COMP5521
Group Project

20001138G Cheuk Man Ting
21000018G Li Yat Long
21006322G To Ching Wai
21014974G Yu Hin Chung Nikko
21045374G Zhang ZhanLei

1. Blockchain Prototype

- **Index**: Height of the current block.
- **Timestamp**: Current time divided by 1000 (UNIX timestamp in seconds)
- **Previous Block Hash**: 256-bit hash of previous block header
- **Current Block Hash**: Hash of current block
- **Difficulty**: Number of bits at the beginning of block hash
- **Nonce**: Random number used to calculate the block hash which fulfills the difficulty
- **Merkle root of transactions**: 256-bit hash based on all the transactions in the block
- **Data**: the list of transactions

```
{"index":1,"previousHash":"91a73664bc84c0baa1fc75ea6e4aa6d1d20c5df664c724e3159aefc2e1186627","timesta
mp":1645799653,"data":[{"txIns":[{"signature":"","txOutId":"","txOutIndex":1}],"txOuts":
[{"address":"048f6eb168ff45d55e71c1a0ef3d6bfb94b9ec03f6e93d38a3bbb93160ec35f76013feeda65df8ffb6f9391b
e6d3e527aba48b4403f9ab8b7beb8664045a0ac8dc","amount":50}],"id":"089382501a6aff5b8cf01e5c528108cf18280
8714f6cfa2c08981d352c1db400"}],"hash":"6021856d4b77d79cf0fdf6c5f43c9297685984b9a429923c819a82de8a574e
c7","difficulty":0,"nonce":0,"merkleRoot":"9fc172448f81fd10c1f49943be373bc4adb6929f8b8b8a675d185de841
94206d","merkleTree":{"roots":[{"index":15,"parent":31,"hash":{"type":"Buffer","data":
[222,204,59,24,101,161,215,253,23,170,203,193,167,31,51,64,28,123,162,153,205,104,125,35,203,213,165,
57,61,169,242,246]},"size":1024,"data":null},{"index":39,"parent":47,"hash":{"type":"Buffer","data":
[213,109,112,45,28,20,148,46,29,118,152,172,5,12,162,184,85,182,43,65,65,90,66,96,90,15,93,221,32,243
,115,32]},"size":512,"data":null},{"index":49,"parent":51,"hash":{"type":"Buffer","data":
[200,188,230,193,202,91,214,74,67,128,123,17,245,182,83,190,81,172,207,49,14,44,91,119,10,24,3,47,237
,97,27,175]},"size":128,"data":null},{"index":52,"parent":53,"hash":{"type":"Buffer","data":
[210,182,140,125,161,38,203,98,133,50,49,158,235,172,131,173,176,236,115,145,244,20,117,139,78,247,24
7,58,254,80,146,225]},"size":64,"data":{"type":"Buffer","data":
[56,97,55,102,49,101,97,52,98,98,100,97,50,54,99,97,54,51,53,98,51,100,55,49,50,49,55,52,48,102,55,48
,54,54,56,56,102,57,52,49,52,53,101,52,100,54,55,101,50,102,51,98,101,98,56,97,101,51,48,50,57,97,102
,52]}}],"blocks":27}},
```


Index

Current Block Hash

Merkle root

Previous Block Hash

Difficulty

Data/ transaction

Timestamp

Nonce

2. Mining

- **Proof-of-Work algorithm**: By adjusting nonce
  - Only the valid block will be added to the blockchain
- **Dynamic difficulty**
  - Difficulty evaluate every 10 blocks (interval)
  - Target time generate each block = 10 seconds
    - If time taken > 2 * interval * target_time, difficulty - 1
    - If time taken < 2 * interval * target_time, difficulty + 1

