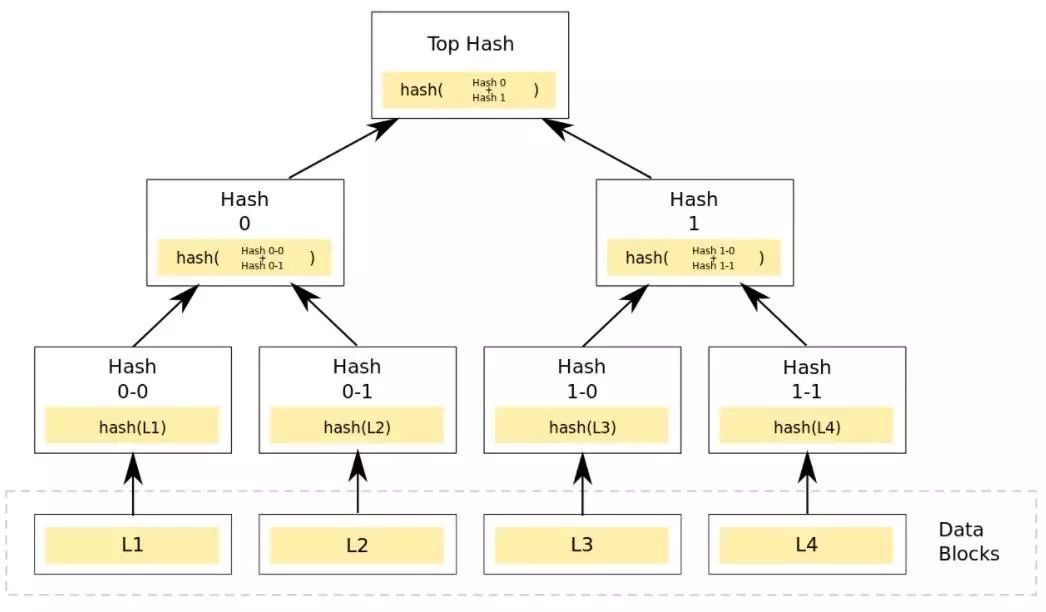
# project-5

# Impl Merkle Tree following RFC6962

Merkle trees是区块链的基本组成部分，通过以二叉树的形式保存区块数据节点的hash值的形式，可以快速归纳和校验区块数据的存在性和完整性。

如图所示，Merkle trees的叶子节点保存每个区块数据的hash值，而父节点保存其两个子节点的hash值之后的hash值，这样保证了区块数据难以被更改，保证了数据的完整性和安全性。



# 实验步骤

创建：本次实验中，我们先用random.randint生成10万的数据作为区块数据，再据此生成merkle tree。

检查：当我们检查或者证明某个节点是否存在时，只需给出该节点的相邻兄弟节点和父节点即可，具体方式是对于待证明的节点，我们只需要确切找出它的hash值是否与树中某节点hash值相同以及该节点在树中的位置，可以求解其在list中对应的下标n；

我们需要讨论待求是偶节点还是奇节点。根据其下标，如果是奇数：首先检查是否是最后一个节点。 如果是，则不操作，进入下一个循环。

（即本层没有该节点的兄弟节点，即在Merkle Tree的逻辑表示中，该节点并不位于本层，而是在更高的层。）

找到该节点之后，将其右节点（下标+1的节点）与该节点一起写入列表。 如果是偶数：将其左节点（下标为-1的节点）与该节点一起写入列表；之后我们就可反复地调用上面的程序，最后将根节点放入树中。

验证是否其父节点的hash值等于hash（子节点的hash值之和），然后再照此验证其祖父节点。此外我们还应该检查该树是否符合merkle tree的定义结构。

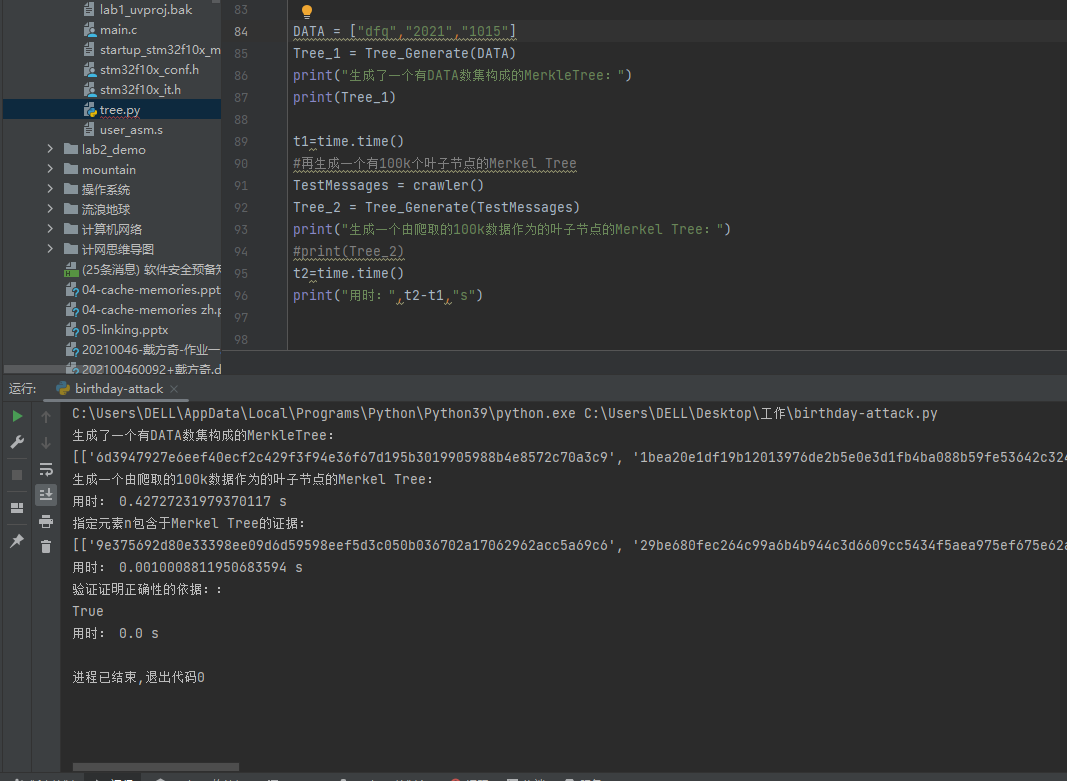
# 实验环境

Windows10

PYCHARM 2022

CPU：11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz

# 实现效果

结果如图：

其中生成一个100k数据的merkle tree 用时0.4s，

给出节点的存在性证据用时0.001s，

验证的时间可以忽略不计，很小。

# 小组分工：

戴方奇202100460092个人完成project5