

7 Lecture - CS502

Important Subjective

1. **What is a greedy algorithm, and what is its basic approach?**

Answer: A greedy algorithm is an algorithmic paradigm that follows the problem-solving heuristic of making the locally optimal choice at each stage with the hope of finding a global optimum. The basic approach of a greedy algorithm is to make the best possible choice at each step without considering the overall effect on the solution.

What is the difference between a greedy algorithm and a dynamic programming algorithm?

Answer: In a greedy algorithm, we make the locally optimal choice at each stage without considering the overall effect on the solution. In contrast, dynamic programming is a bottom-up approach that breaks the problem into smaller subproblems, solves each subproblem once, and saves the result for future use.

What is the fractional knapsack problem, and how can it be solved using a greedy algorithm?

Answer: The fractional knapsack problem is a variation of the classic knapsack problem, where the items can be divided into smaller pieces. A greedy algorithm can be used to solve the fractional knapsack problem by sorting the items by their value-to-weight ratio and adding them to the knapsack until it is full.

Can a greedy algorithm be used to solve the minimum spanning tree problem? If so, how?

Answer: Yes, a greedy algorithm can be used to solve the minimum spanning tree problem. The most commonly used algorithm for this is Prim's algorithm, which starts with an arbitrary vertex and repeatedly adds the edge with the smallest weight that connects a vertex in the tree to one outside the tree.

What is the Huffman coding algorithm, and how does it work?

Answer: Huffman coding is a lossless data compression algorithm that uses variable-length codes to represent data. The algorithm works by building a Huffman tree based on the frequency of occurrence of each symbol in the input. The tree is then used to generate the variable-length codes for each symbol.

What is the job sequencing problem, and how can it be solved using a greedy algorithm?

Answer: The job sequencing problem is a scheduling problem where a set of jobs with different deadlines and profits needs to be scheduled on a single machine. A greedy algorithm can be used to solve the job sequencing problem by sorting the jobs by their profits in descending order and scheduling them in the order of their sorted list.

What is the coin change problem, and how can it be solved using a greedy algorithm?

Answer: The coin change problem is a classic problem where a given sum of money needs to be paid using the minimum number of coins. A greedy algorithm can be used to solve the coin change problem by repeatedly selecting the largest coin denomination that is less than or equal

to the remaining amount.

Can a greedy algorithm be used to solve the traveling salesman problem? If not, why not?

Answer: No, a greedy algorithm cannot be used to solve the traveling salesman problem because it does not have the optimal substructure property required for a greedy algorithm to work.

What is the knapsack problem, and how can it be solved using a greedy algorithm?

Answer: The knapsack problem is a classic optimization problem where a set of items with different weights and values needs to be packed into a knapsack with a limited capacity. A greedy algorithm can be used to solve the knapsack problem by sorting the items by their value-to-weight ratio and adding them to the knapsack until it is full.

What is the difference between a greedy algorithm and a backtracking algorithm?

Answer: In a greedy algorithm, we make the locally optimal choice at each stage without considering the overall effect on the solution. In contrast, backtracking is a recursive approach that explores all possible solutions to a problem and backtracks when a dead end is reached.