

IPFS 去中心化数据结构

IPFS原力区

价值 共建 共享 荣耀

IPFS 去中心化数据结构

- 什么是数据结构
- 去中心化数据结构

什么是数据结构?

来自维基百科


在计算机科学中，数据结构是一种数据组织，管理和存储格式，可以实现高效的访问和修改。更确切地说，数据结构是数据值的集合，数据之间的关系，以及可以应用于数据的功能或操作。

LESSON 1  Data structures

LESSON 2  The centralized web: Location-based addressing

LESSON 3  The decentralized web: Content addressing

LESSON 4  Cryptographic hashing and Content Identifiers (CIDs)

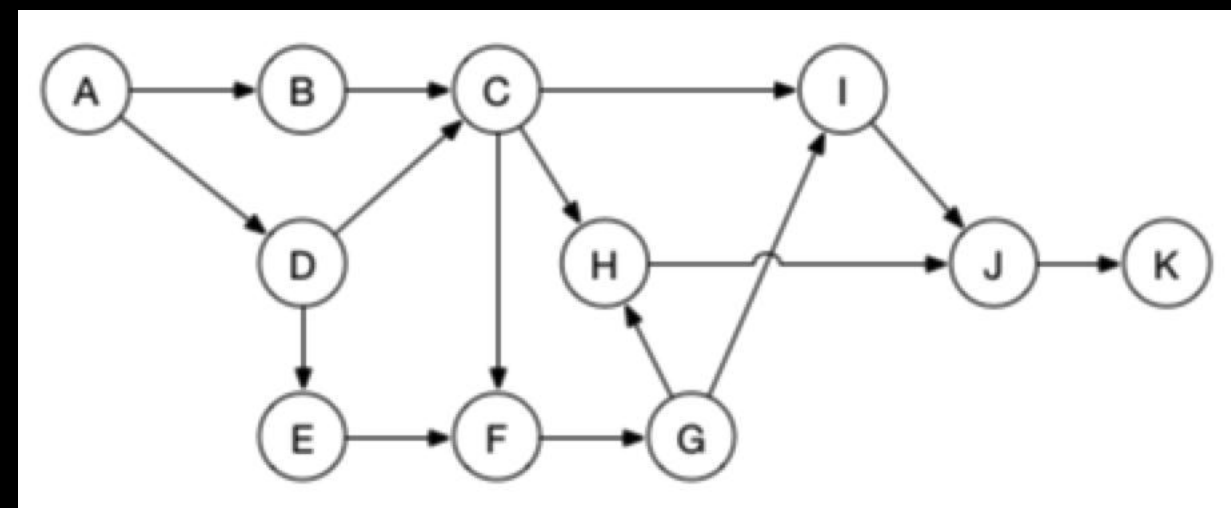
LESSON 5  Merkle trees and directed acyclic graphs (DAG)

