CSI2110A

Assignment2

Instructor:

Lucia Moura

By:

Name:Yi Zhao

StudentID: 8650881

Email: yzhao156@uottawa.ca

Miner ID:yzhao156

- My Miner ID: yzhao156
- Brief description of classes and methods

Main Classes:

- Transaction: store (sender, reciver, amount)
- Block: store(index, timestamp, transaction, nonce, previoushash, hash)

1. Has two constructors separately with perimeter hash and without perimeter hash
2. Has method String Nonce() to calculate nonce(proof of work)

- BlockChain:only store(ArrayList<Block> arrayList)

1. Has two constructors separately with perimeter arrayList and without perimeter arrayList
2. BlockChain BlockChain fromFile(String fileName): get information from file(XXXX.txt) and put the information into arrayList to store.
3. void toFile(String fileName): write the information from arrayList into file(ie. xxxx.txt).
4. void toFile(): ask the file and use Scanner to get the file from user then save the file by calling the method void toFile(String fileName).
5. boolean validateBlockchain(): check if (index, previous hash, hash) is correct.
6. void newtransaction(): kind of UI, ask for input and success only if staisfies the following conditions (1)valid sender/reviever/amount . (2)valid amount (sender have enough balence to send the amount of money). (3)valid blockChain(
7. Boolean validateBlockchain() :checks valid index, valid previous hash, valid hash(starts with 00000)),
8. void add(Block block): if blockchain is valid, add block to arraylist.

- Brief description of how to find nonce

To find the nonce with least digits(to same memory), I first create nonce “!”, then add 1 to nonce each time. For example, “!” add 1 to “”, then “#”……”~”. Then find out the digit is”~”, change it to “!”(first one) and then move to higher significant digit, if the digit is not “~”,add one. Else, move to higher significant digit… The method will break until find the nonce such that the hash starts with “00000”. So that the method will find the every possible combination from 1bit until 18bit, if find one valid nonce break.

```
public String Nonce() throws UnsupportedEncodingException{
    String temp;
    String last = "";
    int index = 0;
    int length = 1;
    char valueAtIndex;
    int trail = 0;

    for(int i=1; i<19; i++){ //max length=19 if does't find return null
        index = 0;
        temp="";
        for (int j=0; j<i; j++){
            temp+="!";  // add a new digit
        }
        while(index != -1){
            last = temp.substring(temp.length()-1, temp.length());
            index = i-1;//point to the last index
            setNonce(temp);
            //System.out.println(temp);//for test
            if(Sha1.hash(this.toString()).substring(0,5).equals("00000")){
                System.out.println("trail: "+trail);
                return temp;//if valid return
            }
            valueAtIndex = temp.charAt(index);
            while(valueAtIndex == ('~')){// if last digit is ~, change it to ! and index--(pointer shift left until the index is not~)
                trail++;
                temp = temp.substring(0, index)+'!'+temp.substring(index+1, temp.length());
                if(index == 0){
                    temp = temp.substring(0, index)+((char)(((int)temp.charAt(index))+1))+temp.substring(index+1, temp.length());
                    index--;
                    break;
                }
                if (index != 0){
                    index--;
                    valueAtIndex = temp.charAt(index);//check the pointer index == ~
                }else{
                    break;
                }
            }
            if((index!=-1)) {
                if (temp.charAt(index) != ('~')){//normal case, add "1" to the current digit
                    temp = temp.substring(0, index)+((char)(((int)temp.charAt(index))+1))+temp.substring(index+1, temp.length());
                }
            }
        }
    }
    return null;
}
```

Transaction#	1	2	3	4	5	6	7	8	9	10
Trails	11325	5945	14525	3727	8378	1840	840	9526	366	8445

Average Trail: 6446.7