Karen Schick

CS 300 DSA: Analysis and Design

Prof Norman Lippincott

April 13, 2025

**Project One - Analysis**

Hash tables, binary search trees (BSTs), and vectors are data structures that all have their own strengths and weaknesses. Hash tables are extraordinarily speedy with a constant time complexity (O(1)) when searching, inserting, or deleting elements in large datasets. However, dealing with collisions is complicated. If collisions are not handled properly, their performance can downgrade to O(n). They also consume a lot of memory due to the use of buckets when mapping keys to values. When balanced, binary search trees maintain a O(log n) time complexity. While slower than hash tables, BSTs are more efficient at maintaining sorted data. However, they are more complex and difficult to implement than either hash tables or vectors.

For small datasets, such as the one used in the Computer Science department at ABCU, the simplicity of a linear search on a vector can be faster and more efficient than using hash tables or binary search trees due to their overhead. Vectors do not use memory structures like buckets used in hash tables or nodes in trees, making them easy to implement. A linear search can be fast on a small dataset due to its linear time complexity (O(n)). Vectors are a sensible and effective choice for small datasets.