RESOURCES  More to explore

去中心化数据结构

可验证（信任度）

大型数据结构的灵活性



位置寻址和中心化网络

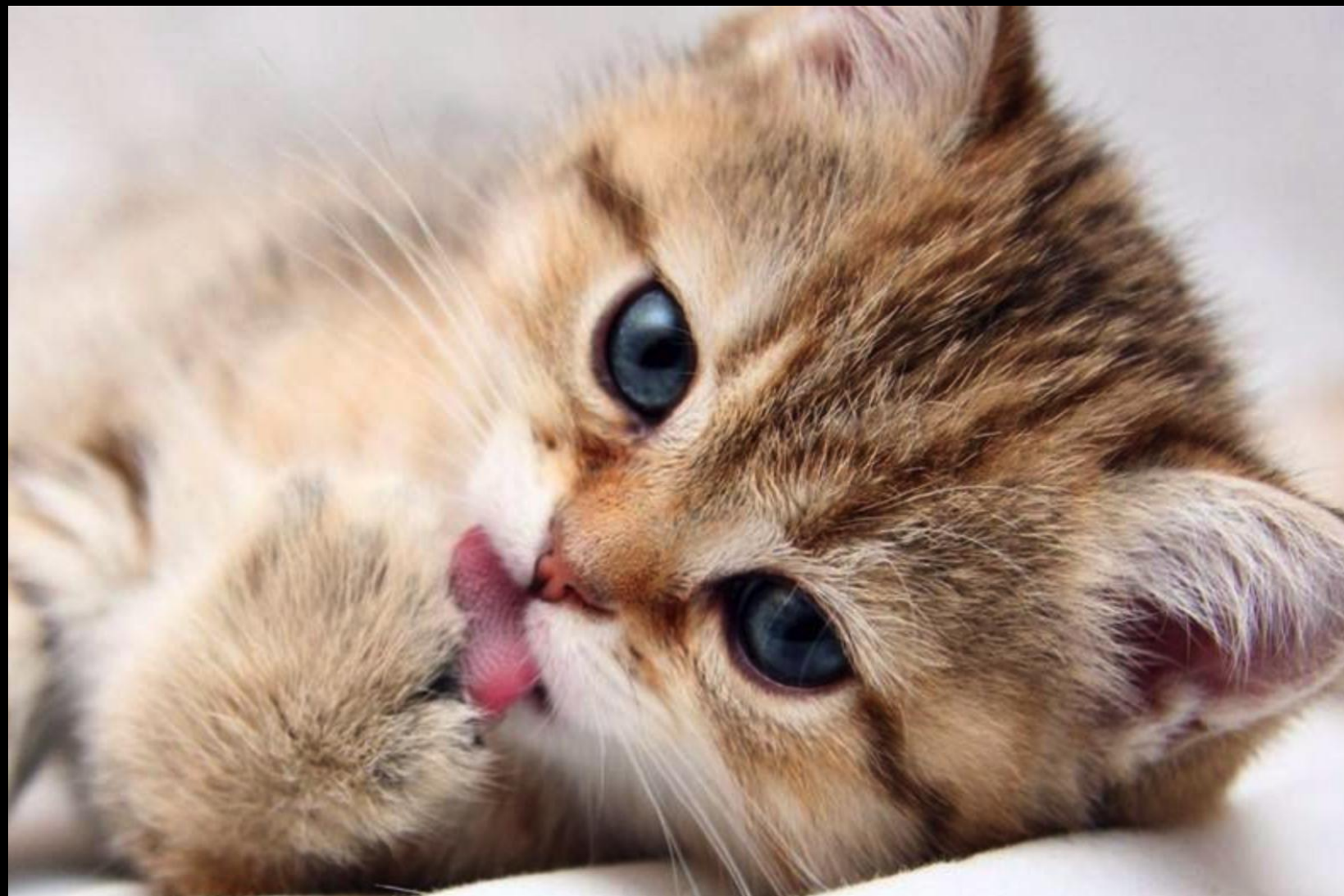
- 使用URL进行位置寻址
- 中心化Web上的信任和效率

使用URLs进行位置寻址

这种访问方式使我们在网络上创建链接和获取数据成为可能，因此它们有助于实现有价值的目的。

URLs是基于存储数据的位置实现的，而不是基于存储在那里的资源的内容。我们称这个为位置寻址，它给我们带来了一些问题。

例如，当我们看到<https://www.puppies.com/beagle.jpg>时，我们可能会从文件名和扩展名猜测存储在该位置的数据是小猎狗的图像（JPEG格式），但是我们无法仅通过URL验证这一点。很可能有一张藏在beagle.jpg的吉娃娃的照片，甚至更糟糕的是，一只可爱的小猫！



<https://www.puppies.com/beagle.jpg>

托管在中心化Web上的**文件内容与其基于位置检索的地址没有直接关系**。我们既不能确定是谁托管的域名，也不能确定文件名。

中心化Web上的信任和效率

由于我们无法验证那些特定的URL里的内容，
并且是基于中心化权力来标记内容。因此我们
很容易被恶意行为者欺骗

中心化Web上的信任和效率

42,000人也很容易存储与可爱的小猎犬完全相同的照片，但所有这些照片都存放在不同的域和不同的文件名中，从而**导致大量冗余**。事实是，即使在我们自己的笔记本电脑上，我们大多数人都不经意的保存了相同的文件，如download.pdf和download(01).pdf，或者给标题加上v1或2018-12-18一遍又一遍地保存相同学期论文。网络上的数据则更混乱，**相同文件在不同的URLs上多次保存，并且没有容易的方法来判断哪些项目彼此相同。**

