Greedy Algorithms

Greedy Algorithms

1. **Definition**
    - An algorithmic approach that makes the locally optimal choice at each step, hoping this leads to a globally optimal solution.
2. **When Greedy Works**
    - Effective when a problem has the **greedy choice property** (a global optimum can be reached by local choices) and **optimal substructure**.
3. **Classic Examples**
    - **Activity Selection**: choosing the maximum number of non-overlapping activities.
    - **Huffman Coding**: building an optimal prefix-free encoding for data compression.
    - **Dijkstra's Algorithm**: finding shortest paths in a weighted graph with non-negative edges.
4. **Limitations**
    - Greedy algorithms don't always produce optimal solutions - they can fail on problems like the 0/1 Knapsack Problem, where dynamic programming is required instead.
5. **Why They're Useful**
    - When applicable, greedy algorithms are typically simpler and faster than alternatives like dynamic programming, since they avoid exploring multiple solution paths.