**Assignment 3 - Hashing Answers:**

* + Table Results:

A yellow and black rectangular table with numbers

Description automatically generated A table with numbers and symbols

Description automatically generated

1. The time complexity for insertion, delete, and search value operations in a hash table **assuming standard independent uniform hashing and a doubly linked list** are as follows:

|  |  |  |
| --- | --- | --- |
| **Worst-Case Time Complexity** | | |
|  | **Unsorted Linked Lists** | **Sorted Link Lists** |
| **Successful Search** |  |  |
| **Unsuccessful Search** |  |  |
| **Insert** |  |  |
| **Delete** |  |  |

By maintaining a sorted linked list in each of the hash table slots, the worst-case time complexity is slightly worse than in an unordered LL. Linked lists are not random-access data structures like arrays so we can’t use a more efficient algorithm such as binary search to speed up the search time. Additionally, by maintaining a sorted order it requires the algorithm to potentially update the entire LL upon inserting a new element since it is no longer sufficient to just insert the new element at the beginning of the list and leave it as-is.

It is possible that sorting the linked list *slightly* speeds up the search function in the average-case, but those differences would be marginal, and the worst-case complexity remains . Since sorting the LL makes the time complexity worse for insertions and doesn’t provide any significant benefits, this change would not be recommended