CS 3364 ANALYSIS OF ALGORITHMS

PROJECT 2

Course Sequencing Program using DFS.

BY : LESLIE OMONZANE R11756088

AYOBAMI AJALA R11720649

EWORITSE MABUYAKU R11710212

OLUWAFERANMI OGUNNIRAN R11760605

**Table of Contents**

**1.Introduction**

* Purpose of the Program
* Program Overview

**2.Usage Instructions**

* How to Use the Program
* Supported Operations

**3.Implementation Details**

* Program Structure
* Data Structures
* Functions and Algorithms

**4.Demonstration**

* Example Usage
* Output Interpretation

**5.Conclusion**

* Summary
* Key Takeaways

**Introduction**

**Purpose of the Program**

The purpose of the Course Sequencing Program is to determine the order in which courses should be taken by considering their prerequisites. This program uses a topological ordering algorithm to find the correct sequencing, which is essential for students planning their academic curriculum.

**Program Overview**

The program consists of a set of data structures and functions that allow users to:

* Initialize a course library
* Add courses and their prerequisites.
* Display the recommended course sequencing using a topological ordering algorithm

**Usage Instructions**

**How to Use the Program**

To use the Course Sequencing Program, follow these steps:

* Initialize the course library with initializeCourseLibrary().
* Add courses using addCourse(id, name).
* Add course prerequisites with addPrerequisite(parent, child).
* Display the course sequencing:
* Use displayAll() to display a hierarchical view.
* Use topologicalSort() to display a flat list.

**Supported Operations**

The program provides the following key operations:

* Initialization: Initialize the course library and data structures.
* Add Courses: Add new courses with their IDs and names.
* Add Prerequisites: Specify course prerequisites to establish dependencies.
* Display Sequencing: Visualize the course sequencing using either a hierarchical or flat list view.

**Implementation Details**

**Program Structure**

The Course Sequencing Program is structured as follows:

* Data Structures: It uses arrays and structures to store course information and prerequisites.
* Functions: The program includes functions to manage the course library and perform topological sorting.

**Data Structures**

* Course: A structure to represent a course with an ID and name.
* CourseLibrary: A structure to store all courses and their prerequisites.

**Functions and Algorithms**

* initializeCourseLibrary(): Initializes the course library and related data structures.
* addCourse(id, name): Adds a new course to the library.
* addPrerequisite(parent, child): Specifies course prerequisites.
* DPS(): A Depth-First Search algorithm to find the topological order.
* topologicalSort(): Performs topological sorting and displays the course sequencing.
* recursiveDisplay(): Recursively displays the hierarchical view of courses.

1. **Demonstration**

**Example Usage**

Here's a simple example of how to use the program:

* Initialize the course library.
* Add courses and their prerequisites.
* Display the recommended course sequencing using either displayAll() for a hierarchical view or topologicalSort() for a flat list.

**Output Interpretation**

The output will present the course sequencing based on prerequisites, providing a clear order in which courses should be taken. For hierarchical views, the output represents the course hierarchy with dependencies.

**Conclusion**

**Summary**

The Course Sequencing Program is a practical tool for students and educators to determine the order of taking courses. It employs a topological ordering algorithm to efficiently compute the sequencing while considering prerequisites.

**Key Takeaways**

1. The program allows for the input of course data and dependencies.
2. Users can visualize the sequencing using both hierarchical and flat list views.
3. Understanding course dependencies is crucial for efficient academic planning.