**NOTES**

1. What is Dynamic Programming?

Dynamic programming is a powerful technique for solving optimization problems. It can be used to solve a wide variety of problems, including the knapsack problem, the shortest path problem, and the longest common subsequence problem. Dynamic programming is based on the principle of optimality, which states that an optimal solution to a problem can be constructed from optimal solutions to its subproblems.

1. **Principle of Optimality**

The principle of optimality is the key to dynamic programming. It states that an optimal solution to a problem can be constructed from optimal solutions to its subproblems. This means that we can solve a problem by breaking it down into smaller subproblems, solving the subproblems, and then combining the solutions to the subproblems to get the solution to the original problem.

1. **Applications of Dynamic Programming**

Dynamic programming has a wide variety of applications, including optimization problems, computer science, economics, and finance. In computer science, dynamic programming is used to solve a variety of problems, such as the knapsack problem, the shortest path problem, and the longest common subsequence problem. In economics, dynamic programming is used to solve problems such as optimal allocation of resources and optimal pricing.

**Objective:** Find the path with the minimum total weight between two specified nodes in a graph.

**Constraints:**

The graph may contain directed or undirected edges.

Edge weights represent the distance or cost of traversing an edge.

The path must connect the source node to the destination node.

* Define states and subproblems.
* Recursively compute shortest distances for subproblems
* Construct the shortest path from the shortest distances.
* Trace back through the shortest distances to identify the path nodes.
* The shortest path consists of the nodes that contribute to the minimum total distance from the source node to the destination node.
* Navigation systems: Finding the fastest route between two locations on a map.
* Network optimization: Optimizing data routing and traffic flow in computer networks.
* Transportation planning: Designing efficient transportation systems and minimizing travel times.