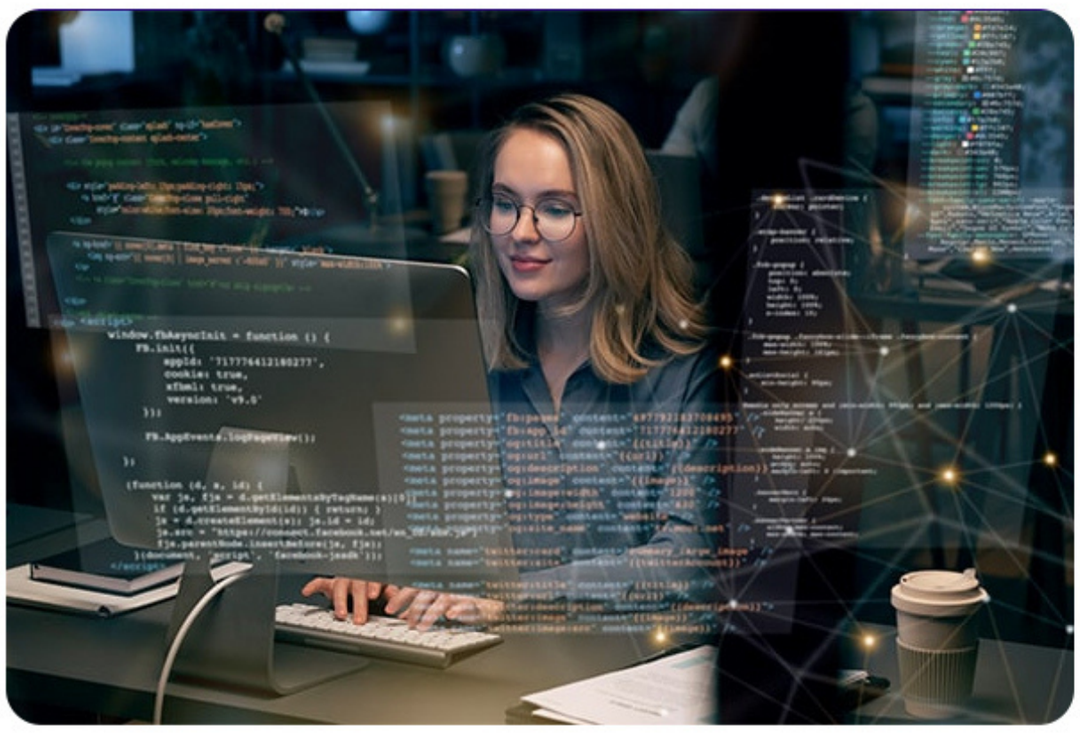
Breadth-first search



Depth-first search

AlgoTutor

Take The First Step Towards Fulfilling a Career



Mastering DSA & System Design

Advance Data Science & ML

Full Stack Web Development - MERN

Mastering DSA & System Design with Full Stack Specialization

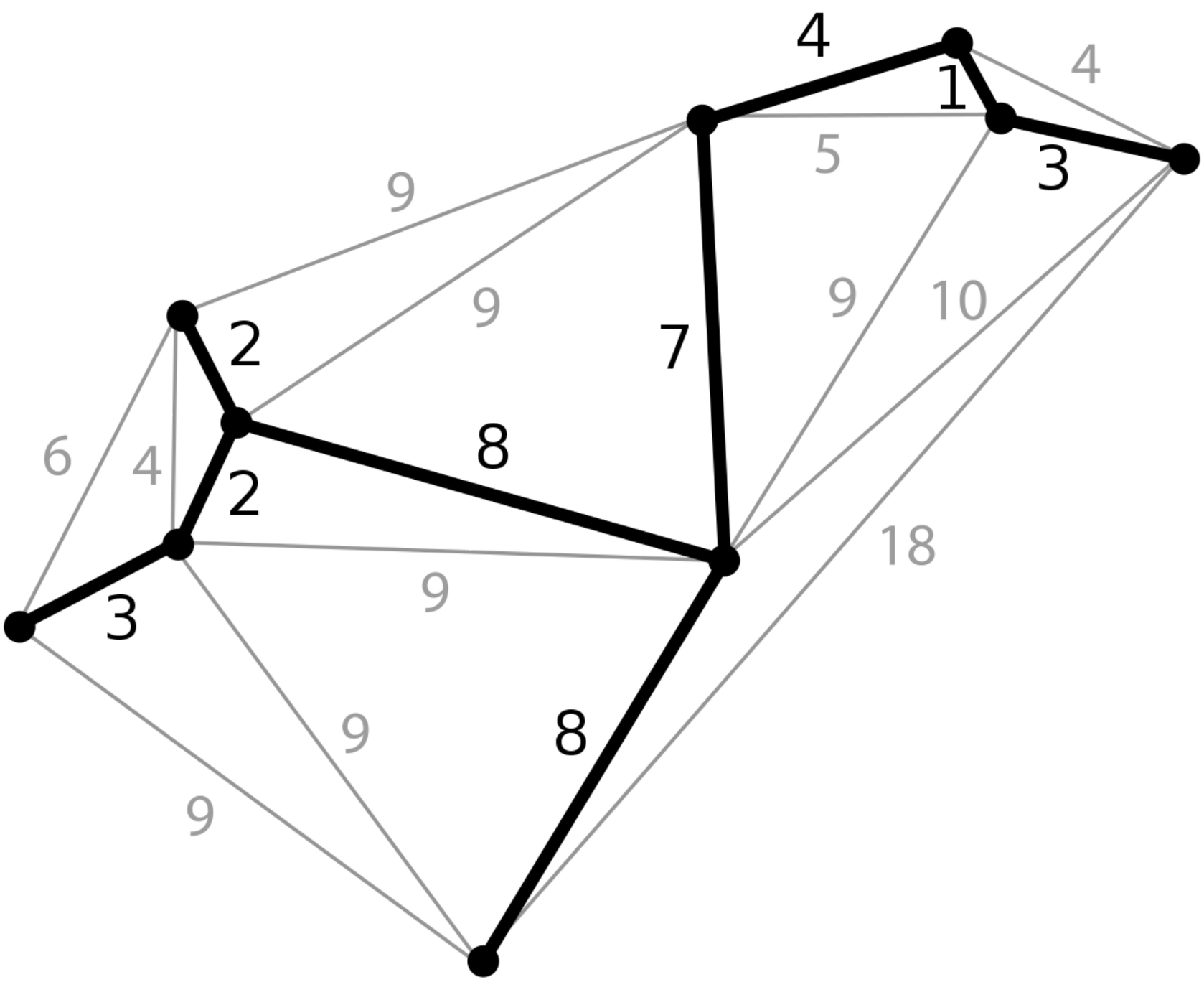
For Admission Enquiry +91 - 72600 58093

AlgoTutor

Day 6

Minimum Spanning Trees

- Study minimum spanning tree algorithms, particularly Prim's and Kruskal's algorithms.
- Practice finding minimum spanning trees in graphs.

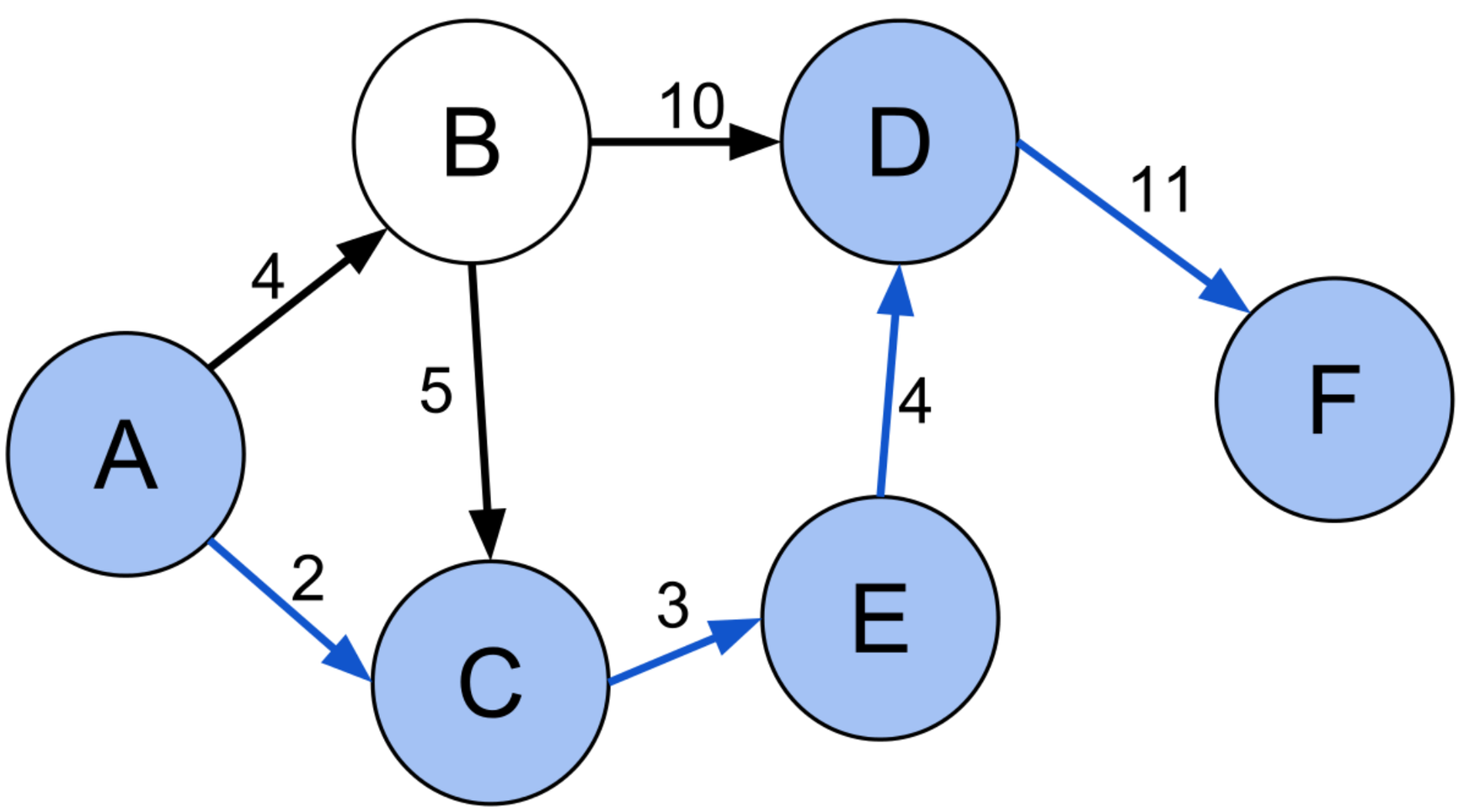


AlgoTutor

Day 7

Shortest Path Algorithms

- Learn about shortest path algorithms, especially Dijkstra's and Bellman-Ford algorithms.
- Solve problems related to finding shortest paths in weighted graphs.



Day 8

Advanced Graph Algorithms

- Study advanced graph algorithms like Floyd-Warshall for all-pairs shortest paths.
- Solve problems that require complex graph analysis.

AlgoTutor

Day 9

Graph Problems in Competitive Programming

- Explore graph problems that frequently appear in coding competitions.
- Practice problems from platforms like Codeforces and AtCoder.

