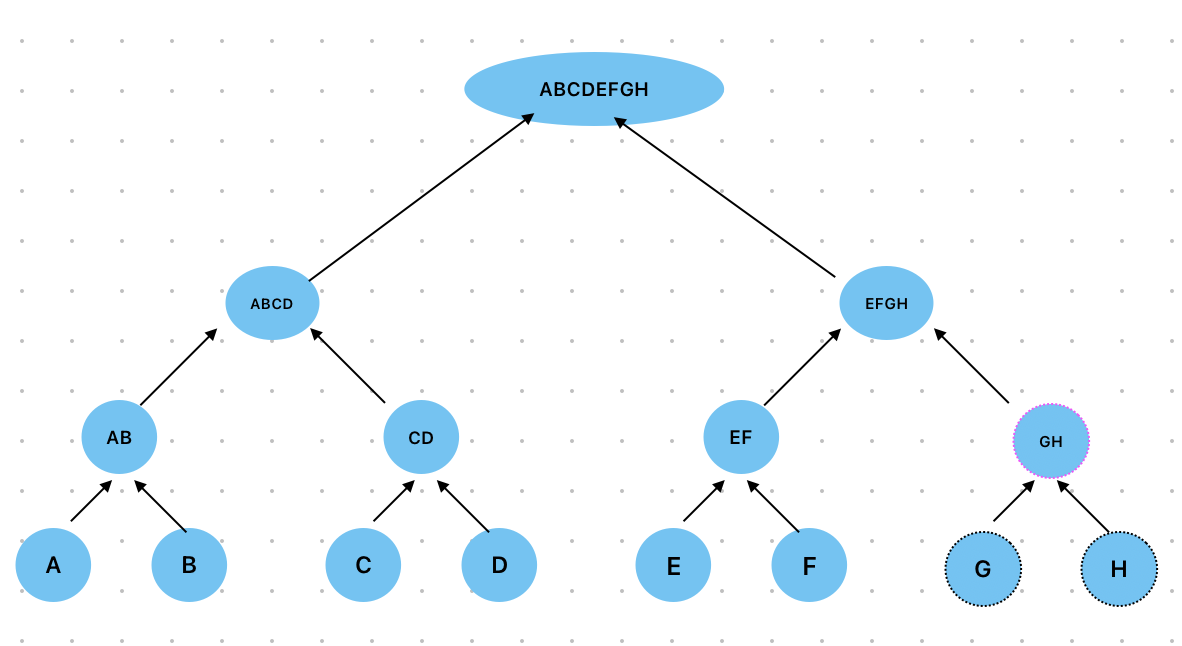
# 白名单证明问题

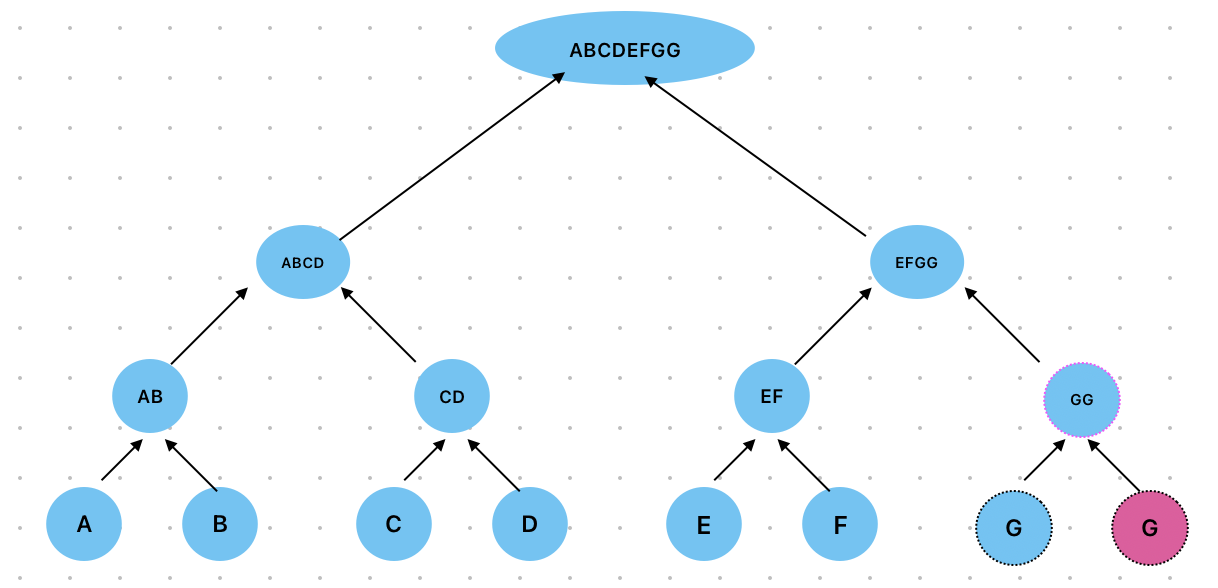
***⚫ 问题描述***：

默克尔树是一个完全二叉树， 其作用是快速归纳和校验区块数据的存在性和完整性。它的每个节点都是哈希（散列）值，其中叶子节点存放的是原始数据的哈希值，原始数据可以是交易数据，也可以是账户地址。具体构造方式如下图M-1所示，A~H节点是叶子节点，也就是原始信息的哈希值，AB代表将A和B的哈希值合并（字符串拼接）后计算得到新的哈希值，CD代表将C和D合并后计算得到的新哈希值，同理，ABCD代表将AB和CD合并后计算的哈希值，默克尔树的构建方式就是哈希值向上传递。