2.1. Proof-of-Work algorithm

```
const findBlock = (index: number, previousHash: string, timestamp: number, data: Transaction[], difficulty: number, merkleRoot: string): Block => {
    let nonce = 0;
    while (true) {
        const hash: string = calculateHash(index, previousHash, timestamp, data, difficulty, nonce);
        if (hashMatchesDifficulty(hash, difficulty)) {
            return new Block(index, hash, previousHash, timestamp, data, difficulty, nonce, merkleRoot, gen);
        }
        nonce++;
    }
};
```

```
const hashMatchesDifficulty = (hash: string, difficulty: number): boolean => {
    const hashInBinary: string = hexToBinary(hash);
    const requiredPrefix: string = '0'.repeat(difficulty);
    return hashInBinary.startsWith(requiredPrefix);
};
```

- When block ≠ prefix of the difficulty ⇒ nonce + 1
- Hash is unique due to increasing the value of nonce
- Process will be ended if the generated hash matches the difficulty

2.2. Dynamic Difficulty

```
const getAdjustedDifficulty = (latestBlock: Block, aBlockchain: Block[]) => {
    const prevAdjustmentBlock: Block = aBlockchain[blockchain.length - DIFFICULTY_ADJUSTMENT_INTERVAL];
    const timeExpected: number = BLOCK_GENERATION_INTERVAL * DIFFICULTY_ADJUSTMENT_INTERVAL;
    const timeTaken: number = latestBlock.timestamp - prevAdjustmentBlock.timestamp;
    if (timeTaken < timeExpected / 2) {
        return prevAdjustmentBlock.difficulty + 1;
    } else if (timeTaken > timeExpected * 2) {
        return prevAdjustmentBlock.difficulty - 1;
    } else {
        return prevAdjustmentBlock.difficulty;
    }
};
```

- Objective: Ensure generated block time interval is controllable
- Difficulty is evaluated **every 10 blocks**
- Target time interval of generating each block = 10 seconds
- If time > twice of BLOCK_GENERTION_INTERVAL * DIFFICULITY_ADJUST_INTERVAL

  ⇒ **Difficulty - 1**
- If time < twice of BLOCK_GENERTION_INTERVAL * DIFFICULITY_ADJUST_INTERVAL

  ⇒ **Difficulty + 1**

3. Transaction

- **Digital signature** on TxIN by user's private key in ECDSA
- **Coinbase transaction**: Amount of bitcoins + A script which specifies the conditions under which the bitcoins associated with this output can be spent
  - In our project, 50 coins awarded to miners for mining each block

```
const COINBASE_AMOUNT: number = 50;
```

