**Run Time Analysis**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Binary Search Tree | Vector | Hash Table |
| Load Data | O(log n) | O(1) | O(1) |
| Search | O(log n) or O(n) | O(n) | O(1) |
| Sort/Print | O(n) | O(log n) | O(n) |

**Advantages and recommendation**

There is no structure that is going to be a one size fits all solution to all problems. If that was the case, then there really wouldn’t be a need for the various types of data structures we have available to use. Some structures, such as a hash table, will have a run time of either O(1) or O(n) depending on collisions. Generally a hash table will run around O(1) though. A binary search tree tends to stick to O(log n) pretty consistently but being unbalanced could put that tree at O(n). And vectors can also be a great choice because of how fast it can append items to a list.

If I had to give my recommendation on which data structure we should use then I would have to say the binary search tree. This is because we are going to be working with university courses and will need to be able to search and display courses that are going to be in an alphanumeric format. A binary search tree is going to be easier to traverse and display the necessary information due to the fact that it doesn’t need to be sorted. So I feel that this would be the most effective data structure for our assignment.