BINARY TREE

For the binary tree load the big o notation will be O(N)

There is 1 line of code that executed before the for loop and 12 lines of code that execute during the loop. This gives us 1+O(12N) in the worst-case scenario. However using the two rules of big O notation we get rid of constants and keep the highest terms thus O(N).

VECTOR

For the vector load the big o notation will be O(N)

There is 1 line of code that executed before the for loop and 8 lines of code that execute during the loop. This gives us 1+O(8N) in the worst-case scenario. However, using the two rules of big O notation we get rid of constants and keep the highest terms thus O(N).

HASH TABLE

For the hash table load the big o notation will be O(N)

There is 1 line of code that executed before the for loop and 8 lines of code that execute during the loop. This gives us 1+O(8N) in the worst-case scenario. However, using the two rules of big O notation we get rid of constants and keep the highest terms thus O(N).

Each type of data storage unit of vector, hash table and binary tree each have different advantages. The vector is a linear list that must be accessed sequentially which can be time consuming especially in the worst-case scenario of a sear when the search variable is not in the vector and the entire structure must be searched. As well sorting a vector takes many steps and will depend on the method used to sort but because it is linear it is automatically difficult to sort. Hash tables use hash code linked to each added data to store the data. This works best when the key id of the data is a unique identifier. Hashing gets time consuming when there are multiple of the same data so that there must be a lot of rerouting of the data. If properly implemented a hash table is fast for insertion, deletions, and searches. Binary trees inherently start sorting data as it is being inserted and thus allow for fast sorting times. However deleting can be complicated and search, insertion and deletion times will depend on the height of the tree.

For the computer science department program, I will use a hash table. The reason being that the course ids are unique which should mitigate duplicate entries as well the search, insertion and deletions will be fast. The binary tree may offer faster sorting but I don’t think this will be enough to make it advantageous to use instead of hash table.