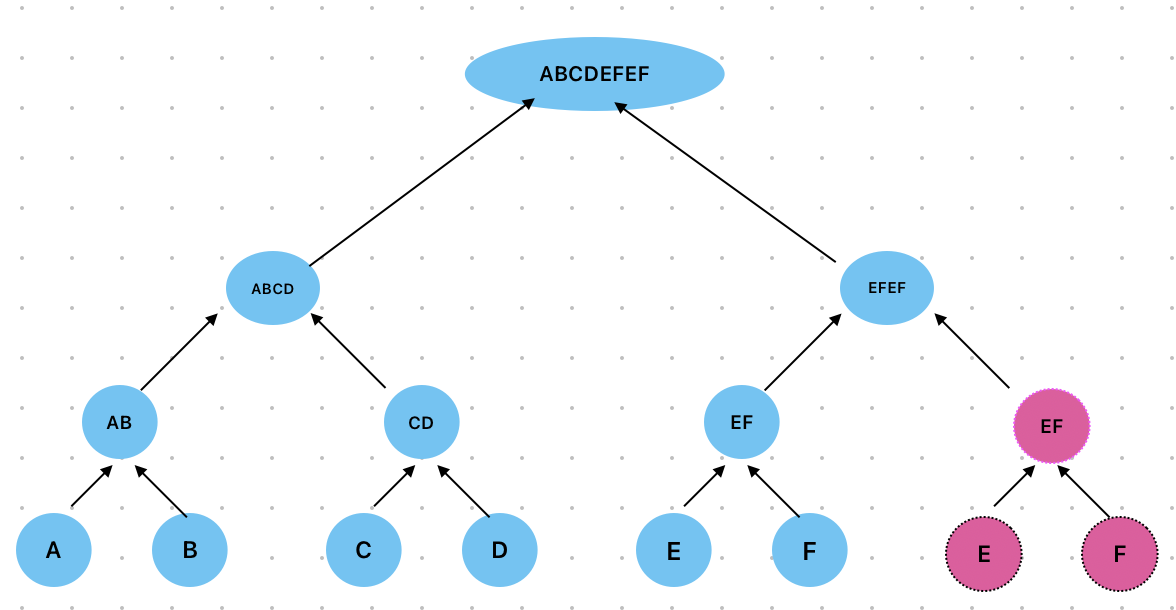


图M-1 完全默克尔树

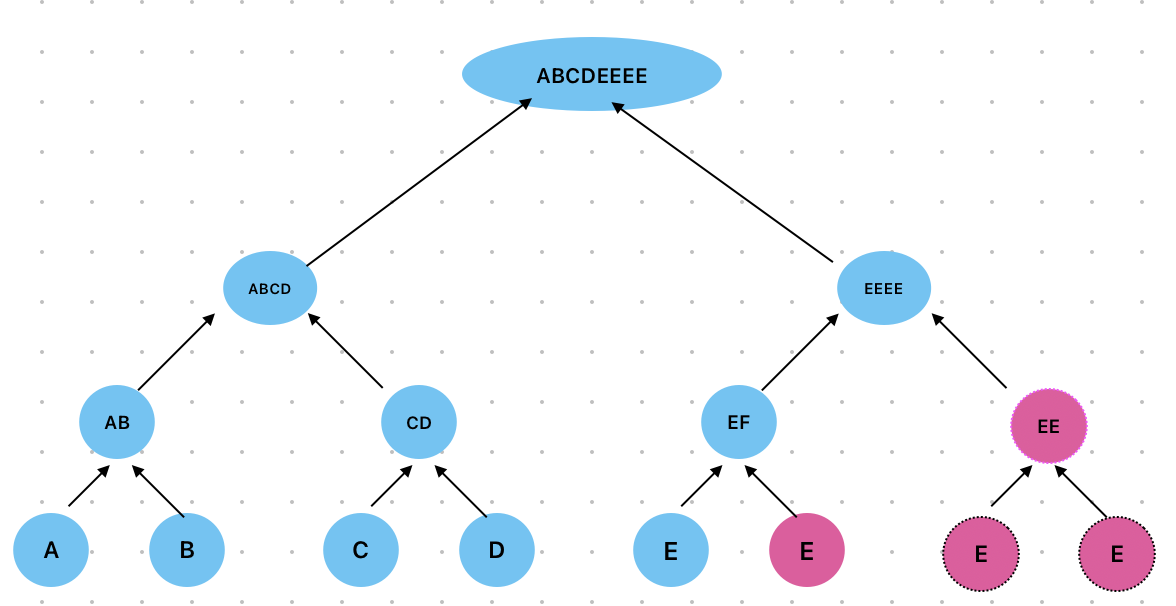
一些特殊情况也需要被考虑，当叶子节点数量无法达到2的整数次幂时，需要用最后的叶子节点来补齐，基本原则是计算一个新的父节点时，如果右子节点缺失，则右子节点直接用左子节点补齐。例如下图M-2，图M-3，图M-4是几种补齐方式，其中紫色节点代表是补齐的节点。



