Vector Pseudocode:

Program Main:

Display Menu:

Menu Option 1: Load Courses

CREATE a course data structure 1

Read rows in csv file using a FOR LOOP n

Add each course to course data structure 1

Menu Option 2: Print Course List

Sort courses using alphanumeric order from lowest to highest 1

Display courses using a FOR LOOP n

Menu Option 3: Print Course Information

Display Course Title and any prerequisites for any individual course 1

Menu Option 4: Exit Program

Exit Program

|  |  |  |  |
| --- | --- | --- | --- |
| Code | Line Cost | #Times Executed | Total Cost |
| CREATE a course data structure | 1 | 1 | 1 |
| Read rows in csv file using a FOR LOOP | n | n | n |
| Add each course to course data structure | 1 | n | n |
| Sort courses using alphanumeric order from lowest to highest | 1 | n | n |
| Display courses using a FOR LOOP | 1 | n | n |
| Display Course Title and any prerequisites for any individual course | 1 | 1 | 1 |
| Total Cost | | | 4n+2 |
| Runtime | | | O(n) |

Hash Table Pseudocode:

Program Main:

Display Menu:

Menu Option 1: Load Courses

Size the Hash Table 1

Read rows in csv file using a FOR LOOP n

Add each course to the hash table 1

Menu Option 2: Print Course List

Sort courses using alphanumeric order from lowest to highest 1

Display courses using a FOR LOOP n

Check each bucket 1

Check next bucket 1

Menu Option 3: Print Course Information

Display Course Title and any prerequisites for any individual course 1

Menu Option 4: Exit Program

Exit Program

|  |  |  |  |
| --- | --- | --- | --- |
| Code | Line Cost | #Times Executed | Total Cost |
| Size the Hash Table | 1 | 1 | 1 |
| Read rows in csv file using a FOR LOOP | n | n | n |
| Add each course to the hash table | 1 | n | n |
| Sort courses using alphanumeric order from lowest to highest | 1 | n | n |
| Display courses using a FOR LOOP | n | n | n |
| Check each bucket | 1 | n | n |
| Check next bucket | 1 | n | n |
| Display Course Title and any prerequisites for any individual course | 1 | 1 | 1 |
| Total Cost | | | 6n+2 |
| Runtime | | | O(n) |

Binary Search List Pseudocode:

Program Main:

Display Menu:

Menu Option 1: Load Courses

CREATE a Binary Search List 1

Read rows in csv file using a FOR LOOP n

Add each course to the Binary Search List 1

Menu Option 2: Print Course List

Sort courses using alphanumeric order from lowest to highest (BSL) logn

Display courses using a WHILE LOOP logn

Menu Option 3: Print Course Information

Display Course Title and any prerequisites for any individual course 1

Menu Option 4: Exit Program

Exit Program

|  |  |  |  |
| --- | --- | --- | --- |
| Code | Line Cost | #Times Executed | Total Cost |
| CREATE a Binary Search List | 1 | 1 | 1 |
| Read rows in csv file using a FOR LOOP | n | n | n |
| Add each course to the Binary Search List | 1 | n | n |
| Sort courses using alphanumeric order from lowest to highest | logn | logn | logn |
| Display courses using a WHILE LOOP | logn | logn | logn |
| Display Course Title and any prerequisites for any individual course | 1 | 1 | 1 |
| Total Cost | | | 2n+2logn+2 |
| Runtime | | | O(n) |

Vector Evaluation:

Advantages:

* Easy to resize.
* Efficient display speed that becomes less efficient with size.
* Can be almost fully recursive.

Disadvantages:

* Uses a higher than normal amount of memory to accommodate for possible new elements.
* Singly Linked Lists take time to traverse.

Hash Table Evaluation:

Advantages:

* Memory usage is medium.
* Implementation is simple.

Disadvantages:

* Resizing can be finicky.
* Could potentially use a LARGE amount of space.
* Not very recursive.

Binary Search Tree Evaluation:

Advantages:

* Easy to resize.
* Display speed is faster than Vectors.
* Can be fully recursive.

Disadvantages:

* Complicated implementation.
* Memory usage is determined as size increases.

**Which Should Be Used?** I suggest a Vector or Binary Search Tree. This will not need to be resized, this means that all three are decent for this. Display speed of Vectors are decent for this, as we will not have a ridiculous number of items. Vectors are the happy medium of the three. Binary Search Trees will be faster, but they are much more complicated to implement.