区块链实验报告

1. 存储设计

用结构体实现存储，设计block，transaction，input，output四个结构体，数据元素类型按照题目要求。出于简洁考虑，Block中的transaction，transaction中的input和output都用vector存储。此外，为方便之后的判断，在transaction中增加“int illegal”项，用来记录交易是否违法，在output中加入“int used”项，用来记录该input是否已经被使用。

Block按照单链表方式来进行存储。

1. 数据读入

设计四个函数顺序分别读入block，transaction，input和output，存入对应位置。

1. 非法交易判断

对于每个交易：

初始化inputValue=0，遍历各个input，用函数countInput（）来计算input的总value值。

countInput（）的返回值为所有input对应output的value，同时根据不同的非法情况返回特定值如0，-1，-2来标记引用transaction非法，引用output已经被使用，引用output无法找到等情况。并对其进行特判，如果判断为非法情况，将目前交易的illegal赋值为1，输出非法字串和非法原因。

遍历output，得到总的outputValue。

比较inputValue和ouputValue的大小，如果inputValue大于outputValue，仍然为非法，illegal赋值为1，输出非法id和非法原因。

此部分源码：

int illegal(block \*blocks)

{

    int count=0;

    block \*head=blocks;

    while(blocks!=NULL)

    {

        for(auto &transactions :blocks->transactions)

        {

            long long inputValue=0;

            long long outputValue=0;

            if(transactions.is\_coinbase)continue;

            inputValue=countInput(transactions,head);

            if(inputValue==-1)//-1 代表引用的交易非法

            {

                transactions.legal=0;

                count++;

                cout<<"id of illegal transactions: "<<transactions.txid<<endl;

                cout<<"Height:"<<blocks->Height<<endl;

                cout<<"reanson is using an illegal output"<<endl;

                continue;

            }

            else if(inputValue==-2)//代表input已经被使用。

            {

                transactions.legal=0;

                count++;

                cout<<"id of illegal transactions: "<<transactions.txid<<endl;

                cout<<"Height:"<<blocks->Height<<endl;

                cout<<"reason is using a used input"<<endl;

                continue;

            }

            else if(inputValue==0)//代表找不到交易

            {

                transactions.legal=0;

                count++;

                cout<<"id of illegal transactions: "<<transactions.txid<<endl;

                cout<<"Height:"<<blocks->Height<<endl;

                cout<<"reason is can't find the input"<<endl;

            }

            else{

            for(auto output:transactions.outputs)

            {

                outputValue+=output.value;

            }

            if(outputValue>inputValue)

            {

                count++;

                cout<<"id of illegal transactions: "<<transactions.txid<<endl;

                cout<<"Height:"<<blocks->Height<<endl;

                cout<<"reanson is input value less than output value"<<endl;

                cout<<"(output value: "<<outputValue<<">input value: "<<inputValue<<")"<<endl;

                transactions.legal=0;

            }

        }}

        blocks=blocks->nextblock;

    }

    return count;

}

long long countInput(transaction transactions,block \*hd)

{

    int value=0;

    for(auto &in: transactions.inputs)

    {   int heigt=in.pre\_block;

        block \*blocks=hd;

        string id=in.prevTxID;

        while(blocks->Height!=heigt&&(blocks->nextblock!=NULL)){//找到preHeight对应区块

            blocks=blocks->nextblock;

        }

        int flag=0;

        for(auto &tr :blocks->transactions)//遍历区块的交易，找到所需要的。

        {

            if(tr.txid==id)

            {flag=1;

                if(tr.legal)

                {

                    if(!tr.outputs[in.prevTxOutIndex].used){

                    if(in.prevTxOutIndex<tr.output\_count)

                    {value+=tr.outputs[in.prevTxOutIndex].value;

                    }

                    tr.outputs[in.prevTxOutIndex].used=1;}

                    else if(in.prevTxOutIndex>=tr.output\_count){

                        return 0;

                    }

                    else{

                        return -2;

                    }

                }

                else{

                    return -1;

                }

            }

        }

        if(flag==0)

        {

            return 0;

        }

    }

    return value;

}

1. 交互界面设计

首先读入数据输出基本信息:

