Engineering Blockchain and Web3 Apps
Programming Assignment: Merkle Tree

In this project you will implement a Python function to generate a Merkle proof. Please download the starter code in `proj-merkle.zip`. In it you will find three files:

1. `prover.py`: This python script first generates a list of a thousand strings that will make up the root of a Merkle tree. It then calls a function `gen_merkle_proof()` to generate a Merkle proof for one of the leaves (leaf #743). Finally, it writes the proof to a text file `proof.txt`. **Your job is to implement the function** `gen_merkle_proof()`. The missing code in that function can be implemented in less than ten lines of Python.
2. `verifier.py`: This python script contains a hardcoded Merkle root for the Merkle tree generated by the prover. The script loads the proof in `proof.txt` and verifies it with respect to the hardcoded Merkle root. You should not make any changes to this file. However, you should familiarize yourself with the function `compute_merkle_root_from_proof()`, which is the core of the verifier. This will help you implement the function `gen_merkle_proof()`. Your prover will need to generate a Merkle proof that is accepted by this verifier.
3. `proof-for-leaf-95.txt`: is an example Merkle proof (for leaf #95) that is accepted by the verifier. Your task is to generate a proof file like this for leaf #743.

After you implement the function `gen_merkle_proof()` in `prover.py`, running both scripts one after the other should generate the following output:

```
$ python3 prover.py

    I generated 1000 leaves for a Merkle tree of height 10.
    I generated a Merkle proof for leaf #743 in file proof.txt

$ python3 verifier.py

    Have hardcoded root of committed Merkle tree.
    I verified the Merkle proof: leaf #743 in the committed tree is "data item 743".
```

Try changing one character in `proof.txt` and check that the verifier now rejects the proof.

Note: there are many implementations of Merkle trees on the web. However, it is more fun, and certainly more instructive, to implement the prover yourself.

Deliverables. Please submit your file `prover.py` via CourseWorks.

Additional reading. We discussed Merkle trees in class, but if you want to read more about them, then a good resource is `https://decentralizedthoughts.github.io/2020-12-22-what-is-a-merkle-tree/`.