3.1. Digital signature


```
{"index":47,"previousHash":"0e00de023f5e3933351b527d23d8da1069510b63553c3e2377f0250207dbea32","timestamp":1647960058,"data":
[{"txIns":[{"signature":"","txOutId":"","txOutIndex":47}],"txOuts":
[{"address":"048f6eb168ff45d55e71c1a0ef3d6bfb94b9ec03f6e93d38a3bbb93160ec35f76013feeda65df8ffb6f9391be6d3e527aba48b4403f9ab8b7beb86
64045a0ac8dc","amount":50}],"id":"3d0fd9ab14898b94d336ead3af4cf4bf42d9979b55dc3a0c883b7582b9555692"},{"txIns":
[{"txOutId":"5d14b6d9038b5a1073911d6f274c08c1a9ed17bc4bbedc7f19b5f211fcd36d5e","txOutIndex":0,"signature":"3045022100ac9c12395f74bf
aa5974ce1da5334e7ed229b2c02ef09c4a6f06810390188b8e02200311cdb9e263244c3e2e2461a09d2034bbc98442d1af14f7a7a25f7c10d11d27"}],"txOuts":
[{"address":"04bfcab8722991ae774db48f934ca79cfb7dd991229153b9f732ba5334aafcd8e7266e47076996b55a14bf9913ee3145ce0cfc1372ada8ada74bd2
87450313534b","amount":35},
{"address":"048f6eb168ff45d55e71c1a0ef3d6bfb94b9ec03f6e93d38a3bbb93160ec35f76013feeda65df8ffb6f9391be6d3e527aba48b4403f9ab8b7beb866
4045a0ac8dc","amount":15}],"id":"1b19fc8c63641612bf3a5fcf2a83fddd5da54380999385b0a783fd32e79310ca"}],"hash":"0843ebe4c064708a1771b7
45b82767754e148616089b9e853851c098ac27a84a","difficulty":4,"nonce":28,"merkleRoot":"27ce50535e8fbacd6aebba4cb1006ad06a05d1eda5aefec
07daa7b1324c339f4","merkleTree":{"roots":[{"index":31,"parent":63,"hash":{"type":"Buffer","data":
[39,206,80,83,94,143,186,205,106,235,186,76,177,0,106,208,106,5,209,237,165,174,254,192,125,170,123,19,36,195,57,244]},"size":2048,
"data":null},{"index":71,"parent":79,"hash":{"type":"Buffer","data":
[79,175,218,120,138,102,68,16,209,37,4,27,230,137,40,211,211,245,64,38,74,157,202,78,178,42,186,94,235,94,153,17]},"size":512,"data
":null},{"index":83,"parent":87,"hash":{"type":"Buffer","data":
[31,35,194,97,136,44,132,245,237,131,23,150,181,99,163,139,153,231,147,198,17,36,157,12,152,180,216,15,40,238,72,135]},"size":256,"
data":null},{"index":89,"parent":91,"hash":{"type":"Buffer","data":
[205,14,54,116,165,215,33,162,49,109,92,4,179,250,119,239,137,246,244,231,154,223,18,235,166,118,120,128,141,204,24,168]},"size":12
8,"data":null},{"index":92,"parent":93,"hash":{"type":"Buffer","data":
[181,119,27,65,166,254,89,184,150,107,206,151,32,199,197,11,236,123,144,104,202,6,183,43,156,183,49,245,150,224,177,164]},"size":64
,"data":{"type":"Buffer","data":
[49,98,49,57,102,99,56,99,54,51,54,52,49,54,49,50,98,102,51,97,53,102,99,102,50,97,56,51,102,100,100,100,53,100,97,53,52,51,56,48,5
7,57,57,51,56,53,98,48,97,55,56,51,102,100,51,50,101,55,57,51,49,48,99,97]}}],"blocks":47}},
```


- Each user owns a set of unique private + public keys
- Signature ⇒ Sign the transaction input with the sender's private key in ECDSA
- Public key ⇒ Verify credibility of coins in the transaction

  ⇒ Used as address of user during the transaction

3.2. Coinbase transaction

```
vincentli@ViNCENT-AxE naivecoin-simple % curl -X POST http://localhost:3001/mineBlock
{"index":68,"previousHash":"32737695b093eba95cb5a59a0c4d16761b20c8087ac342392a99dfbdec788891","timestamp":1648369813,"data":[{"txIns":[{"signature":"","txOutId":"","txOutIndex":68}],"txOuts":[{"address":"048f6eb168ff45d55e71c1a0ef3d6bfb94b9ec03f6e93d38a3bbb93160ec35f76013feeda65df8ffb6f9391be6d3e527aba48b4403f9ab8b7beb8664045a0ac8dc","amount":50}],"id":"4694c2b2bc81114e86ccc55d0ce207eaea9eccb2db85f88703e2cf13edd0d3a4"}],"hash":"3e91acb4974aa29e545bb8422e9795385e128c8d671985bf2f3f7c6f57d9c8fb","difficulty":2,"nonce":3,"merkleRoot":"2e23a415cc0d0e0c83edc6e81e442a103dac7baabb78aa38346f39428393cb26","merkleTree":{"roots":[{"index":0,"parent":1,"hash":{"type":"Buffer","data":[46,35,164,21,204,13,14,12,131,237,198,232,30,68,42,16,61,172,123,170,187,120,170,56,52,111,57,66,131,147,203,38]},"size":64,"data":{"type":"Buffer","data":[52,54,57,52,99,50,98,50,98,99,56,49,49,49,52,101,56,54,99,99,99,53,53,100,48,99,101,50,48,55,101,97,101,97,57,101,99,99,98,50,100,98,56,53,102,56,56,55,48,51,101,50,99,102,49,51,101,100,100,48,100,51,97,52]}}],"blocks":1}}%
```

