Carl LaLonde

8/24/2024

Run-time and Memory Analysis

Vector – worst case for reading the file and creating course objects is O(n)

Hash Table – worst case for reading the file and creating course objects is O(n2)

Binary Search Tree – worst case for reading the file an creating course objects is O(n2)

Vectors are easily more simple to implement. However, if the vector grows beyond its size, it must be resized which can be an inefficient way to use memory. Vectors are also not very efficient for adding or removing items in the middle of the vector.

Hash Tables offer quick searches and adding of items. They are also easily resizable. However, Hash Tables require more memory for storing keys and handling collisions. They may also require further steps for sorting items.

Binary Search Trees, if balanced, can be great with memory usage. Data is easily sorted, inserted and removed. However, if the tree becomes unbalanced, it can lead to poor performance such as slower searches, insertions and deletions of items. BST’s are more difficult to implement.

I believe a BST is the best option for the ABCU project. If balanced, a BST will allow for courses to be searched for, added and or removed with ease. BST’s may also be more memory efficient with larger data sets which I’m not entirely sure would apply for this particular application.