Matthew Trembley

Southern New Hampshire University

10/3/2021

CS-260-R1998

5-1 Programming Journal

The impact a hashing algorithm has on performance is quite significant. A hashing algorithm can be very fast, as long as there are minimal collisions. A collision occurs when an item being inserted into the hash table lands in the same bucket as an existing item in a hash table. This can help considerably when dealing with values that are not in any particular order. If an array or vector has not been sorted, searching for a value means that you will need to iterate through at least half of it to retrieve a desired value – depending on the algorithm that is set up. With hashes, the location is essentially generated by the value itself. ZyBooks says that the biggest advantage of using hash tables is that searching for an object can require only one search, instead of O(N) for other types of searches through list. If this list is perhaps hundreds of thousands long, this could take a very, very long time.  
 Beyond indexing data, other uses a hashing algorithm may have are finding certain values, inserting them, or removing them, much faster. As said above, searching through a list will take O(N) time, but using hash tables will take O(1) time, which is significantly faster.