- Purpose: To encourage miners mining the unconfirmed transaction in the transaction pool to activate the transaction
- A coinbase transaction: only includes output is introduced to reward miners 50 coins per each block mined
- As miners usually obtain multiple coinbase transactions, in order to prevent the transaction ID being always the same due to the same amount of the coins and the same output address, block height (txOutIndex) is introduced to the input of coinbase transactions

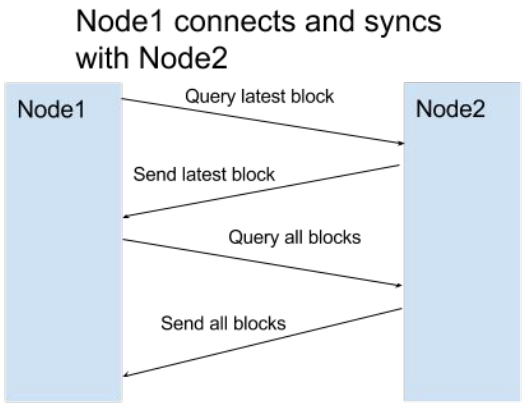
4. Network

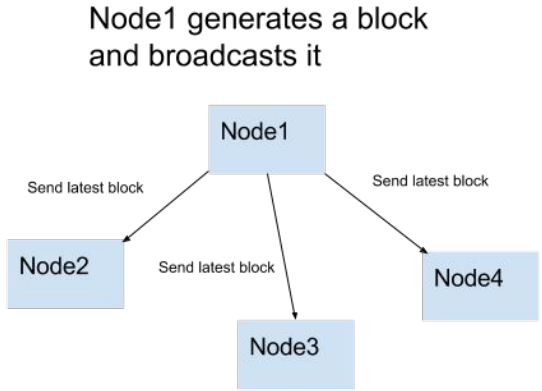
- **Peer-to-peer connection (P2P)** through websockets – **broadcast** & obtain new blocks/ transaction pool/ UTXO
- **Validate new blocks received**: Check if the new blocks that we receive from other miners are valid or not
- **Longest chain rule**: The algorithm to decide which chain to be continued

4.1. Coinbase transaction

```
vincentli@ViNCENT-AxE naivecoin-simple copy % curl http://localhost:3001/peers
[]%
vincentli@ViNCENT-AxE naivecoin-simple copy % curl http://localhost:3002/peers
[]%
vincentli@ViNCENT-AxE naivecoin-simple copy % curl -H "Content-type:application/json" --data '{"peer" : "ws://localhost:6001"}' http://localhost:3002/addPeer
vincentli@ViNCENT-AxE naivecoin-simple copy % curl http://localhost:3001/peers
["::ffff:127.0.0.1:52431"]%
vincentli@ViNCENT-AxE naivecoin-simple copy % curl http://localhost:3002/peers
["127.0.0.1:6001"]%
```

- Connect two Naivecoin programs by using websockets (port: 3001 and 3002)





