CALIFORNIA STATE UNIVERSITY
FULLERTON

CPSC 335: Algorithm Engineering

Department of Computer Science, College of Engineering and Computer Science

Instructor: Dr. Shah, Adjunct Faculty, Computer Science https://www.hassanonline.us/

Lecture # 3 Notes: Introduction to Greedy Algorithms

Greedy algorithms are a type of algorithmic strategy designed to solve problems by making the most advantageous choice at any given moment. Imagine you're faced with making decisions in stages, and at each stage, you choose the option that offers the most immediate benefit, hoping that these local optimal choices will lead to a globally optimal solution.

Why Do We Need Greedy Algorithms?

1. **Simplicity and Efficiency**: Greedy algorithms are straightforward to understand and easy to implement. They can provide efficient solutions to complex problems without the need for exhaustive searches.
2. **Real-Time Decision Making**: In many real-world scenarios, decisions need to be made quickly with the information available at the moment. Greedy algorithms excel in these environments by offering a method to make immediate, optimal choices.
3. **Optimization Problems**: They are particularly useful in optimization problems, where the goal is to find the best solution among many possible options, such as minimizing costs or maximizing profits.

Principles of Greedy Algorithms

Greedy algorithms operate on two main principles:

- **Greedy Choice Property**: The choice made at each step should be the best or most optimal, given any particular situation without worrying about future consequences.
- **Optimal Substructure**: A problem has an optimal substructure if an optimal solution to the entire problem contains optimal solutions to its sub-problems.

Examples and Real-World Applications

1. Coin Change Problem

- **Simple Explanation**: Suppose you need to give change for a certain amount of money using the fewest coins possible. A greedy algorithm approach would start by selecting the highest-value coin that does not exceed the remaining amount and repeat this process until the change is made completely.
- **Real-World Application**: Vending machines use this algorithm to minimize the number of coins dispensed as change.

2. Huffman Coding

- **Simple Explanation**: Huffman Coding is used to compress data without losing any information. It assigns shorter codes to more frequent characters and longer codes to less frequent ones, thus reducing the overall size of the data.
- **Real-World Application**: Used in compressing data for transmission over the internet, making websites load faster and reducing network congestion.

3. Task Scheduling

- **Simple Explanation**: Imagine you have a list of tasks, each with a deadline and a penalty for not completing it on time. A greedy algorithm can schedule the tasks in such a way as to minimize the total penalty, even if it means some tasks are not completed.
- **Real-World Application**: Used in operating systems to schedule tasks for execution, in project management, and in CPU scheduling in computers to optimize performance.

4. Prim's and Kruskal's Algorithms for Minimum Spanning Tree

- **Simple Explanation**: To connect a set of points (like cities) with the least total length of roads (or cables), these algorithms help find the most efficient way to do so without forming any loops.
- **Real-World Application**: Designing network layouts for electrical grids, computer networks, and road systems to minimize the total construction cost.

Summary

Greedy algorithms are invaluable in solving optimization problems efficiently and effectively. They offer simple yet powerful solutions to a wide range of real-world issues, from data compression to network design, making them a key tool in the arsenal of computer scientists and engineers. By understanding and applying greedy algorithms, we can devise solutions that are not only theoretically optimal but also practical and applicable in everyday scenarios.
