Aidan Duffy

Boston University

METCS 526

Module 4 Homework (Problems 1 through 5)

**Problem 1 (10 points)**

**Text, letter

Description automatically generated**

**Problem 2 (10 points).**

**Text, letter

Description automatically generated**

**Problems 3 (10 points each).**

**Diagram

Description automatically generated**

**Problem 4 (10 points).**

**A picture containing diagram

Description automatically generated**

**Problem 5 (10 points).**

**Text, letter

Description automatically generated**

**Problem 6 (10 points).**

I ran the program several times, here are a couple of the outputs:

Number of keys = 100000

HashMap average total insert time = 7.4, ArrayList = 1.3, LinkedList = 1.5

HashMap average total search time = 2.4, ArrayList = 5940.9, LinkedList = 17547.2

Number of keys = 100000

HashMap average total insert time = 7.2, ArrayList = 1.2, LinkedList = 1.7

HashMap average total search time = 2.5, ArrayList = 5935.7, LinkedList = 17344.5

One thing that I observed almost instantly – mostly due to the fact that I had placed print statements throughout my program to signal progress – was that for the outermost loop (the one that runs 10 times), everything up until the array list search executed instantaneously to the human eye. After that, the linked list search loop also hung for quite a big of time. These were confirmed by the outputs of the program. Both insertion and search for the HashMap ran in constant time, despite the worst possible time being O(n). The insertion loops for both the ArrayList and LinkedList ran in constant time, as well. However, the search loops for those two ran in N time, though the ArrayList was significantly faster (~3x faster) than the LinkedList.

This experiment solidified the notion from lecture/readings that big-Oh analysis truly is the “worst case” and certainly does not happen always. Also, it reaffirmed the efficiency of hash data structures like HashMaps and Hash Tables.