Day 10

Review and Practice

- Review all the graph concepts and problems you've learned.
- Solve more challenging graph problems and solidify your understanding.

AlgoTutor

Important Practice Questions

01. Clone Graph

Given a reference of a node in a connected undirected graph.

Return a deep copy (clone) of the graph.

Each node in the graph contains a value (int) and a list (List[Node]) of its neighbors.

```
class Node {
    public int val;
    public List<Node> neighbors;
}
```

Example 1:

Input: adjList = [[2,4],[1,3],[2,4],[1,3]]

Output: [[2,4],[1,3],[2,4],[1,3]]

Practice

AlgoTutor

02. Course Schedule

There are a total of numCourses courses you have to take, labeled from 0 to numCourses - 1.

You are given an array prerequisites where prerequisites[i] = [ai, bi] indicates that you must take course bi first if you want to take course ai.

- For example, the pair [0, 1], indicates that to take course 0 you have to first take course 1.

Return true if you can finish all courses. Otherwise, return false.

Example 1:

Input: numCourses = 2, prerequisites = [[1,0]]

Output: true

Example 2:

Input: numCourses = 2, prerequisites = [[1,0],[0,1]]

Output: false

Practice

AlgoTutor

03. Number of Islands

Given an m x n 2D binary grid grid which represents a map of '1's (land) and '0's (water), return the number of islands.

An island is surrounded by water and is formed by connecting adjacent lands horizontally or vertically. You may assume all four edges of the grid are all surrounded by water.

Example 1:

Input: grid = [
  ["1","1","1","1","0"],
  ["1","1","0","1","0"],
  ["1","1","0","0","0"],
  ["0","0","0","0","0"]
]

Output: 1

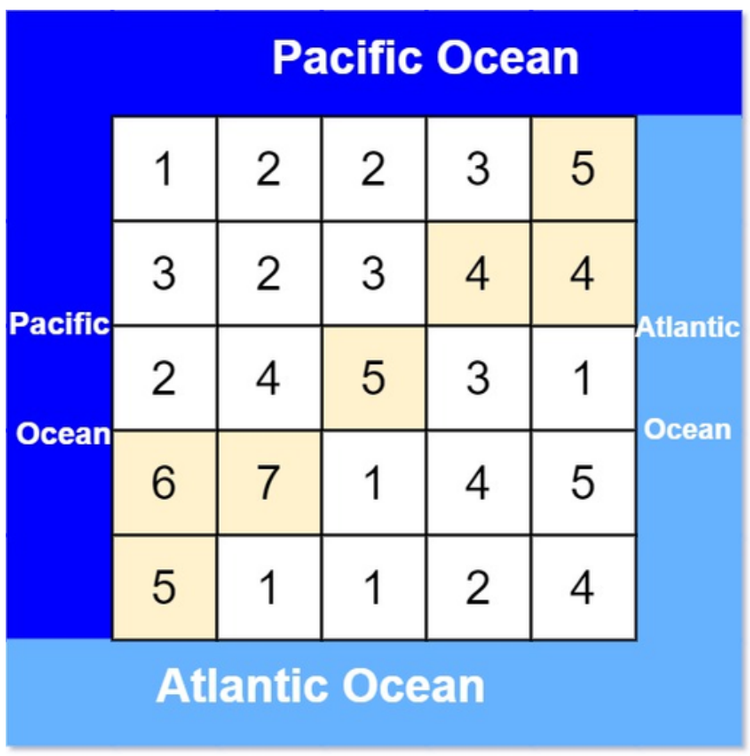
Practice

04. Pacific Atlantic Water Flow

There is an m x n rectangular island that borders both the Pacific Ocean and Atlantic Ocean. The Pacific Ocean touches the island's left and top edges, and the Atlantic Ocean touches the island's right and bottom edges.

The island is partitioned into a grid of square cells. You are given an m x n integer matrix heights where heights[r][c] represents the height above sea level of the cell at coordinate (r, c).

Example 1:



Input: heights = [[1,2,2,3,5],[3,2,3,4,4],[2,4,5,3,1],[6,7,1,4,5],[5,1,1,2,4]]

Output: [[0,4],[1,3],[1,4],[2,2],[3,0],[3,1],[4,0]]

Practice

AlgoTutor

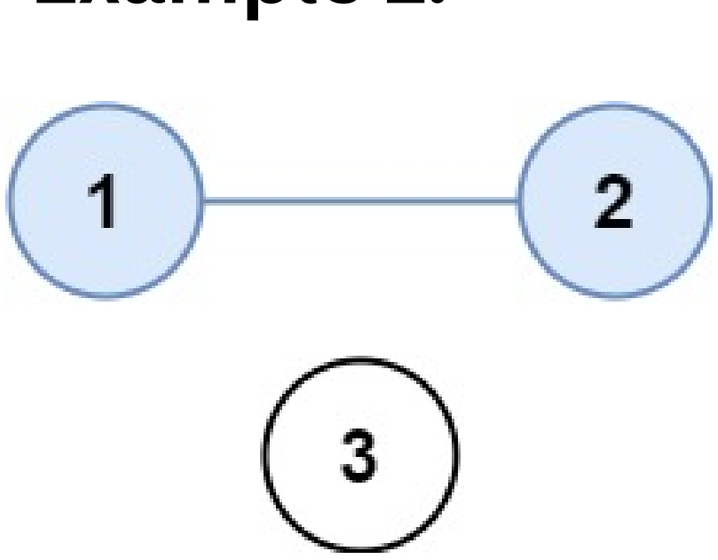
05. Number of Provinces

There are n cities. Some of them are connected, while some are not. If city a is connected directly with city b, and city b is connected directly with city c, then city a is connected indirectly with city c.

A province is a group of directly or indirectly connected cities and no other cities outside of the group.