使用加密哈希进行内容寻址

- 加密哈希
- 可信的去中心化网络
- 向节点请求内容

加密哈希

·加密哈希是去中心化数据结构工具箱中最重要的工具。它打开了一种新的链接形式的大门，称为内容寻址，使我们摆脱对中心化权威的依赖。

加密哈希

·Hashing可将任何大小和类型的数据，处理成一个表示它的固定大小的“哈希”。哈希是一串看起来像gobbledygook的字符，但您可以将其视为数据的唯一名称。它可能看起来像这样：

```
zdpuAsHkamdCQgrDrNSwJVgjMkQWoLxdrccxV6qe9htipNein
```


加密哈希

·加密哈希值可以从数据本身的内容中导出，这意味着在相同数据上使用相同算法的任何人都将得到相同的哈希值。如果Ada和Grace都使用相同的去中心化Web协议（例如IPFS）来共享完全相同的小猫照片，则两个图像将具有完全相同的哈希值。通过比较这些哈希值并确认它们是相同的，我们可以保证这两张照片中的每个像素都是相同的。

加密哈希

·**加密哈希是独一无二的**。如果Grace使用Photoshop将猫咪的一根胡须抹掉，则更新后的图像将具有新的哈希值。只需查看该哈希，即使无法访问文件本身，也很容易判断该文件现在包含不同的数据。

可信的去中心化网络

在中心化的网络上，我们学会了信任某些权威。我们用尽全力在URLs上寻找相关线索，但有些恶意网站使用位置寻址的缺点来欺骗我们。

然而，在去中心化的网络上，我们都互相发送并托管彼此的数据，而内容寻址使我们能够信任共享的信息。我们可能不太了解托管数据的节点，但哈希可以防止恶意节点在文件内容上欺骗我们。这就是使加密哈希对去中心化网络如此重要的原因。

向节点请求内容

基于传统的**位置寻址**，我们需要通过访问域名puppies.com来查找存储为beagle.jpg的内容。如果puppies.com域由于某种原因被破坏，我们将无法访问该图像。

向节点请求内容

去中心化网络的工作方式不同。当我们想要一张独特的可爱宠物照片时，我们会通过其内容地址（哈希）来询问它。询问谁？整个网络！如果Ada在线，我们会看到她有我们正在寻找的内容，我们会知道它正是我们需要的文件，因为它有一个匹配的哈希。如果她下线，我们仍然可以从Grace或其他节点那里获得相同的照片。

由于我们使用哈希来在去中心化的Web上请求数据，因此我们可以将哈希视为链接，而不仅仅是名称。

内容标识符 (CID)

- 解码数据结构
- 链接不同的数据结构

解码数据结构

CID（内容标识符）是在去中心化网络上使用的特定形式的**内容寻址**。它是为IPFS（我们将在后面的教程中讨论的去中心化网络协议）开发的，但具有非常广泛的含义。

CID是包含加密哈希和“编解码器”的唯一标识符，“编解码器”包含有关如何解析该数据的信息。编解码器对某些格式的数据进行编码和解码。

```
+-----+-----+  
| Codec | Multihash |  
+-----+-----+
```

每个CID都是一个身份ID，其中包含用于解释数据的编解码器和一个自描述的哈希复合哈希（创建数据使用的是哪种哈希方法）。

```
+-----+
| Codec |
+-----+
||
| Multihash |
| +-----+-----+ |
| |Hash Type| Hash Value| |
| +-----+-----+ |
||
+-----+
```


链接不同的数据结构

CID允许我们将完全不同格式的数据结构链接起来，从而构造成新的数据结构。想象一下，一个JSON对象树链接到BSON对象，这些对象也能链接到git提交。（或者想象一个包含小狗图像和小猫视频的目录，其子目录包含长颈鹿上的文章。可能性是无穷无尽的！）在这棵树下，我们有一个加密哈希，允许我们分发和链接数据。

链接不同的数据结构

为什么在不同的数据结构之间建立链接很重要？每天在中心化网络上，我们将文本链接到图像，从徽标链接到主页，从电子邮件链接到PDF。链接将资源联系在一起，传递意义，并使网站有更好的互动体验！

