**CS 300 Project One: Pseudocode and Runtime Analysis**

**Course Management System**

This program reads a file containing course information and stores it in a vector, hash table, or binary search tree (BST). The program performs the following tasks:

1. Load course data from a file
2. Find a specific course and show prerequisites
3. Print all courses in order
4. Exit the program

**Runtime Analysis**

Vector:

* Insertion: O(1) (unsorted) or O(n) (if sorted)
* Search: O(n)
* Sorting: O(n log n)
* Printing: O(n log n)

Hash Table:

* Insertion: O(1) (average)
* Search: O(1) (average)
* Sorting: O(n log n) (traversal)
* Printing: O(n)

BST:

* Insertion: O(log n) (balanced) or O(n) (unbalanced)
* Search: O(log n) (balanced) or O(n) (unbalanced)
* Sorting: O(n) (in-order traversal)
* Printing: O(n)

**Comparison of Data Structures**

Vector:

* Advantages: Simple, easy to implement.
* Disadvantages: Slow search (O(n)), sorting required.

Hash Table:

* Advantages: Fast search (O(1)), efficient storage.
* Disadvantages: Collisions possible, uses extra memory.

BST:

* Advantages: Keeps data sorted, efficient when balanced.
* Disadvantages: Slower if unbalanced.

**Recommendation**

The hash table is the best choice because it provides fast search and insertion (O(1) on average). If maintaining order is necessary, a balanced BST is an alternative since it provides O(log n) operations while keeping courses sorted.

For this project, the hash table is recommended for efficiency in managing course data.

Pseudocode

**Vector Implementation**

FUNCTION loadCourses(filePath, courseList)

OPEN file

IF file not found THEN

PRINT "Error"

END IF

FOR each line in file

SPLIT line into tokens

CREATE course object

ADD to courseList vector

END FOR

CLOSE file

END FUNCTION

**Hash Table Implementation**

FUNCTION loadCourses(filePath, courseHashTable)

OPEN file

IF file not found THEN

PRINT "Error"

END IF

FOR each line in file

SPLIT line

CREATE course object

COMPUTE hashKey = Hash(courseNumber)

INSERT course in hashTable

END FOR

CLOSE file

END FUNCTION

**BST Implementation**

FUNCTION insertCourse(node, course)

IF node is NULL THEN

RETURN new node with course

END IF

IF courseId < node.courseId THEN

node.left = insertCourse(node.left, course)

ELSE

node.right = insertCourse(node.right, course)

END IF

RETURN node

END FUNCTION

**Search Course in BST**

FUNCTION searchCourse(node, courseId)

WHILE node is NOT NULL

IF node.courseId == courseId THEN

PRINT course details

ELSE IF courseId < node.courseId THEN

node = node.left

ELSE

node = node.right

END IF

END WHILE

PRINT "Course not found"

END FUNCTION