*Project5: Impl Merkle Tree following RFC6962

代码说明：

此项目是遵循 RFC6962 实现：

（1）构造一个具有 100000 个叶节点的 Merkle 树

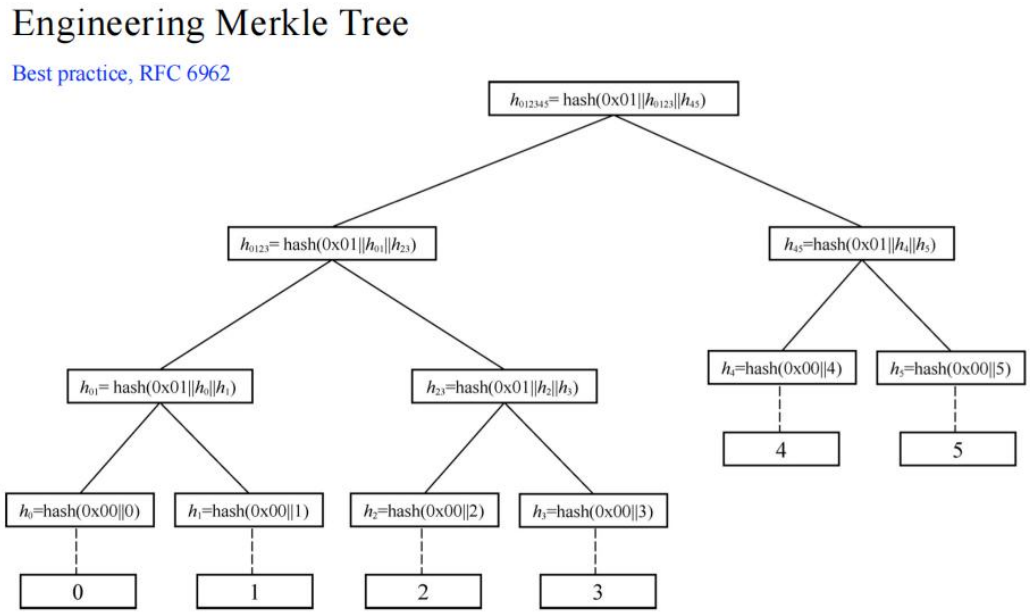
（2）证明某个元素在这个 Merkle 树里

（3）证明某个元素不在这个 Merkle 树里

Merkle 树的建立：

Merkle Tree 是一种基于哈希值构建的树状数据结构。它通过对数据块进行递归地哈希运算，从而生成一个具有高效验证和安全性特性的树。

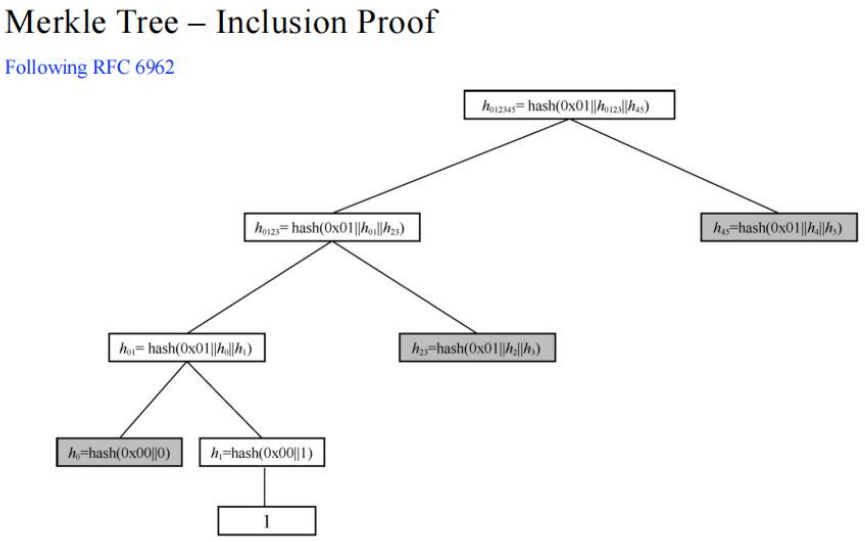
遵循 RFC6962 的 Merkle 树要按照以下规则构建（下图以 6 个叶节点为例建立 Merkle 树）：



