**Vector Pseudocode**

**Main Flow:**

BEGIN

DISPLAY menu

WHILE user does not choose to exit

READ user choice

SWITCH on choice

CASE 1:

CALL loadCoursesFromFile

CASE 2:

CALL sortCoursesByAlphanumeric

CALL displayAllCourses

CASE 3:

PROMPT user for course ID

CALL findAndDisplayCourseInfo

CASE 9:

EXIT

END SWITCH

END WHILE

END

**Loading Courses into Vector:**

FUNCTION loadCoursesFromFile(filePath)

OPEN file at filePath

FOR each line in file

PARSE line into courseID, title, and prerequisites

CREATE new Course object with parsed data

ADD Course object to courses vector

END FOR

CLOSE file

END FUNCTION

**Sorting Courses in Alphanumeric Order:**

FUNCTION sortCoursesByAlphanumeric(courses)

CALL quickSort(courses, 0, length of courses - 1)

END FUNCTION

**Displaying All Courses:**

FUNCTION displayAllCourses(courses)

FOR each course in courses

PRINT courseID and title

END FOR

END FUNCTION

**Finding and Displaying Course Information:**

FUNCTION findAndDisplayCourseInfo(courses, courseID)

FOR each course in courses

IF courseID equals course.courseID

PRINT course.title and course.prerequisites

RETURN

PRINT "Course not found."

END FUNCTION

**Runtime Analysis Chart**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data Structure | Operation | Line Cost | # Times Executes | Total Cost | Runtime |
| Vector | Load Courses from File | 1 | n | n | O(n) |
|  | Sort Courses | 1 | n\*log(n) | n\*log(n) | O(n\*log(n)) |
|  | Search for Course | 1 | n | n | O(n) |

**Advantages and Disadvantages Analysis**

**Advantages:**

* Simplicity: Vectors are easy to implement and work with, particularly for smaller datasets.
* Sequential Access: Vectors allow for direct access to elements via indexing, making them suitable for scenarios where data needs to be accessed frequently in a linear fashion.
* Memory Management: Memory is contiguous, which can lead to better cache performance.

**Disadvantages:**

* Poor Performance with Large Data Sets: Inserting or deleting elements in a vector can be slow, especially as the size of the vector increases because it might require shifting elements.
* Sorting Required: To maintain an ordered list, vectors require sorting, which adds overhead, particularly as the number of courses increases.
* Inefficient Search: Searching for an element requires O(n) time, making it less efficient for large datasets.