You are given an n x n matrix isConnected where isConnected[i][j] = 1 if the ith city and the jth city are directly connected, and isConnected[i][j] = 0 otherwise. Return the total number of provinces.

Example 1:



Input: isConnected = [[1,1,0],[1,1,0],[0,0,1]]

Output: 2

Practice

06. Rotting Oranges

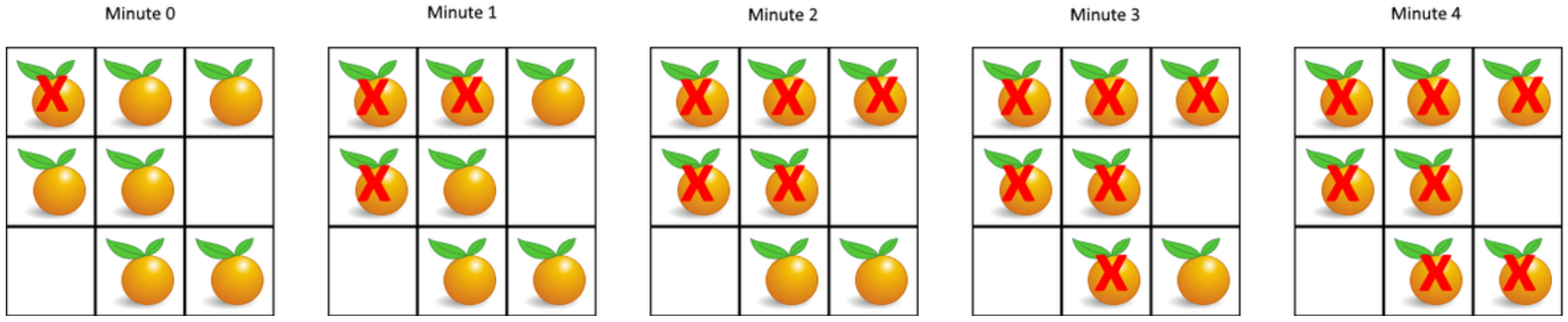
You are given an m x n grid where each cell can have one of three values:

- 0 representing an empty cell,
- 1 representing a fresh orange, or
- 2 representing a rotten orange.

Every minute, any fresh orange that is 4-directionally adjacent to a rotten orange becomes rotten.

Return the minimum number of minutes that must elapse until no cell has a fresh orange. If this is impossible, return -1.

Example 1:



Input: grid = [[2,1,1],[1,1,0],[0,1,1]]

Output: 4

Practice

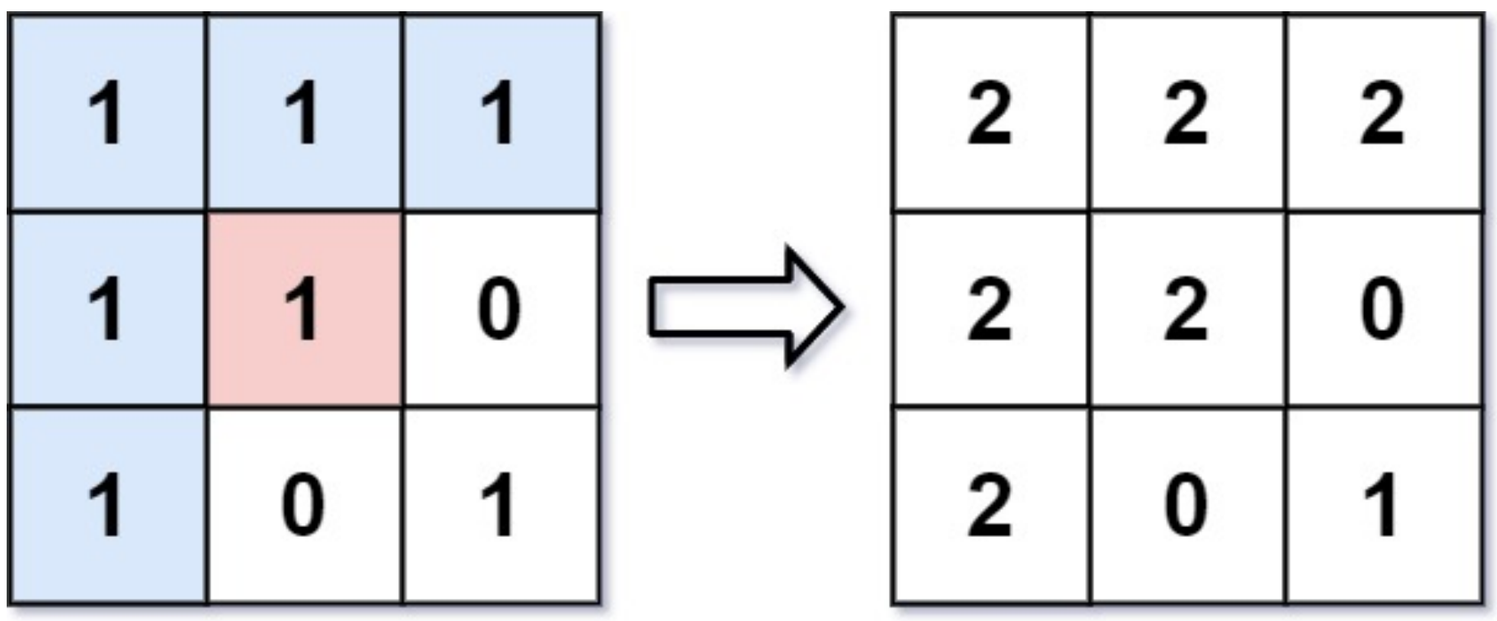
07. Flood Fill

An image is represented by an m x n integer grid image where image[i][j] represents the pixel value of the image. You are also given three integers sr, sc, and color.

