Part 1 of Project 2: Part 2 proposal

1 Merkle Tree

We propose the integration of Merkle trees into our project 2 to enhance data integrity, facilitate efficient transaction verification, and improve overall security. Merkle trees provide a compact and secure way to represent the transaction history within each block.

1.1 Method

Merkle Tree for Transaction Integrity: Utilize a Merkle tree structure to organize and verify the integrity of transactions within each block. This ensures that any alteration or corruption of a single transaction is quickly detectable through the Merkle tree.

1.2 Metric

Transaction Verification Speed: Measure the speed at which nodes can verify transactions using Merkle proofs compared to traditional methods.

1.3 Implementation Plan

Code Integration: Implement the necessary changes to integrate Merkle trees into the existing blockchain codebase.
Testing and Validation: Conduct thorough testing to ensure the correct functioning of Merkle tree integration and validate its impact on transaction verification speed,

2 Dynamic list for disseminating information(Broadcast)

We propose the implementation of a dynamic peer list to enhance the flexibility and scalability of our blockchain network. This improvement aims to facilitate dynamic peer discovery, adapt to changing network conditions, and optimize the dissemination of transactions and block data.

2.1 Method

Use the protocol in P2P network, when a new node wants to join in the network, it asks its neighbor to get the address of their neighbors, and add them to its address list, and so on.

3 Remove one assumption

In Part I, we assume that miner does not include transactinos that can not be done in its own UTXO set. Actually, malicious miner can do this and we should detect it and drop the block(chain).

3.1 Implementation Plan

It is easy in one block case. In chain case (which happens when there's fork), the miner should undo its own different blocks and redo the longer chain from another miner. In the redo process, verify every transaction in these blocks to check whether they can be done in its own UTXO set.
To test this, we design a malicious miner and see if the chain or block will be dropped by other miners.

4 Optimize the process of handling fork

For now, we send the whole chain to miner when miner receive a block that has greater ID and cannot be added to its chain. The cost is very high when there are many miners and blocks.

4.1 Method

Only send the different part. This needs interaction between the two miners.