Reed Murray

CS-300 Data Structures and Algorithms

Runtime analysis for populating, sorting, and printing the nodes of a data structure.

|  |  |  |
| --- | --- | --- |
| Vector | Hash Table | Binary Search Tree |
| The Vector is a simple data structure which uses indexing to hold values. When inserting or deleting items from a Vector, the runtime is O(n) because indexes must be shifted. However accessing a Vector is O(1) because of the indexing structure. | The Hash table maps the values to a memory address using hashing. This technique means that values at memory addresses do not ever need to be shifted. Searching, and inserting into the Hash table have a runtime of O(1), because the hashing equation evaluates the memory address directly. However the Hash Table cannot sort quickly. | The Binary Search Tree is consistent, running at O(log (n)) for access, search, insertion, and deletion. For the binary search tree to run efficiently, it must be balanced. If the tree has only 8 nodes, but has a height of 7, it is a straight line and won’t run to it’s potential. This is a disadvantage because the height must be monitored and maintained within the program. |