图M-2 节点补齐方式（一）

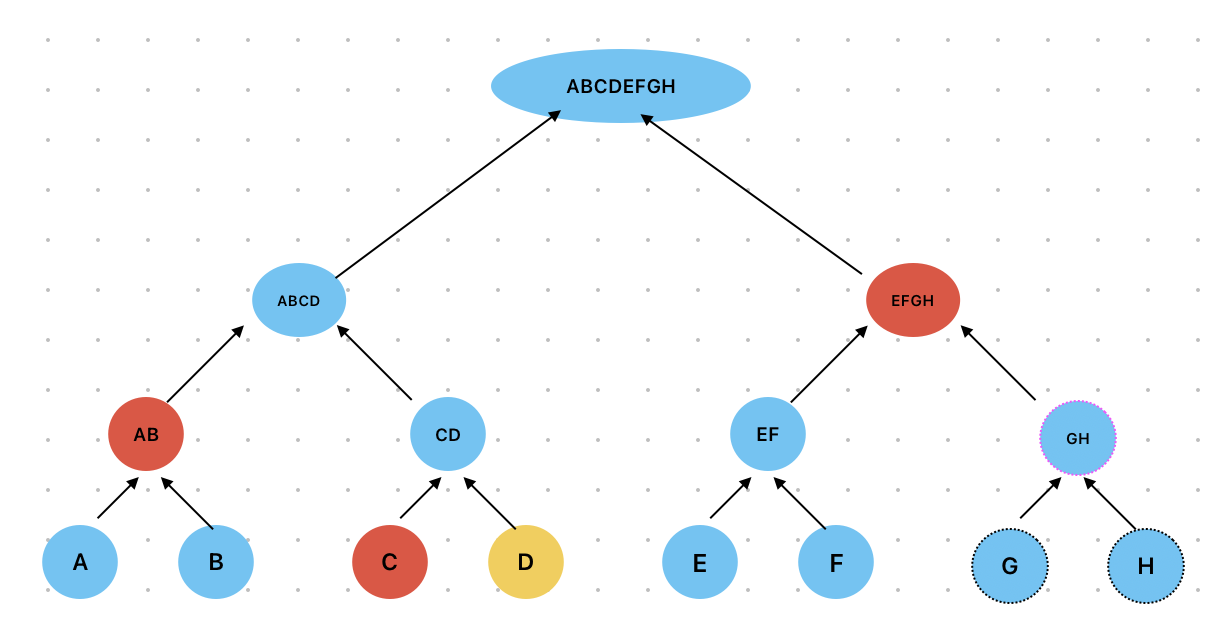


图M-3 节点补齐方式（二）



图M-4 节点补齐方式（三）

在发行NFT（非同质化Token）时，通常会设置一些白名单用户，这些白名单用户的账户地址需要在智能合约内设置，但如果每个地址都设置一次白名单会造成大量的合约交互而产生高昂的费用。通用的解决办法是使用默克尔树来提供一个证明，证明方式由下图M-5所示，若要证明D节点在默克尔树种，则需要提供一个数组[C,AB,EFGH]，在已知D的情况下，提供C可以结算CD，再与AB结合可以计算ABCD，再与EFGH就可以计算得到根。验证者只需按照这样的计算方式就可以得到一个新的根节点，若结算结果与合约中预先设置的根节点是一致的，则代表D节点在默克尔树中。



M-5 默克尔树的证明路径

⚫**要求**：

根据输入的若干账户地址，为指定白名单账户地址生成一个默克尔树证明。

**特别提示：**验证程序会根据证明来计算最终根，因为计算AB和BA哈希值的结果不同，验证程序会判断A>B时使用A+B，反之使用B+A来计算哈希值。

题目中统一采用SHA256作为哈西函数，参考如下。

**python**：

import hashlib

# address代表要计算的内容，target为哈希值

target = hashlib.sha256(address.encode("utf-8")).hexdigest()

⚫ 样例：

◼ 样例 1

输入：

['0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db',

'0x78731D3Ca6b7E34aC0F824c42a7cC18A495cabaB',

'0x617F2E2fD72FD9D5503197092aC168c91465E7f2',

'0x17F6AD8Ef982297579C203069C1DbfFE4348c372'],

'0x78731D3Ca6b7E34aC0F824c42a7cC18A495cabaB'

输出：

['f0dde07f0dde3868468860f5617988fbf30d993e330c755bf0eb49cef2287dd2', '5fe3502257b199fee23fe49724a5139a4b90e4c653bf605576137fccc1a6cd23']

◼ 样例 2

输入：

['0x17F6AD8Ef982297579C203069C1DbfFE4348c372',

'0x5c6B0f7Bf3E7ce046039Bd8FABdfD3f9F5021678',

'0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7',

'0x1aE0EA34a72D944a8C7603FfB3eC30a6669E454C',

'0x0A098Eda01Ce92ff4A4CCb7A4fFFb5A43EBC70DC',

'0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c'],

'0x0A098Eda01Ce92ff4A4CCb7A4fFFb5A43EBC70DC'

输出：

['9507a7723f3b892f6b6748a38d674426a45d5ffe61fda2aa6cbfa0766018127c', 'ea64e21b2f0054ecc30ed10193f3929868bbe95c75071c1657a6b777a9cf038c', '88bb57733a97615c7b1534c210aa1554055a7a89b34981b7d8f058df54728251']