CS 201: Data Structures II
Merkle Tree

L2 Group 2

Spring 2023

1 Group Members

1. Ayila Emad
2. Daniyal
3. Dua Batool
4. Rohan Raj

2 Data Structure

We will be using Merkle tree which is a tree-like hash-based data structure where each leaf node represents a hash value of a piece of data. Every other node represents the hash value of the concatenation of its child nodes in a recursive hashing process, with the root node representing as single hash value that is considered as the summary of the dataset. The Merkle Tree is commonly used to verify the integrity and authenticity of large datasets.

3 Application

1. **Blockchain technology:** Merkle trees are widely used in blockchain technology to ensure the authenticity and integrity of transactions and blocks in the blockchain. Each block in the blockchain contains a Merkle tree of the transactions in that block, and the root hash of this tree is included in the block header.
2. **Version control systems:** Merkle trees are used in version control systems, such as Git, to efficiently track changes to large code repositories. In Git, each commit contains a Merkle tree of the changes made to the codebase, allowing for efficient verification of the integrity of the codebase.
3. **File systems:** Merkle trees can be used in file systems to efficiently verify the integrity of large files. Instead of having to hash the entire file, a Merkle tree can be constructed, allowing for efficient verification of the integrity of the file.

4 Functionality

1. **Insertion** This function allows the user to insert a new data element into the Merkle Tree. Once inserted, the Merkle tree is updated by recalculating the hash values for the nodes along the path from the inserted leaf node to the root node of the tree.

2. **Deletion** This function allows the user to delete a data element from the Merkle Tree. Once deleted, the Merkle tree is updated by recalculating the hash values for the nodes along the path from the inserted leaf node to the root node of the tree.

3. **Verification** This function allows the user to verify the integrity of a data element or a set of data elements in the Merkle Tree. The user can verify whether a particular data element is present in the Merkle Tree and whether the data element has been tampered with using the hash values.

4. **Hashing** This function allows the user to compute the hash of a data element or a set of data elements in the Merkle Tree. The hash value is used to verify the integrity of the data element or to compare it with other data elements in the Merkle Tree.

5. **Merkle Proof** This function allows the user to generate a Merkle Proof, which is a set of hash values that can be used to prove the inclusion or non-inclusion of a data element in the Merkle Tree. The Merkle Proof can be used to provide evidence that a particular data element is present or absent in the Merkle Tree without revealing the actual data element.

Hash_Tree.svg.png

5 Datasets

Provide a brief description of the datasets that will be used (if any).

6 Work Distribution

Fill in the table which indicates the work distribution of each member.

Item	Activity	ID
1	Activity 1	ae07352
2	Activity 2	member2
3	Activity 3	member3
4	Activity 4	member4

7 Attribution

OpenAI. (2021). ChatGPT [Computer software] Simplilearn. (n.d.). Merkle Tree in Blockchain. Simplilearn. Pandey, S. (2018, August 9). Merkle Tree: A Simple Explanation and Implementation. Davis, B. (2018, July 11). Applications of the Merkle Tree Data Structure. Medium.

References

[1] OpenAI. (2021). ChatGPT [Computer software]