一，功能说明

实现了两个基于EVM的两个区块链（本地网络测试链chainA和测试链chainB）之间的跨链转账。采用哈希时间锁的的方式实现。

二，实现原理

**哈希时间锁原理**：所谓哈希时间锁就是利用hash值和时间T锁住资产的合约。要取走合约中锁定的资产需满足两个条件中一个：一，在没超过时间T之前调用哈希时间锁合约，由合约指定的receiver账户调用合约相关方法，输入参数key，合约判断sha256（key）==hash，则可以取走资产；二，超过了时间T,则资产的锁定账户可以调用合约相关方法，提走资产。

**利用哈希时间锁转账的原理**：举例说明。假如Alice想用chainA链上的10个eth换Bob在chainB链上的10个eth。Alice会先构造Secretkey，并计算hash=sha256(Secretkey)。选择一个未来时间T1，利用hash和T1在chainA链上锁定自己账户上的10个eth，并把hash和T1发给Bob。Bob在chainA链上验证了hash和T1的真假后，也设计一个时间T2(now<T2<T1)，并用hash和T2在chainB链上锁定了自己账户上的10个eth。这里T2的选择要注意，T2距离当前时间要足够长，避免Alice没有足够的时间从chainB上提取资产，其次，T2距离T1的时间要足够长，避免在Alice提走chainB上的资产后，Bob自己没有足够的时间从chainA上提走锁定的资产。在Bob在chainB上锁定好资产后，Alice因为有Secretkey，所以Alice可以调用chainB链上的合约提走chainB链上的eth。在Alice提币后，因为Bob监听了Alice的提币事件，所以Bob马上就知道了，Bob调用合约查看了Alice输入的Secretkey，也知道了密码，然后Bob可以调用chainA上的合约输入密码提走chainA上的资产。如果在T2到了之后，Alice没有输入密码取走资产，那Bob也不知道密码，那Bob在T2超时后可以取走自己的资产，Alice在时间T1超时后可以提走自己的资产。双方都没有损失（除了少量的gas费用）。

三，项目结构

**1，智能合约部分**：

Migrations：迁移合约

HashedTimeLock：哈希时间锁合约

|------ newTransaction 锁定资产方法

|------withdraw 提取资产方法

|------getTransaction 获取锁定资产条目信息

|------refund 时间超时后取回资产的方法

**2，web部分**：

Web采用vue实现，有四个页面，分别进行newTx，getTx，withdraw，refund功能操作

src

|--- router //路由

|---index.js

|--- Utils //工具方法

|---index.js

|--- Views

|---getTx.vue

|---newTx.vue

|---refund.vue

|---withdraw.vue

|--- App.vue

|--- main.js