总区块数，总交易数，非法交易ID和原因，非法交易数，合法交易数和用户输入提示。

用户可输入1，2，3分别代表输入高度，ID和退出。

如输入1，之后再输入高度，可输出block中所有交易信息。

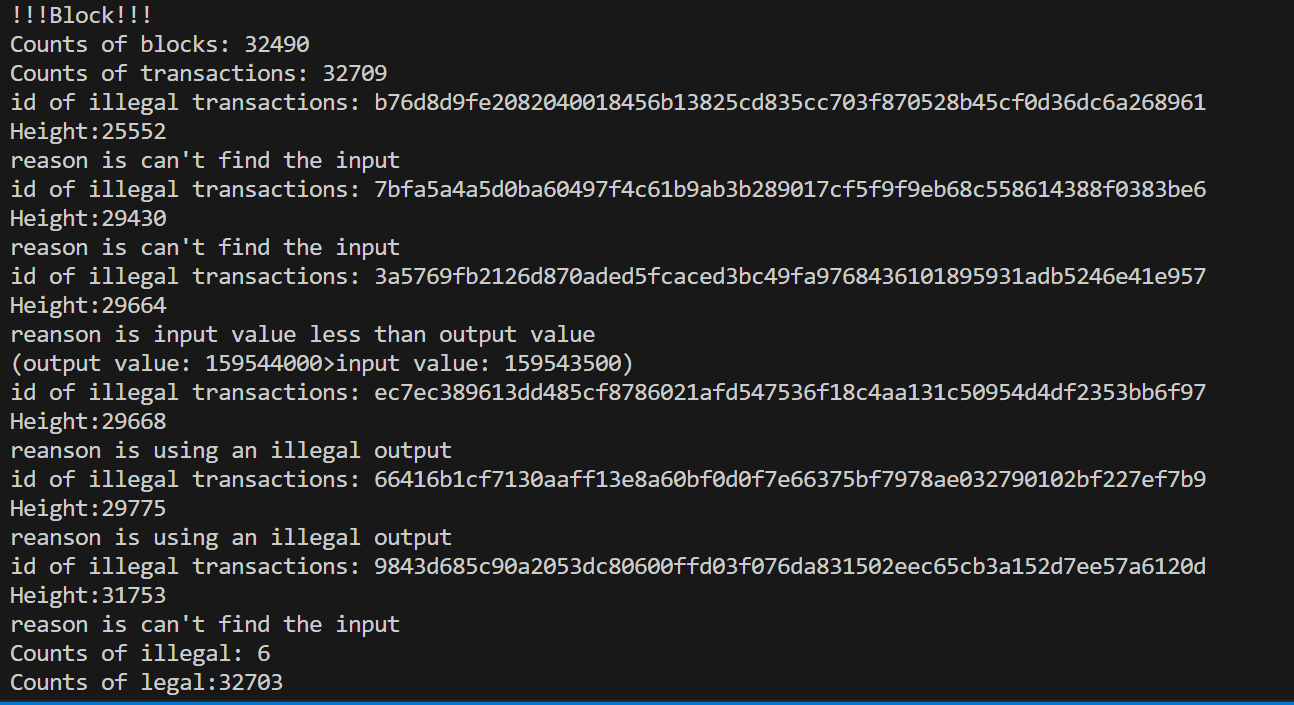
如输入2，之后再输入ID，可得到特定id的交易信息，同时如果找不到会提示未找到。

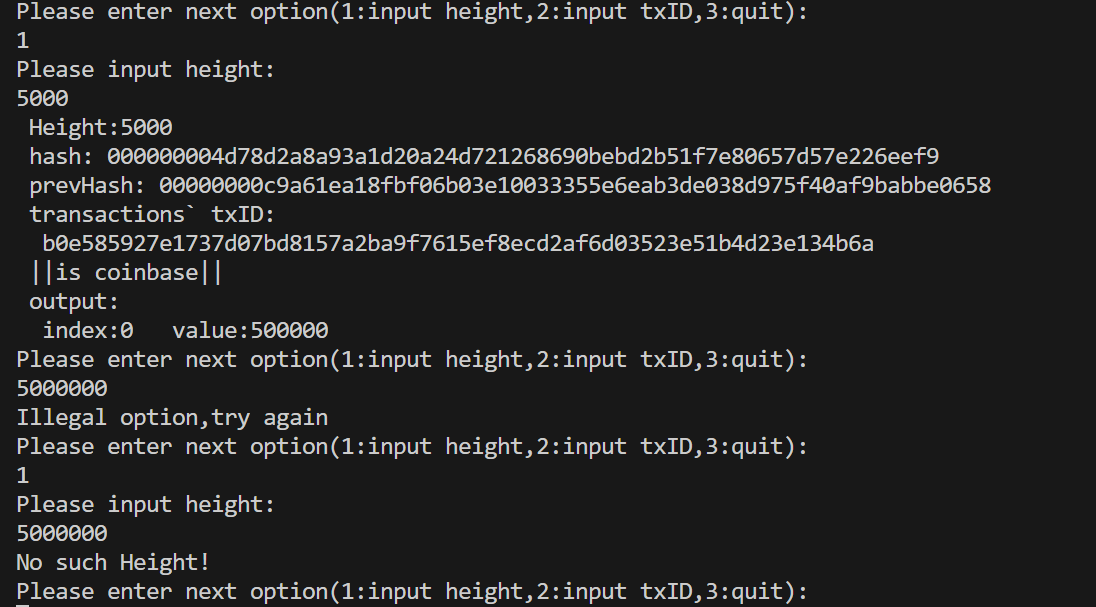
如输入3，结束程序。

输入其他数字，提示操作非法，重新输入。

1. 输出截图

显示基本信息：



功能1，输出高度对应信息：

功能2，输出ID对应交易：

