problem_5_blockchain

March 11, 2020

0.0.1 Analyze:

I need to create a private blockchain. luckily I have a Blochchian ND of Udacity. I need to create a Block class and Blockchain class. I chose a list for Blockchain . because I can search Block by height (O(1) time complexity). and the block has been sorted by append order. I need to link each block by hash. so I need to previous block hash linked current block. at last, the Block adds to Blockchain.

The function **add_block()** , **get_block()** takes O(1) time complexity.

I think it's linear space complexity O(n), 1 expanding list.

[ ]:

```
import time

class Block:
    def __init__(self, data):
        self.hash = ""
        self.height = 0
        self.body = data
        self.time = time.time()
        self.previousblockhash = ""

    def __repr__(self):
        return str(self.__dict__)
```

[ ]:

```
import hashlib

class Blockchain:
    def __init__(self):
        self.chain = []
        self.add_block(Block("Genesis block"))

    def add_block(self, new_block):
        new_block.height =  len(self.chain)

        if(len(self.chain) > 0):
            new_block.previousblockhash = self.chain[len(self.chain)-1].hash

        new_block.hash = self._calc_hash(new_block)
        self.chain.append(new_block)

    def get_block(self, height):
        return self.chain[height]

    def _calc_hash(self,data):
        sha = hashlib.sha256()
        hash_str = repr(data).encode('utf-8')
        sha.update(hash_str)
        return sha.hexdigest()

    def __repr__(self):
        return str(self.__dict__)
```

[ ]:

```
# Create a private blockchain, there is a genesis block when inital
private_blockchain = Blockchain()
print(private_blockchain.chain)
```

[ ]:

```
block1 = Block("First Block")
block2 = Block("Second Block")
block3 = Block("Third Block")
block4 = Block("Fourth Block")
block5 = Block("Fifth Block")
block6 = Block("Sixth Block")

print(block1)
print(block2)
print(block3)
print(block4)
print(block5)
print(block6)
```

[ ]:

```
private_blockchain.add_block(block1)
private_blockchain.add_block(block2)
private_blockchain.add_block(block3)
private_blockchain.add_block(block4)
private_blockchain.add_block(block5)
private_blockchain.add_block(block6)

print(private_blockchain)
```

[ ]:

```
print(private_blockchain.get_block(3))
```