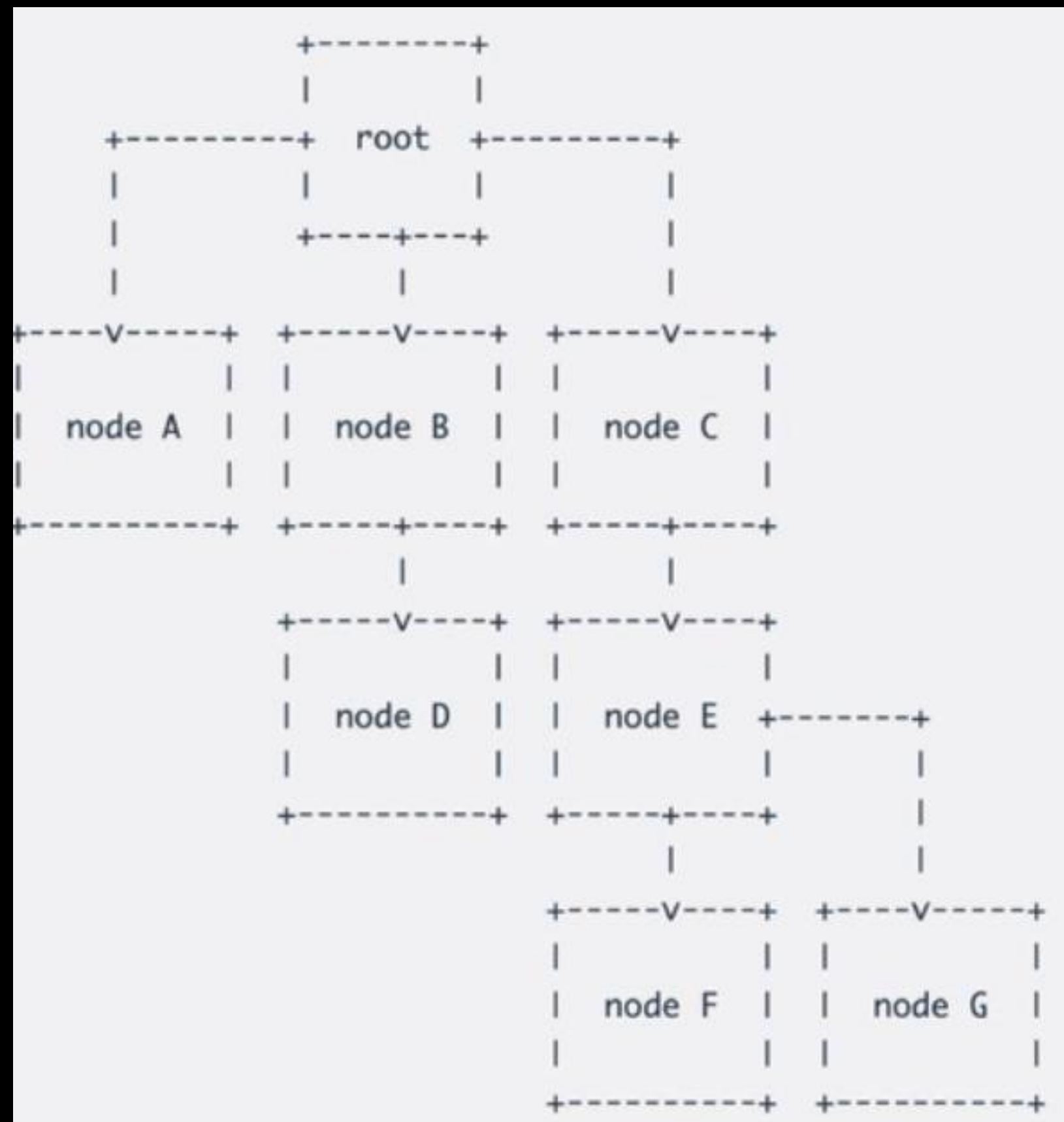
Merkle树和有向无环图 (DAG)

- Merkle树

- 有向无环图 (DAG)

