Assignment 11

Due on 4/11/2011, 11:59 PM

1. What is an algorithm? Give a real-world example of an algorithm.
2. What is an algorithm's efficiency? Explain the concept of Big O notation.
3. What is the importance of time complexity in algorithm design?

Part A: The Knapsack Problem

1. Describe the Knapsack problem in your own words.
2. Explain how dynamic programming can be used to solve the Knapsack problem.
3. Provide an example of a Knapsack problem and its solution using dynamic programming.

Part B: Sorting Algorithms

1. Discuss the concept of sorting algorithms and their importance.
2. Explain the differences between Insertion Sort and Quick Sort.
3. Implement both Insertion Sort and Quick Sort algorithms and compare their performance.

Part C: Graph Theory

1. Define a graph and explain its significance in computer science.
2. Describe Dijkstra's algorithm and its application.
3. Implement Dijkstra's algorithm to find the shortest path between two vertices in a given graph.

Part D: C++ Programming

1. Write a C++ program that implements the Knapsack problem using dynamic programming.
2. Test your program with various data sets and analyze its performance.
3. Document your findings and optimizations for improving the algorithm's efficiency.