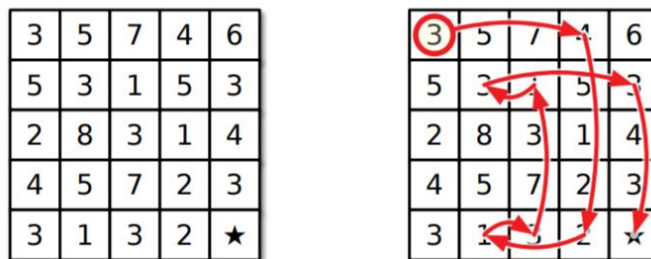


Problems

1. A graph $G = (V, E)$ is *bipartite* if the vertices V can be partitioned into two subsets L and R , such that every edge has one vertex in L and the other in R .
 - (a) Prove that every tree is a bipartite graph.
 - (b) Describe and analyze an efficient algorithm that determines whether a given undirected graph is bipartite.
2. Let G be a connected graph, and let T be a depth-first spanning tree of G rooted at some node v . Prove that if T is also a breadth-first spanning tree of G rooted at v , then $G = T$.
3. In class we saw that to prove the correctness of Dijkstra's algorithm we need to assert that all edges weights in the graph are nonnegative. In this problem we examine this assumption closely.
 - (a) Consider the following algorithm for finding shortest paths in graphs with negative edge weights: add a large constant to each edge weight so that all the weights become positive; then run Dijkstra's algorithm. Is this algorithm correct? Prove your answer.
 - (b) Suppose that the source node s has no incoming edges, and all of the negative weight edges are those that leave s (so, in particular, there are no negative cost cycles). Can Dijkstra's algorithm, started at s , fail on such a graph? Prove your answer.
4. A **number maze** is an $n \times n$ grid of positive integers. A token starts in the upper-left corner; your goal is to move the token to the lower-right corner. On each turn, you are allowed to move the token up, down, left, or right; the distance you may move the token is determined by the number on its current square. For example, if the token is on a square labeled 3, then you may move the token three steps up, three steps down, three steps left, or three steps right. However, you are never allowed to move the token off the edge of the board.

Describe and analyze an efficient algorithm that either returns the minimum number of moves required to solve a given number maze, or correctly reports that the maze has no solution.



A 5×5 number maze that can be solved in eight moves.

5. Whenever groups of pigeons gather, they instinctively establish a *pecking order*. For any pair of pigeons, one pigeon always pecks the other. Surprisingly, the overall pecking order can contain cycles—for example, pigeon A pecks pigeon B , which pecks pigeon C , which pecks pigeon A .
 - (a) Prove that any finite set of pigeons can be arranged in a row from left to right so that every pigeon pecks the pigeon immediately to its left.
 - (b) Suppose you are given a directed graph representing the pecking relationships among a set of n pigeons. The graph contains one vertex per pigeon, and an edge $i \rightarrow j$ if and only if pigeon i pecks pigeon j . Describe and analyze an algorithm to compute a pecking order for the pigeons, as guaranteed by part (a).