

10. Dynamic Programming:

Dynamic programming solves complex problems by breaking them down into smaller overlapping subproblems.

It uses techniques like memoization (storing results of expensive function calls) and tabulation (filling a table of values) for efficient computation.

Example: The knapsack problem, where you aim to maximize the value of items in a knapsack with limited capacity, can be solved using dynamic programming.

11. Hashing:

Hashing maps data to a fixed-size array using a hash function. It provides efficient retrieval and insertion of elements.

Hash tables, which are often used for hashing, store key-value pairs and handle collisions (when two elements map to the same position) using techniques like chaining or open addressing.

Example: Storing a phone book as a hash table, where names are the keys and corresponding phone numbers are the values.

Please note that these descriptions still provide a high-level overview. Each concept encompasses more details, algorithms, and

variations that can be explored in depth through dedicated resources on Data Structures and Algorithms.

preview from Notesale.co.uk
Page 4 of 4