*//Project One*

//Initialize

Data Structure

User input

//Display menu in a Loop

Display Menu

1. Load Data
2. Print course list
3. Print course details

9. Exit

User Input

If 1

Load Data

If 2

Print Course List

Alphanumeric Course List

If 3

Print course details

Course ID

Course Title

Prerequisites

If 9

Exit Program

End

*//Vector*

Sort the course data

Alphanumerically by course number

Ascending

//Print

Print Sorted courses

End

*//Hash Table*

Sort Keys

Alphanumerically by course number

Print Sorted Courses

Hash Table

For Each Key sorted

Print the course hash table

End

*//Tree*

//Print

Print the Sorted Courses

// Traversal Function

In Order Traverse

End

Evaluation

Vector

Open File: O(1)

Read Lines: O(n)

Parse Lines: O(n)

Prerequisites O(n)

Course Objects: O(n)

**Worst case running time is O(n)**

The advantages are it is simple to understand and efficient, The disadvantages are that the search is linear and it is not as fast at searching.

Hash Table

Open File: O(1)

Read File: O(n)

Parsing: O(n)

Course Objects: O(n)

Storing Course in hash Table: O(n)

**Worst case running time is O(n)**

The advantages are a fast search and fast insertion and deletion. It is good for searching by numbers and with larger data sets. Disadvantages are it’s not the best for sorting data in order and it can be more complex than vectors.

Tree

Open File: O(1)

Read File: O(n)

Course Object: O(n)

Insert into Tree: O(n)

**Worst case running time: O(n)**

Advantages are it sorts automatically and has a fast search time. Disadvantages are It is more complex than vectors and hash tables.

So, I would recommend using a hash table since the program needs to have a fast course lookup and hash tables are great with numbers and large data sets.