4.1. Coinbase transaction


```
Received message: {"type":0,"data":null}
Received message: {"type":2,"data":"[{\"index\":68,\"previousHash\":\"32737695b093eba95cb5a59a0c4d16761b20c8087ac342392a9
9dfbdec788891\",\"timestamp\":1648369813,\"data\":[{\"txIns\":[{\"signature\":\"\",\"txOutId\":\"\",\"txOutIndex\":68}],\
"txOuts\":[{\"address\":\"048f6eb168ff45d55e71c1a0ef3d6bfb94b9ec03f6e93d38a3bbb93160ec35f76013feeda65df8ffb6f9391be6d3e52
7aba48b4403f9ab8b7beb8664045a0ac8dc\",\"amount\":50}],\"id\":\"4694c2b2bc81114e86ccc55d0ce207eaea9eccb2db85f88703e2cf13ed
d0d3a4\"}],\"hash\":\"3e91acb4974aa29e545bb8422e9795385e128c8d671985bf2f3f7c6f57d9c8fb\",\"difficulty\":2,\"nonce\":3,\"m
erkleRoot\":\"2e23a415cc0d0e0c83edc6e81e442a103dac7baabb78aa38346f39428393cb26\",\"merkleTree\":{\"roots\":[{\"index\":0,
\"parent\":1,\"hash\":{\"type\":\"Buffer\",\"data\":[46,35,164,21,204,13,14,12,131,237,198,232,30,68,42,16,61,172,123,170
,187,120,170,56,52,111,57,66,131,147,203,38]},\"size\":64,\"data\":{\"type\":\"Buffer\",\"data\":[52,54,57,52,99,50,98,50
,98,99,56,49,49,49,52,101,56,54,99,99,99,53,53,100,48,99,101,50,48,55,101,97,101,97,57,101,99,99,98,50,100,98,56,53,102,5
6,56,55,48,51,101,50,99,102,49,51,101,100,100,48,100,51,97,52]}}],\"blocks\":1}}]"}
blockchain possibly behind. We got: 67 Peer got: 68
replacing unspentTxouts with: [
  [Object],
  [Object],
  [Object],
  [Object],
  [Object],
  [Object],
  [UnspentTxOut]
]
Received message: {"type":3,"data":null}
Received message: {"type":4,"data":"[]"}
```


- Node A received p2p message from Node B

4.1. Coinbase transaction


```
Received message: {"type":0,"data":null}
Received message: {"type":2,"data":"[{\"index\":67,\"previousHash\":\"2416f01ba16fe041154afca2f0220bf7fc588881f0a15788e13
36c2fde32814a\",\"timestamp\":1648280442,\"data\":[{\"txIns\":[{\"signature\":\"\",\"txOutId\":\"\",\"txOutIndex\":67}],\
"txOuts\":[{\"address\":\"049dcc80e4388af21b179920694530e0d6a5290224043ced0a4ebc654daf57a818eff5a9f99f689ea871cdc0803bb33
565384945e78beeb202b9aced0f95905888\",\"amount\":50}],\"id\":\"62eed69e07fea11839c1f56c4c78f843a7fb5ee22414182f6ba9903192
4c2324\"}],\"hash\":\"32737695b093eba95cb5a59a0c4d16761b20c8087ac342392a99dfbdec788891\",\"difficulty\":2,\"nonce\":0,\"m
erkleRoot\":\"b20505a1d5e577a4121dc2d84945a6b5d42ebcfc7981ae9d9f7337d9f499360b\",\"merkleTree\":{\"roots\":[{\"index\":0,
\"parent\":1,\"hash\":{\"type\":\"Buffer\",\"data\":[178,5,5,161,213,229,119,164,18,29,194,216,73,69,166,181,212,46,188,2
52,121,129,174,157,159,115,55,217,244,153,54,11]},\"size\":64,\"data\":{\"type\":\"Buffer\",\"data\":[54,50,101,101,100,5
4,57,101,48,55,102,101,97,49,49,56,51,57,99,49,102,53,54,99,52,99,55,56,102,56,52,51,97,55,102,98,53,101,101,50,50,52,49,
52,49,56,50,102,54,98,97,57,57,48,51,49,57,50,52,99,50,51,50,52]}}],\"blocks\":1}}]"}
received blockchain is not longer than received blockchain. Do nothing
Received message: {"type":2,"data":"[{\"index\":68,\"previousHash\":\"32737695b093eba95cb5a59a0c4d16761b20c8087ac342392a9
9dfbdec788891\",\"timestamp\":1648369813,\"data\":[{\"txIns\":[{\"signature\":\"\",\"txOutId\":\"\",\"txOutIndex\":68}],\
"txOuts\":[{\"address\":\"048f6eb168ff45d55e71c1a0ef3d6bfb94b9ec03f6e93d38a3bbb93160ec35f76013feeda65df8ffb6f9391be6d3e52
7aba48b4403f9ab8b7beb8664045a0ac8dc\",\"amount\":50}],\"id\":\"4694c2b2bc81114e86ccc55d0ce207eaea9eccb2db85f88703e2cf13ed
d0d3a4\"}],\"hash\":\"3e91acb4974aa29e545bb8422e9795385e128c8d671985bf2f3f7c6f57d9c8fb\",\"difficulty\":2,\"nonce\":3,\"m
erkleRoot\":\"2e23a415cc0d0e0c83edc6e81e442a103dac7baabb78aa38346f39428393cb26\",\"merkleTree\":{\"roots\":[{\"index\":0,
\"parent\":1,\"hash\":{\"type\":\"Buffer\",\"data\":[46,35,164,21,204,13,14,12,131,237,198,232,30,68,42,16,61,172,123,170
,187,120,170,56,52,111,57,66,131,147,203,38]},\"size\":64,\"data\":{\"type\":\"Buffer\",\"data\":[52,54,57,52,99,50,98,50
,98,99,56,49,49,49,52,101,56,54,99,99,99,53,53,100,48,99,101,50,48,55,101,97,101,97,57,101,99,99,98,50,100,98,56,53,102,5
6,56,55,48,51,101,50,99,102,49,51,101,100,100,48,100,51,97,52]}}],\"blocks\":1}}]"}
received blockchain is not longer than received blockchain. Do nothing
Received message: {"type":3,"data":null}
Received message: {"type":4,"data":"[]"}
```


