Declaration

1) I have accurately identified and included the sources of all facts, ideas, opinions, and viewpoints from others in the assignment references. All direct quotations, paraphrasing, and discussions of ideas from books, journal articles, internet sources, course materials, or any other sources used are properly acknowledged and cited in the assignment references.
2) I have not used unauthorised artificial intelligence tools or aids.
3) I understand and am compliant with ATU's policy and procedures regarding Academic Integrity and I am aware of the consequences of any violations.
4) I have followed the referencing guidelines recommended in the assignment instructions and / or programme documentation.
5) I understand that not adhering to the word limits on each section will result in penalties.
6) By signing this form or by submitting material online I confirm that this assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other programme of study.
7) By signing this form or by submitting material for assessment online I confirm that I have read and understood AQAE022 Academic Integrity Policy.

I confirm I agree to this declaration. ☒

Print Signature: Kevin Egan

Date: 26/12/2025

1.Hashing Fundamentals
Explain the following:

1.1 Hash Function

A function which by using an input key, creates a mapping to an index in a hash table (GeeksForGeeks, 2025).

1.2 Hash Table

A hashtable is a data structure which has an index value which is generated by a hash function. It allows for the storage of key-value pairs. (Domino Data Lab, n.d.) It also allows for a search/insertion/deletion time of O(1).

1.3 Hash Collision

This is an event that occurs when the hash function assigns two different inputs the same hash value as output. This is a phenomenon which can occur due to mathematical limitations since a hash function can accept an infinite number of inputs, yet produces a finite number of possible outputs (JumpCloud, 2025).

1.4 Load Factor

This is a measure of how full the storage of a hash table is. It is calculated from the ratio of how many elements are stored in the hash table vs the total number of slots/buckets available. It is used to determine the probability of collisions and to determine the performance of the hash table in search, insertion, deletion (Fiveable, n.d.)

1.5 Open Addressing

Open Addressing is a solution to the problem of hash collision. All items get stored within the hash table. When a collision happens, it looks for the next available spot in the hashtable (Black, 2015).

2. Compare Hashing vs Tree based structures

2.1 How hash based data structures differ from tree based data structures.

Hash based data structures store data as key:value mappings in an unsorted way. Hash tables also use more memory than trees as space is required for upcoming inputs.

Tree based data structures are neatly sorted, with a node typically having two children nodes. The nodes are ordered hierarchically; left child's value is smaller than the parent, right child's value is greater than the parent. In the case of AVL trees and Red Black Trees, these are self balancing, ensuring that performance remains as optimal as possible (Educative Inc, 2025).

Another difference is that hash tables do not permit duplicates, whereas tree based data structures do.

2.2. Advantages and Disadvantages of Hashing vs Tree Based for insert,delete,search

For a hash map, the average time complexity for search, insert, and delete is O(1). This makes it a good solution for operations which require constant operation time for searching, inserting, and deleting items.

For a tree structure, the average time complexity for search, insert, delete is O(log n). This means that search, insert, and delete operations take longer (Educative Inc, 2025). Though minheaps/maxheaps are tree structures which can have a time complexity of O(1) to find the lowest/highest (depending on whether minheap or maxheap is used) value.

Q5. Describe how Question 4 (LRU cache) can be updated to support concurrency

There are potential race conditions in the current LRU cache as it is possible that different threads can access the hashmap/linked list, modify it, causing unpredictable behaviour for the other threads. Thread safety measures should be implemented to ensure that multiple threads can modify the hashmap without causing issues for other threads. This could involve coarse grained locking by locking the hashmap and linkedlist to be used by only one thread at at time. Or It could involve fine grained locking by using readLock and writeLock in the get() and put() methods, ensuring that only one thread at a time can read/write to the hashmap/LinkedList. This allows for greater performance/concurrency compared to the coarse grained solution.

There is also the possibility of using ConcurrentHashMap to make the hashmap threadsafe (GeeksforGeeks, 2025), though there is no “Concurrent” counterpart for the LinkedList.

References

Black, Paul, "open addressing", in *Dictionary of Algorithms and Data Structures* [online], Paul E. Black, ed. 17 September 2015. (Accessed 20 December 2025) Available from: https://www.nist.gov/dads/HTML/openAddressing.html

Domino Data Lab (n.d.) What are hash tables? Domino Data Lab. Available at: https://domino.ai/data-science-dictionary/hash-table (Accessed: 20 December 2025).

Educative, Inc. (2025) Binary search tree vs. hash tables. Available at: (Accessed: 24 December 2025).

Fiveable (n.d.) Load factor - (Data Structures) - Vocab, Definition, Explanations. Fiveable. Available at: https://fiveable.me/key-terms/data-structures/load-factor (Accessed: 20 December 2025).

GeeksforGeeks (2025) ConcurrentHashMap in Java. Available at: https://www.geeksforgeeks.org/java/concurrenthashmap-in-java/ (Accessed: 26 December 2025).

GeeksforGeeks (2025) Hash Functions and Types of Hash functions. GeeksforGeeks. Available at: https://www.geeksforgeeks.org/dsa/hash-functions-and-list-types-of-hash-functions/ (Accessed: 20 December 2025).

JumpCloud (2025) What Is a Hash Collision? JumpCloud. Available at: https://jumpcloud.com/it-index/what-is-a-hash-collision (Accessed: 20 December 2025).