Cory Hurlbut

CS300 Project 1

**Pseudocode**

1. **Resubmit pseudocode from previous pseudocode assignments and update as necessary**.

Resubmitted with this assignment in separate files.

1. **Create pseudocode for a menu**.

Choice = 0

While choice is not 4

Print list of options

Cin choice

Switch choice

Case 1

Load data using the pseudocode to open a file and load data

Case 2

Call function to print (pass root or nothing to make it print all)

Case 3

Call function to print(pass specific node to print it and its prereqs)

Case 4

Exit program

default

throw error(invalid choice)

1. **Design pseudocode that will print out the list of the courses in the Computer Science program in alphanumeric order.**

**BST:**

if node is not null

recurse(node.left)

print node

recurse(node.right)

**HashTable(vector of linked lists):**

Keys[]

Loop through hashtable slots

Loop through nodes in slot

Add each key to keys[]

Keys.sort()

Loop through keys

Print key and hashtable.get(key)

**Vector:**

Sort vector

Loop through vector

Print element

**Evaluation**

1. **Evaluate the run-time and memory of data structures that could be used to address the requirements**.

**Analysis:** The reading of the data is all O(N) because the code is about the same. It differs in manipulating the structures but not in creating them.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Vector | HashTable | BST |
| Loading data | O(N) | O(N) | O(N) |

**Recommendation:**

In the worst case, all of them are O(N) for reading the file, preparing it for loading, and then loading it into the data structures. Since the program will be mostly used for searching and printing, a hash table might be the best solution since its average complexity for lookup is O(1). Depending on the intended use, it could also be useful to use a Vector, as it also does lookups and inserts at the end(loading) at an average of O(1).