Homework One

Li Fangwen 1901212601

In my homework, I design a class named “TxHandler” to deal with the transaction got from the system. First, it will get a input of UTXOPool to show the coins have not been used, then when at a epoch, a list of possible transactions will be added into the TxHandler, the method of “handleTxs” will deal with those transactions and use the method of “isvalidTx” to get a list of valid transactions. In the validation, I test that if the signature is correct, if the coin had not been spent and if the output has exceeded the input. The details are in the comment of the code.

In the test part, I design five test plans.

The first is :

*/\*\*  
 \*In this test, we set five test to test  
 \* 1. valid transactions to show basic functions  
 \* 2.Test double-spent  
 \* 3.invalid output number(wrong number of coins to transfer)  
 \* 4.invalid signature  
 \* 5.Test the input not exist in the UtxoPool  
 \* In this part, I use the function "assertTrue", and if there is not error showed,  
 \* it means that the test is success and those function is correct.  
 \* And in each test, it will print that if each transaction is valid  
 \*  
 \*/  
  
// test case 1: valid transactions to show basic functions  
//tx0: Scrooge --> Scrooge 25coins [Create Coins]  
//tx1: Scrooge --> Scrooge 4coins [Divide Coins]  
// Scrooge --> Scrooge 5coins  
// Scrooge --> Scrooge 6coins  
//tx2: Scrooge --> Alice 4coins [Pay separately]  
// Scrooge --> Alice 5coins  
// Scrooge --> Bob 6coins  
//tx3: Alice --> Alice 2coins [Divide Coins]  
// Alice --> Alice 2coins  
//tx4: Alice --> Bob 2coins [Pay jointly]  
// Alice --> Bob 5coins*

The second test is :

*// test case 2: Test double-spent  
//tx0: Scrooge --> Scrooge 25coins [Create Coins]  
//tx1: Scrooge --> Alice 20coins [Pay Coins]  
//tx2: Scrooge --> Bob 20coins [\*Double-spending\*]*

The third test is :

*// test case 3: invalid output number(wrong number of coins to transfer)  
// tx0: Scrooge --> Scrooge 25coins [Create Coins]  
//tx1: Scrooge --> Alice 20coins [Pay Coins]  
//tx2: Alice --> Bob -5coins [\*Invalid output number\*]  
//tx3: Alice --> Bob 90coins [\*Invalid output number\*]*

The forth test is :

*// test case 4: invalid signature  
// signature is not valid  
//tx0: Scrooge --> Scrooge 25coins [Create Coins]  
//tx1: Scrooge --> Alice 20coins [Pay Coins]  
//tx2: Alice --> Bob 20coins [\*Signed by Bob\*]*

The fifthe test is :

*//test case 5: Test the input not exist in the UtxoPool  
//tx0: Scrooge --> Scrooge 25coins [Create Coins]  
//tx1: Scrooge --> Alice 20coins [Pay Coins \*NOT added to UTXO pool\* due to accidents]  
//tx2: Alice --> Bob 15coins [Pay Coins \*Previous Tx NOT in UTXO pool\*]*