Chapter 1

# Other Graph Algorithms

Now that you are familiar with Dijkstra's algorithm for finding the shortest path in a graph, you are well-equipped to understand more graph algorithms. This document will discuss two other important algorithms: Depth-First Search (DFS) and the Bellman-Ford algorithm.

## 1.1 Depth-First Search

Depth-First Search (DFS) is an algorithm for traversing or searching tree or graph data structures. DFS uses a stack (or sometimes recursion, which uses the system stack implicitly) to explore the graph in a depthward motion until it hits a node with no unvisited adjacent nodes, then it backtracks.

The procedure is as follows:

1. Push the root node into the stack.
2. Pop a node from the stack, and mark it as visited.
3. Push all unvisited adjacent nodes into the stack.
4. Repeat steps 2 and 3 until the stack is empty.

DFS is particularly useful for solving problems like connected-component detection in graphs and maze-solving.

## 1.2 Bellman-Ford Algorithm

The Bellman-Ford algorithm is another shortest path algorithm like Dijkstra's. However, unlike Dijkstra's algorithm, Bellman-Ford can handle graphs with negative weight edges.

The algorithm works as follows:

1. Assign a tentative distance value for every node: set it to zero for our initial node and to infinity for all other nodes.
2. For each edge $(u, v)$ with weight $w$, if the current distance to $v$ is greater than the

distance to $u$ plus $w$, update the distance to $v$ to be the distance to $u$ plus $w$.

3. Repeat the previous step $|V| - 1$ times, where $|V|$ is the number of vertices in the graph.
4. After the above steps, if you can still find a shorter path, there exists a negative cycle.

If the graph does not contain a negative cycle reachable from the source, the shortest paths are well-defined, and Bellman-Ford will correctly calculate them. If a negative cycle is reachable, no solution exists, but Bellman-Ford will detect it.

---

Appendix A

# Answers to Exercises

# Index