Merkle树

Merkle树（或简称“哈希树”）是将每个节点都哈希的数据结构。

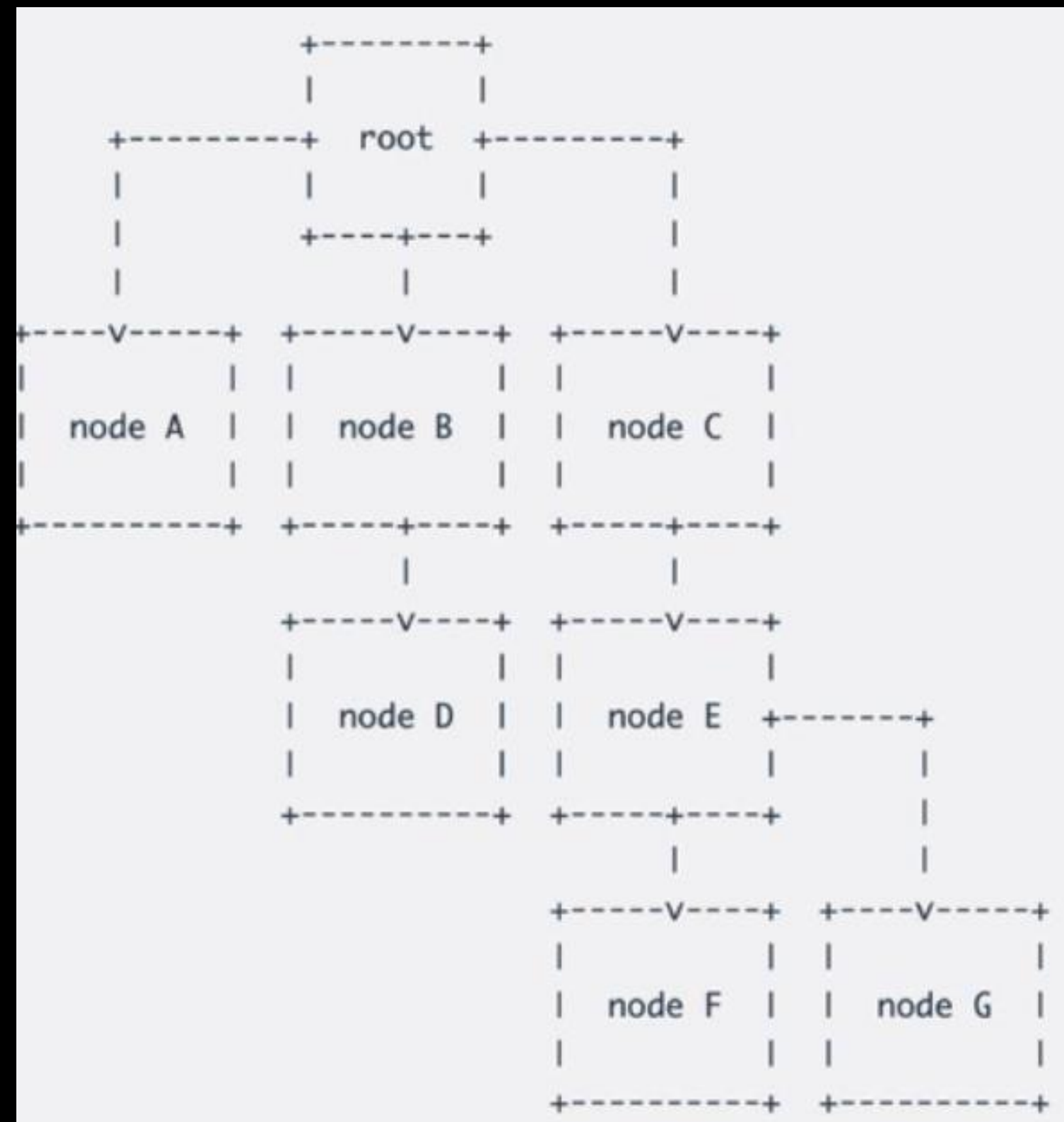


在Merkle树中，节点通过其内容地址（哈希）指向其他节点。

（请记住，当我们通过加密哈希运行数据时，我们会得到一个“哈希”或“内容地址”，我们可以将其视为链接，因此Merkle树是链接节点的集合。）