在本项目中我创建了一个具有 100000 个叶节点（h_0—h_{99999}）的 Merkle 树。

存在性证明：

对于随意选择的一个结点（函数 inclusion 里参数可改成 0——99999 中的任意一个整数，证明其在 Merkle tree 里），我可以根据其的 hash 值以及树中的所有相关节点（如下图阴影所示）计算出一个根值，与建立的 Merkle tree 的根值进行比较。如果它们是相同的，那它存在于 Merkle 树中。



排除性证明：

排除性证明只能用于叶节点 hash 前的数是紧挨着递增或递减的 Merkle tree。

对于随意选择的一个结点（函数 inclusion 里参数可改成 0——99999 中的任意一个非整数，证明其不在 Merkle tree 里），我可以根据验证其下取整的整数和其上取整的整数都在 Merkle tree 里来证明其不在 Merkle tree 里。

实现方式：python

效果：在自己电脑上 CPU：11 代 i7

创建了一个具有 100000 个叶节点（h_0—h_{99999}）的 Merkle 树，并输出其根的 hash 值。证明了 6 在 Merkle tree 里。证明了 12.5 不在 Merkle tree 里。

```
============================ RESTART: E:\project5.py ==================
创造一个有100000个叶节点的Merkle tree
创造成功！
Merkle tree的根的hash值：
e486114c73f9da0c288e8dbf045f51416f412c54be0e17b99fa8632cadaff8f6
证明6在Merkle tree中：
证明成功！ 6 在Merkle tree里
证明12.5不在Merkle tree中：
证明成功！ 12.5不在Merkle tree里
```

还可以输出创建的 Merkle tree 的结点，但是结果太长，代码里注释掉了那部分，若想查看这个有 100000 个叶节点的 Merkle tree 的每个结点，解注释这段即可：

```
'''
print("Merkle tree的结点：")#结果太长，这里注释掉了，若想查看
for i in tree.children:
    print(i)
    print(i.value)
    print(i.hash)
    print("$$$$$$$$$$$$$$$$$$")
'''
```

这里展示 Merkle tree 的结点的一部分：

```
Merkle tree的结点：
<__main__.child object at 0x0000022C13F830D0>
0
e47a5bb02f714db3924da5104de9132fbcd6c6f14df4dd5031311f01772fa9a1
$$$$$$$$$$$$$$$$$$
<__main__.child object at 0x0000022C13F82FB0>
1
702059e92e6808a703e1fbbd055ab3871edef232299ef5ea62d623bb521c0ef7
$$$$$$$$$$$$$$$$$$
<__main__.child object at 0x0000022C13F82F50>
2
cbb967d83e033221f76e01d00785f85cd4163859a2db7d2417352e9aaf63ccb0
$$$$$$$$$$$$$$$$$$
<__main__.child object at 0x0000022C13F83700>
3
70f9e76ea8bbcc34415d2c697627f4ff96f28c03a9a23d083506fc2978d8113d
$$$$$$$$$$$$$$$$$$
<__main__.child object at 0x0000022C13F83760>
4
b969836947c085159b16c403f1bb02019095333e53afc15e044cd3c10d5e1e52
$$$$$$$$$$$$$$$$$$
<__main__.child object at 0x0000022C13F83A30>
5
055b08b696e5e54e8712c67c9a7c26751e285d4c5fede0f4cf9cdb03d922d6aa
$$$$$$$$$$$$$$$$$$
<__main__.child object at 0x0000022C13F839D0>
6
b27a7f979034bf306fbe10e7d868ad4b14a0b4da4c4226945374ada2b53c6fb3
$$$$$$$$$$$$$$$$$$
<__main__.child object at 0x0000022C13F83970>
7
0d70e41b29ee5e02eb54be148e9f03c79d5fbf23e194785daecefe665c2f0ea0
$$$$$$$$$$$$$$$$$$
<__main__.child object at 0x0000022C13F83910>
8
caa8d6e357442bd7c28fd5a6666ac110cf06af33e9a2a7ebcbf00a65478aa025
$$$$$$$$$$$$$$$$$$
<__main__.child object at 0x0000022C13F82CB0>
9
f1f16a835dc1e1bd66b627c99f8d5e70a8fc3b9faa56d2dfd12646b5cc66f681
```

分工：自己独立完成