- Node B received p2p message from Node A

4.2. Validation of received block

```
const isValidBlockStructure = (block: Block): boolean => {
    return typeof block.index === 'number'
        && typeof block.hash === 'string'
        && typeof block.previousHash === 'string'
        && typeof block.timestamp === 'number'
        && typeof block.data === 'object';
```

- Validate block structure

```
const isValidNewBlock = (newBlock: Block, previousBlock: Block): boolean => {
    if (!isValidBlockStructure(newBlock)) {
        console.log('invalid block structure: %s', JSON.stringify(newBlock));
        return false;
    }
    if (previousBlock.index + 1 !== newBlock.index) {
        console.log('invalid index');
        return false;
    } else if (previousBlock.hash !== newBlock.previousHash) {
        console.log('invalid previoushash');
        return false;
    } else if (!isValidTimestamp(newBlock, previousBlock)) {
        console.log('invalid timestamp');
        return false;
    } else if (!hasValidHash(newBlock)) {
        return false;
    }
    return true;
};
```

- Validate new block

4.2. Validation of received block

```
const isValidChain = (blockchainToValidate: Block[]): UnspentTxOut[] => {
    console.log('isValidChain:');
    console.log(JSON.stringify(blockchainToValidate));
    const isValidGenesis = (block: Block): boolean => {
        return JSON.stringify(block) === JSON.stringify(genesisBlock);
    };

    if (!isValidGenesis(blockchainToValidate[0])) {
        return null;
    }
    /*
    Validate each block in the chain. The block is valid if the block structure is valid
      and the transaction are valid
     */
    let aUnspentTxOuts: UnspentTxOut[] = [];

    for (let i = 0; i < blockchainToValidate.length; i++) {
        const currentBlock: Block = blockchainToValidate[i];
        if (i !== 0 && !isValidNewBlock(blockchainToValidate[i], blockchainToValidate[i - 1])) {
            return null;
        }

        aUnspentTxOuts = processTransactions(currentBlock.data, aUnspentTxOuts, currentBlock.index);
        if (aUnspentTxOuts === null) {
            console.log('invalid transactions in blockchain');
            return null;
        }
    }
    return aUnspentTxOuts;
};
```

- Validate blockchain

4.2. Validation of received block

```
const isValidTxForPool = (tx: Transaction, aTtransactionPool: Transaction[]): boolean => {
    const txPoolIns: TxIn[] = getTxPoolIns(aTtransactionPool);

    const containsTxIn = (txIns: TxIn[], txIn: TxIn) => {
        return _.find(txPoolIns, ((txPoolIn : TxIn ) => {
            return txIn.txOutIndex === txPoolIn.txOutIndex && txIn.txOutId === txPoolIn.txOutId;
        }));
    };

    for (const txIn of tx.txIns) {
        if (containsTxIn(txPoolIns, txIn)) {
            console.log('txIn already found in the txPool');
            return false;
        }
    }
    return true;
};
```

