Homework 5

Here is the UI for a snapp that will implement the hash chain proof of age that we have discussed
You can go to the repo for the file

```
import {
Field,
SmartContract,
state,
State,
isReady,
Mina,
Party,
PrivateKey,
method,
UInt64,
shutdown,
Poseidon,
} from 'snarkyjs';

export async function run() {
await isReady;

const Local = Mina.LocalBlockchain();
Mina.setActiveInstance(Local);
const account1 = Local.testAccounts[0].privateKey;
const account2 = Local.testAccounts[1].privateKey;
const account3 = Local.testAccounts[2].privateKey;

const snappPrivkey = PrivateKey.random();
const snappPubkey = snappPrivkey.toPublicKey();

let snappInstance: AgeProof;
let randomSeed = Field.random();
let yearOfBirth = 1996;
let minimumYear = 2004;
//deploy the snapp
await Mina.transaction(account1, async () => {
// account2 sends 1000000000 to the new snapp account
const amount = UInt64.fromNumber(1000000000);
const p = await Party.createSigned(account2);
p.balance.subInPlace(amount);
snappInstance = new AgeProof(snappPubkey);
snappInstance.deploy(amount);
})
```

```
.send()
.wait();
console.log(
'snapp balance after deployment: ',
(await Mina.getBalance(snappPubkey)).toString()
);
await Mina.transaction(account2, async () => {
snappInstance.createHashChainProof(randomSeed, yearOfBirth);
})
.send()
.wait();
let difference = minimumYear - yearOfBirth;
let proofOfDiff = hashNTimes(difference,Poseidon.hash([randomSeed]));

const a = await Mina.getAccount(snappPubkey);
console.log('hash of the age is:', a.snapp.appState[0].toString());
try {
await Mina.transaction(account3, async () => {
snappInstance.verifyIfBornBefore(minimumYear,proofOfDiff);
})
.send()
.wait();
} catch (e) {
console.log(e);
}
}

run();
shutdown();
```