Graph Algorithms Comparison

Here's a comparison of **BFS, DFS, Dijkstra, Bellman-Ford, and Warshall's algorithm**, along with their applications:

Algorithm	Type	Time Complexity	Space Complexity	Applications
BFS (Breadth-First Search)	Graph Traversal	O(V + E)	O(V)	- Finding the shortest path in an unweighted graph - Level-order traversal in trees - Web crawling - Social networking sites (finding friends at a certain distance)
DFS (Depth-First Search)	Graph Traversal	O(V + E)	O(V)	- Detecting cycles in graphs - Topological sorting - Solving puzzles (like mazes) - Pathfinding in AI
Dijkstra	Shortest Path	O((V + E) log V) with a priority queue	O(V)	- Finding the shortest path in weighted graphs (non-negative weights) - GPS navigation systems - Network routing protocols (like OSPF)
Bellman-Ford	Shortest Path	O(V × E)	O(V)	- Finding shortest paths with negative weights - Detecting negative weight cycles - Used in routing algorithms like RIP
Warshall (Floyd-Warshall)	All-Pairs Shortest Path	$O(V^3)$	$O(V^2)$	- Finding shortest paths between all pairs of vertices - Network analysis

Algorithm	Type	Time Complexity	Space Complexity	Applications
				- Transitive closure of a graph

Key Differences:

1. **Graph Type**:
   - **BFS/DFS**: Works on both directed and undirected graphs.
   - **Dijkstra**: Works only with **non-negative** weighted graphs.
   - **Bellman-Ford**: Handles **negative** weights and detects negative cycles.
   - **Warshall**: Finds shortest paths between **all pairs** of nodes.
2. **Efficiency**:
   - **BFS/DFS**: Fastest for simple traversal.
   - **Dijkstra**: Faster for single-source shortest paths in dense graphs.
   - **Bellman-Ford**: Slower but handles negative weights.
   - **Warshall**: Inefficient for large graphs due to cubic complexity.
3. **Use Case**:
   - Use **BFS** for the shortest path in **unweighted** graphs.
   - Use **DFS** for exploring all paths or checking cycles.
   - Use **Dijkstra** for efficient shortest paths in **non-negative weighted** graphs.
   - Use **Bellman-Ford** for graphs with **negative weights**.
   - Use **Warshall** when you need **all pairs shortest paths**.

This should give you a clear understanding of their differences and where each is best applied!