- Validate transaction pool

4.3. Longest chain rule

```
const replaceChain = (newBlocks: Block[]) => {
    const aUnspentTxOuts = isValidChain(newBlocks);
    const validChain: boolean = aUnspentTxOuts !== null;
    if (validChain &&
        getAccumulatedDifficulty(newBlocks) > getAccumulatedDifficulty(getBlockchain())) {
        console.log('Received blockchain is valid. Replacing current blockchain with received blockchain');
        blockchain = newBlocks;
        setUnspentTxOuts(aUnspentTxOuts);
        updateTransactionPool(unspentTxOuts);
        broadcastLatest();
    } else {
        console.log('Received blockchain invalid');
    }
};
```

- It is normal to generate a fork when the transaction is still in unspent transaction output (UTXO) and not yet confirmed in blockchain
- It applies to solve the fork problem and replace the shorter blockchain with longer blockchain

5. Storage

- Stored all raw block data/ UTXO in JSON file locally
- Transaction pool

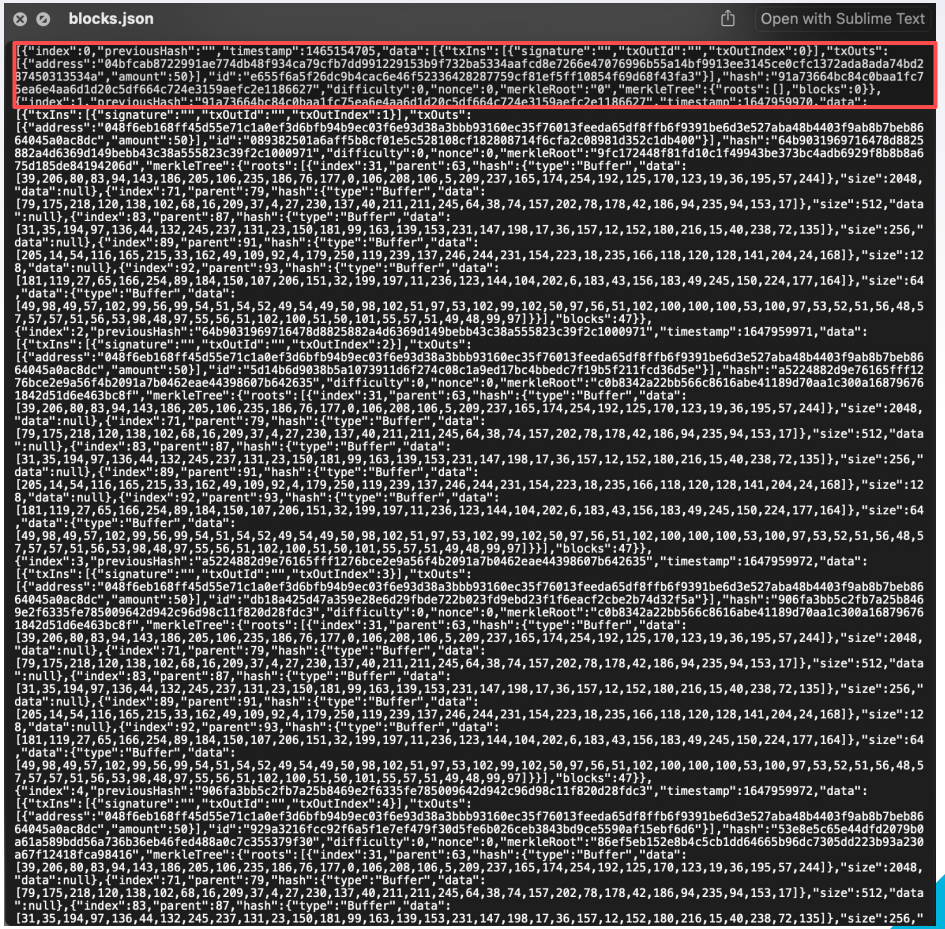
5.1 Storage raw Blockchain data on Disk

```
let blocks_json = fs.readFileSync( filename: "node/data/blocks.json", encoding: "utf-8");
let BLOCKS_FILE = JSON.parse(blocks_json);
```

```
let blockchain: Block[] = BLOCKS_FILE;
```

```
let blocks_json2 = (JSON.stringify(blockchain));
fs.writeFileSync( filename: "node/data/blocks.json",blocks_json2, encoding: "utf-8");
```