You should perform a flood fill on the image starting from the pixel image[sr][sc].

Return the modified image after performing the flood fill.

Example 1:



Input: image = [[1,1,1],[1,1,0],[1,0,1]], sr = 1, sc = 1, color = 2

Output: [[2,2,2],[2,2,0],[2,0,1]]

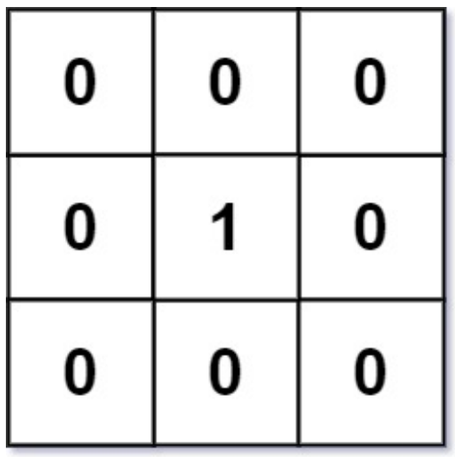
Practice

08. 01 Matrix

Given an m x n binary matrix mat, return the distance of the nearest 0 for each cell.

The distance between two adjacent cells is 1.

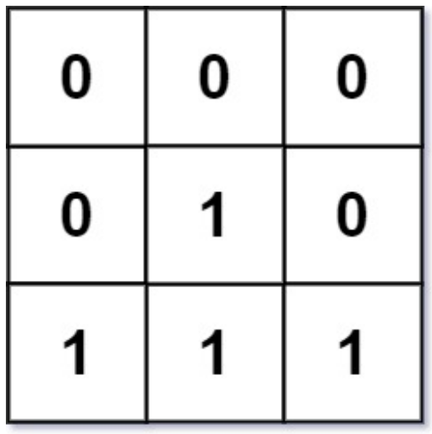
Example 1:



Input: mat = [[0,0,0],[0,1,0],[0,0,0]]

Output: [[0,0,0],[0,1,0],[0,0,0]]

Example 2:



Input: mat = [[0,0,0],[0,1,0],[1,1,1]]

Output: [[0,0,0],[0,1,0],[1,2,1]]

Practice

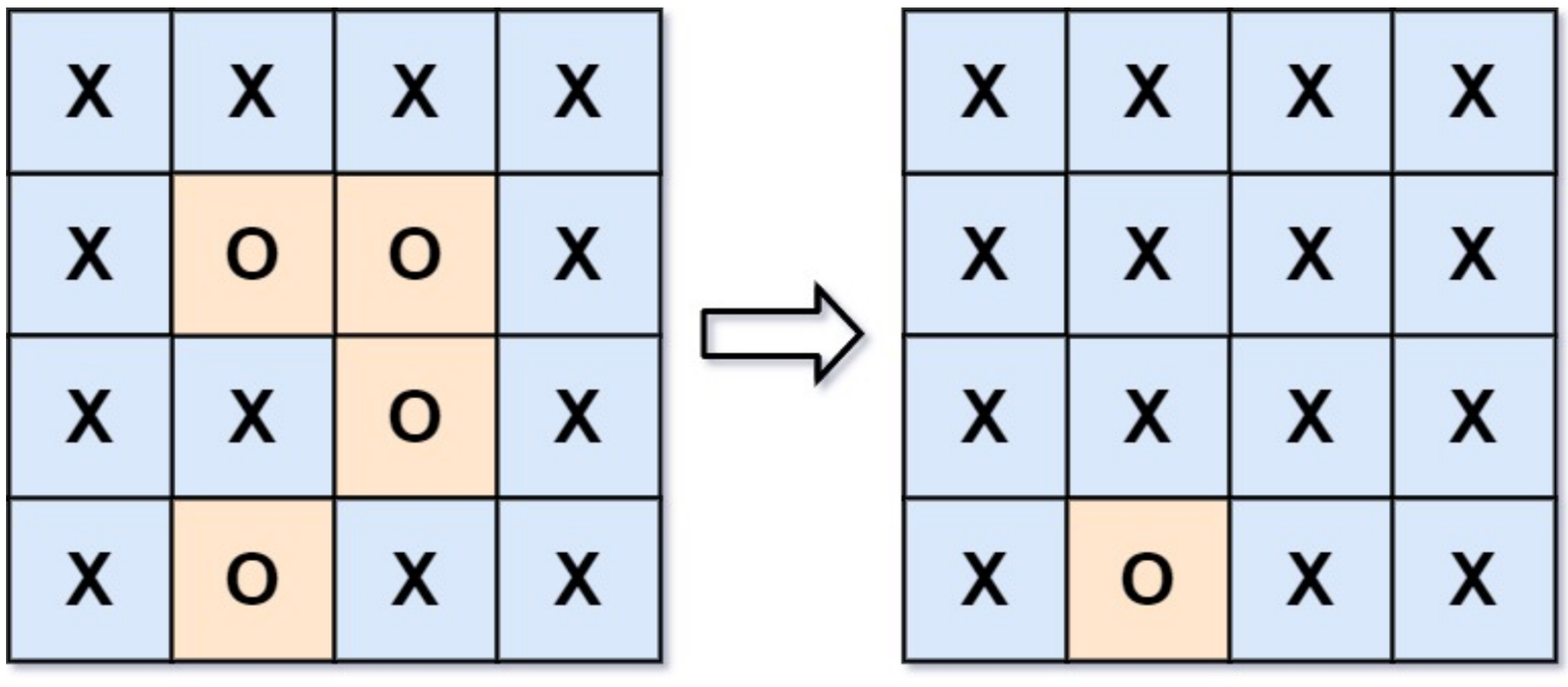
AlgoTutor

09. Surrounded Regions

Given an m x n matrix board containing 'X' and 'O', capture all regions that are 4-directionally surrounded by 'X'.