- Read block history from the file to a list when the Naivecoin program runs
- Add every newly mined block to the JSON File ⇒ To ensure the data in the JSON File is up-to-date
- E.g. **The highlighted area** indicates the genesis block is being written in the blocks.json file. Whenever the program is restarted, it will load from it



5.2. Storing unspent transaction (UTXO) on disk

unspentTxOut.json — Open with Sublime Text


```
[{"txOutId":"e655f6a5f26dc9b4cac6e46f52336428287759cf81ef5ff10854f69d68f43fa3","txOutIndex":0,"address":"04bfcab8722991ae774db48f934ca79cfb7dd991229153b9f732ba5334aafcd8e7266e47076996b55a14bf9913ee3145ce0cfc1372ada8ada74bd287450313534a","amount":50},
{"txOutId":"862aa44d5268195c80033130c9bc649007de00122fe890ba9da998adda52fa86","txOutIndex":0,"address":"048f6eb168ff45d55e71c1a0ef3d6bfb94b9ec03f6e93d38a3bbb93160ec35f76013feeda65df8ffb6f9391be6d3e527aba48b4403f9ab8b7beb8664045a0ac8dc","amount":50},
{"txOutId":"0ecc8b584ab0c55f502cb8344475b0ee23f8c6c37249f80976be3e94e2d849ac","txOutIndex":0,"address":"049dcc80e4388af21b179920694530e0d6a5290224043ced0a4ebc654daf57a818eff5a9f99f689ea871cdc0803bb33565384945e78beeb202b9aced0f95905888","amount":35},
{"txOutId":"0ecc8b584ab0c55f502cb8344475b0ee23f8c6c37249f80976be3e94e2d849ac","txOutIndex":1,"address":"048f6eb168ff45d55e71c1a0ef3d6bfb94b9ec03f6e93d38a3bbb93160ec35f76013feeda65df8ffb6f9391be6d3e527aba48b4403f9ab8b7beb8664045a0ac8dc","amount":15},
{"txOutId":"7521263e0739addb3551e0b84482d67ee372b184be3e592bec28d5e9e04000c5","txOutIndex":0,"address":"048f6eb168ff45d55e71c1a0ef3d6bfb94b9ec03f6e93d38a3bbb93160ec35f76013feeda65df8ffb6f9391be6d3e527aba48b4403f9ab8b7beb8664045a0ac8dc","amount":50},
{"txOutId":"62eed69e07fea11839c1f56c4c78f843a7fb5ee22414182f6ba99031924c2324","txOutIndex":0,"address":"049dcc80e4388af21b179920694530e0d6a5290224043ced0a4ebc654daf57a818eff5a9f99f689ea871cdc0803bb33565384945e78beeb202b9aced0f95905888","amount":50}]
```


```
let transaction_json = fs_2.readFileSync( filename: "node/data/unspentTxOut.json", encoding: "utf-8");
let TRANSACTION_FILE = JSON.parse(transaction_json);
```

```
let unspentTxOuts: UnspentTxOut[] = TRANSACTION_FILE;
```

```
let unspentTxOuts3 = (JSON.stringify(unspentTxOuts));
fs.writeFileSync( filename: "node/data/unspentTxOut.json",unspentTxOuts3, encoding: "utf-8");
```

- Read unspent transactions from the file to a list when the Naivecoin program runs
- Update the latest unspent transactions list to the JSON file once the unspentTxOuts variable is updated (new blocks are mined) ⇒ To ensure the record is up-to-date