A region is captured by flipping all 'O's into 'X's in that surrounded region.

Example 1:



Input: board = [["X","X","X","X"],["X","O","O","X"],
["X","X","O","X"],["X","O","X","X"]]

Output: [["X","X","X","X"],["X","X","X","X"],["X","X","X","X"],
["X","O","X","X"]]

Practice

AlgoTutor

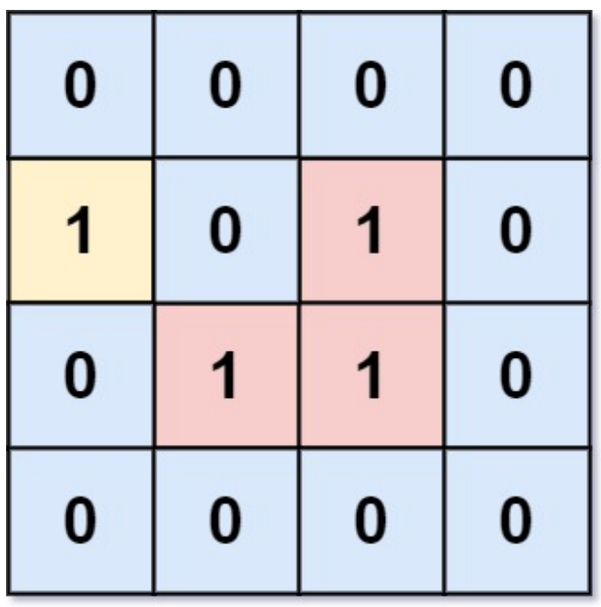
10. Number of Enclaves

You are given an m x n binary matrix grid, where 0 represents a sea cell and 1 represents a land cell.

A move consists of walking from one land cell to another adjacent (4-directionally) land cell or walking off the boundary of the grid.

Return the number of land cells in grid for which we cannot walk off the boundary of the grid in any number of moves.

Example 1:



Input: grid = [[0,0,0,0],[1,0,1,0],[0,1,1,0],[0,0,0,0]]

Output: 3

Practice

11. Word Ladder

A transformation sequence from word beginWord to word endWord using a dictionary wordList is a sequence of words beginWord -> s1 -> s2 -> ... -> sk such that:

- Every adjacent pair of words differs by a single letter.
- Every si for 1 <= i <= k is in wordList. Note that beginWord does not need to be in wordList.
- sk == endWord

Given two words, beginWord and endWord, and a dictionary wordList, return the number of words in the shortest transformation sequence from beginWord to endWord, or 0 if no such sequence exists.

Example 1:

Input: beginWord = "hit", endWord = "cog", wordList = ["hot","dot","dog","lot","log","cog"]

Output: 5

Practice

12. Word Ladder II

AlgoTutor

A transformation sequence from word beginWord to word endWord using a dictionary wordList is a sequence of words beginWord -> s1 -> s2 -> ... -> sk such that:

- Every adjacent pair of words differs by a single letter.
- Every si for 1 <= i <= k is in wordList. Note that beginWord does not need to be in wordList.
- sk == endWord

Given two words, beginWord and endWord, and a dictionary wordList, return all the shortest transformation sequences from beginWord to endWord, or an empty list if no such sequence exists. Each sequence should be returned as a list of the words [beginWord, s1, s2, ..., sk].

Example 1:

Input: beginWord = "hit", endWord = "cog", wordList = ["hot","dot","dog","lot","log","cog"]

Output: [["hit","hot","dot","dog","cog"], ["hit","hot","lot","log","cog"]]

Practice

13. Is Graph Bipartite?

There is an undirected graph with n nodes, where each node is numbered between 0 and n - 1. You are given a 2D array graph, where graph[u] is an array of nodes that node u is adjacent to. More formally, for each v in graph[u], there is an undirected edge between node u and node v. The graph has the following properties:

- There are no self-edges (graph[u] does not contain u).
- There are no parallel edges (graph[u] does not contain duplicate values).
- The graph may not be connected, meaning there may be two nodes u and v such that there is no path between them.

A graph is bipartite if the nodes can be partitioned into two independent sets A and B such that every edge in the graph connects a node in set A and a node in set B.

Return true if and only if it is bipartite.

Example 1:

Input: graph = [[1,2,3],[0,2],[0,1,3],[0,2]]

Output: false

Practice

AlgoTutor

14. Course Schedule II

There are a total of numCourses courses you have to take, labeled from 0 to numCourses - 1. You are given an array prerequisites where prerequisites[i] = [ai, bi] indicates that you must take course bi first if you want to take course ai.

- For example, the pair [0, 1], indicates that to take course 0 you have to first take course 1.

Return the ordering of courses you should take to finish all courses. If there are many valid answers, return any of them. If it is impossible to finish all courses, return an empty array.

Example 1:

Input: numCourses = 2, prerequisites = [[1,0]]

Output: [0,1]

Example 2:

Input: numCourses = 4, prerequisites = [[1,0],[2,0],[3,1],[3,2]]

Output: [0,2,1,3]

Practice

15. Shortest Path in Binary Matrix

Given an n x n binary matrix grid, return the length of the shortest clear path in the matrix. If there is no clear path, return -1.

A clear path in a binary matrix is a path from the top-left cell (i.e., (0, 0)) to the bottom-right cell (i.e., (n - 1, n - 1)) such that:

- All the visited cells of the path are 0.
- All the adjacent cells of the path are 8-directionally connected (i.e., they are different and they share an edge or a corner).

The length of a clear path is the number of visited cells of this path.

Example 1:

Input: grid = [[0,1],[1,0]]

Output: 2

Example 2:

Input: grid = [[0,0,0],[1,1,0],[1,1,0]]

Output: 4

Practice

16. Path With Minimum Effort

AlgoTutor

You are a hiker preparing for an upcoming hike. You are given heights, a 2D array of size rows x columns, where heights[row][col] represents the height of cell (row, col).

You are situated in the top-left cell, (0, 0), and you hope to travel to the bottom-right cell, (rows-1, columns-1) (i.e., 0-indexed). You can move up, down, left, or right, and you wish to find a route that requires the minimum effort.

A route's effort is the maximum absolute difference in heights between two consecutive cells of the route.
Return the minimum effort required to travel from the top-left cell to the bottom-right cell.

Example 1:

Input: heights = [[1,2,2],[3,8,2],[5,3,5]]

Output: 2

Example 2:

Input: heights = [[1,2,3],[3,8,4],[5,3,5]]

Output: 1

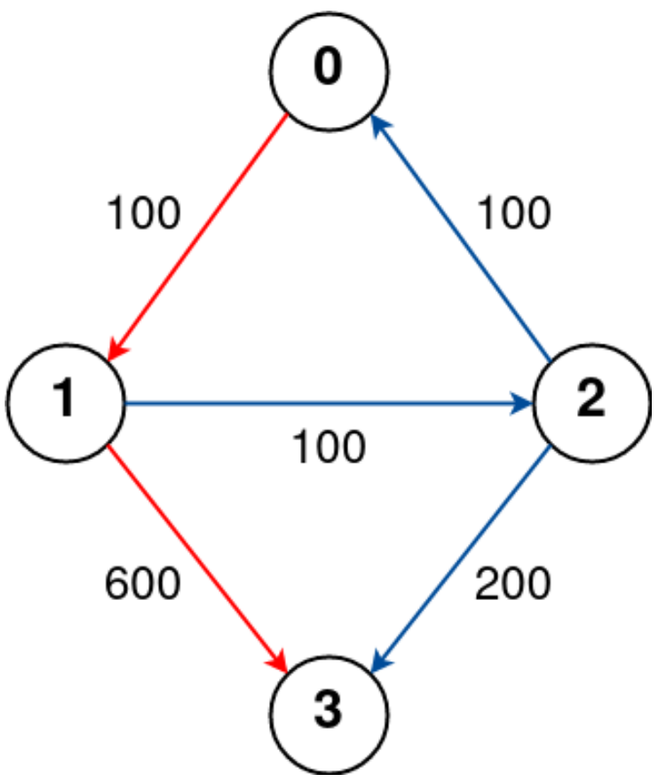
Practice

17. Cheapest Flights Within K Stops

There are n cities connected by some number of flights. You are given an array flights where flights[i] = [fromi, toi, pricei] indicates that there is a flight from city fromi to city toi with cost pricei.

You are also given three integers src, dst, and k, return the cheapest price from src to dst with at most k stops. If there is no such route, return -1.

Example 1:



Input: n = 4, flights = [[0,1,100],[1,2,100],[2,0,100],[1,3,600],[2,3,200]], src = 0, dst = 3, k = 1

Output: 700

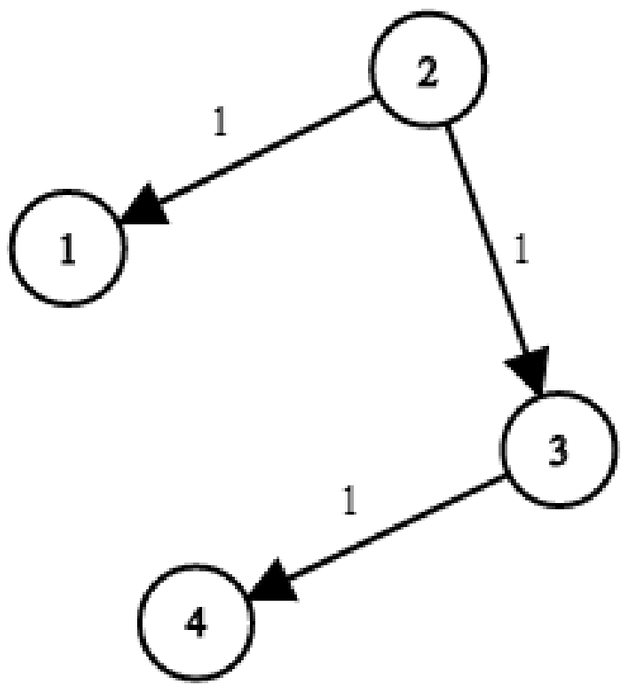
Practice

18. Network Delay Time

You are given a network of n nodes, labeled from 1 to n. You are also given times, a list of travel times as directed edges times[i] = (ui, vi, wi), where ui is the source node, vi is the target node, and wi is the time it takes for a signal to travel from source to target.

We will send a signal from a given node k. Return the minimum time it takes for all the n nodes to receive the signal. If it is impossible for all the n nodes to receive the signal, return -1.

Example 1:



Input: times = [[2,1,1],[2,3,1],[3,4,1]], n = 4, k = 2

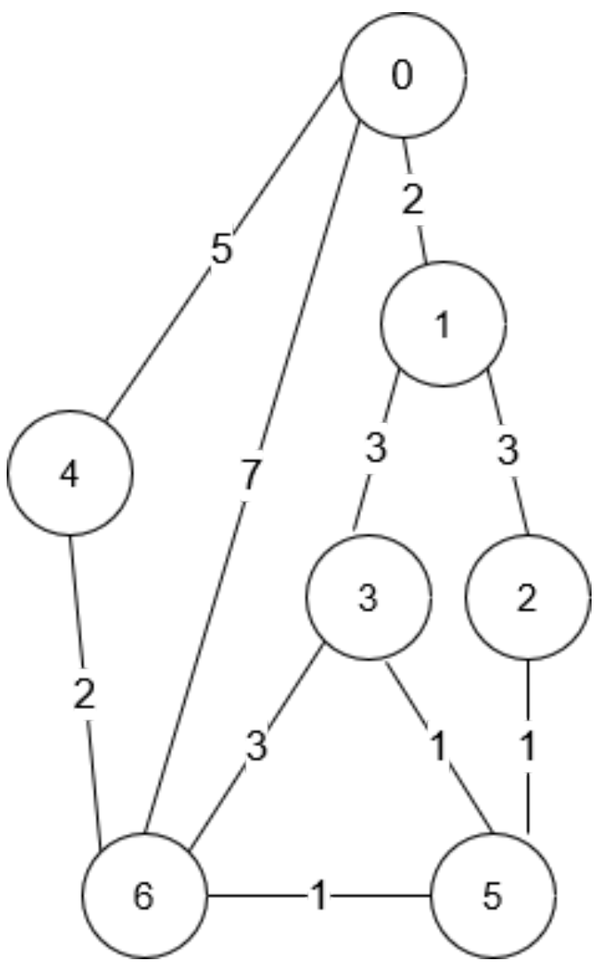
Output: 2

Practice

19. Number of Ways to Arrive at Destination

You are in a city that consists of n intersections numbered from 0 to n - 1 with bi-directional roads between some intersections. The inputs are generated such that you can reach any intersection from any other intersection and that there is at most one road between any two intersections.

Example 1:



Input: n = 7, roads = [[0,6,7],[0,1,2],[1,2,3],[1,3,3],[6,3,3],[3,5,1],[6,5,1],[2,5,1],[0,4,5],[4,6,2]]

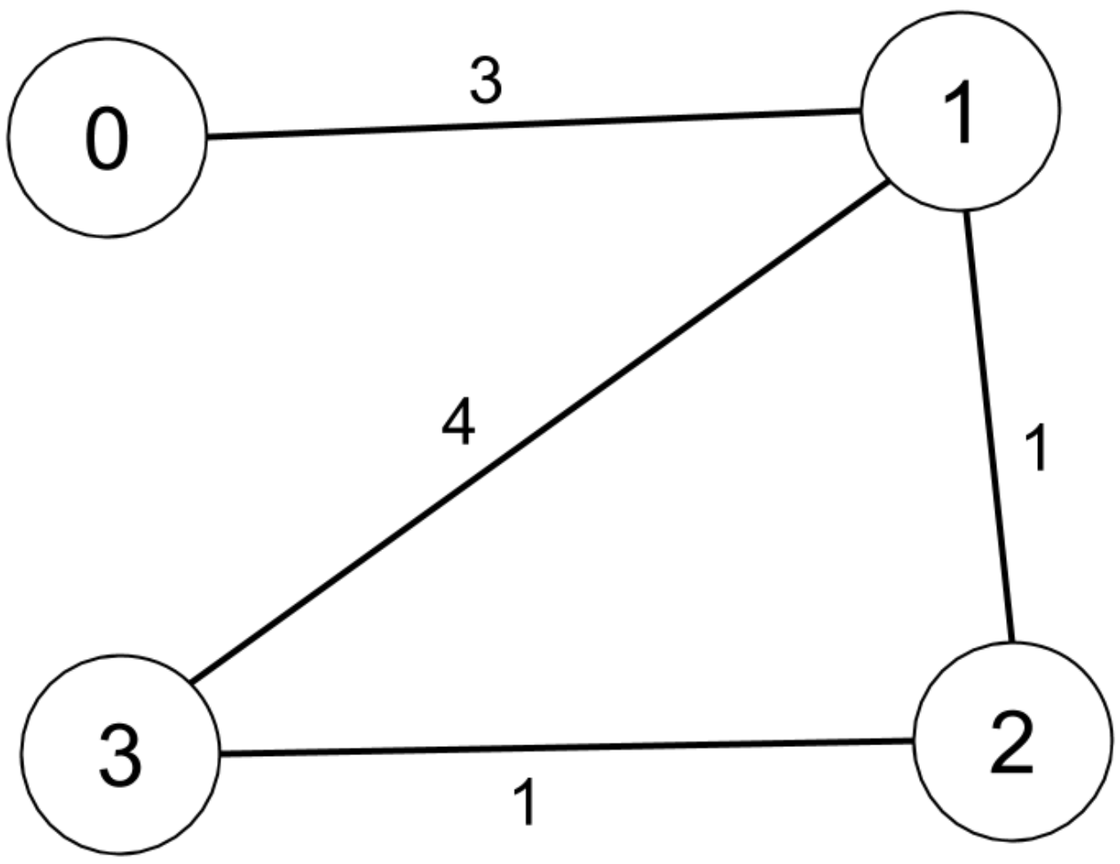
Output: 4

Practice

20. Find the City With the Smallest Number of Neighbors at a Threshold Distance

There are n cities numbered from 0 to n-1. Given the array edges where edges[i] = [fromi, toi, weighti] represents a bidirectional and weighted edge between cities fromi and toi, and given the integer distanceThreshold.

Example 1:



Input: n = 4, edges = [[0,1,3],[1,2,1],[1,3,4],[2,3,1]], distanceThreshold = 4

Output: 3

Practice

AlgoTutor

WHY ALGOTUTOR

- 100% Placement Assistance
- 1-1 personal mentorship from Indusrty experts
- 200+ Successful Alumni
- 147(Avg.)% Salary Hike
- 100% Success Rate
- 23 LPA (Avg.) CTC
- Learn from scratch
- Career Services

For Admission Enquiry

+91-7260058093

info@algotutor.io