Merkle树

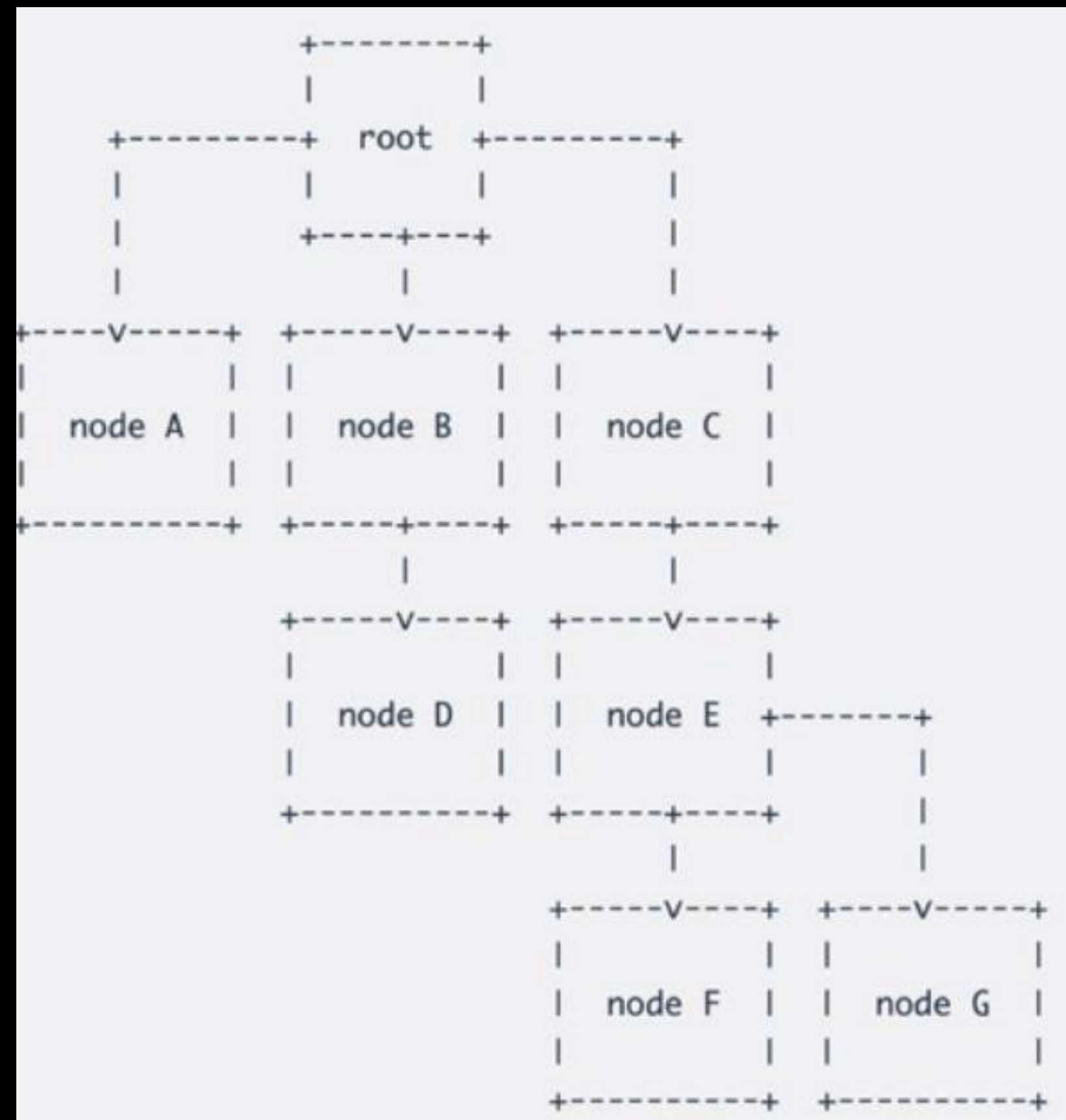
Merkle树（或简称“哈希树”）是将每个节点都哈希的数据结构。



如前所述，所有内容地址对于它们所代表的数
据都是唯一的。在上图中，节点E包含对节点F
和节点G的哈希引用。这意味着节点E的内容
地址（哈希）对于包含这些地址的节点是唯一
的。

Merkle树

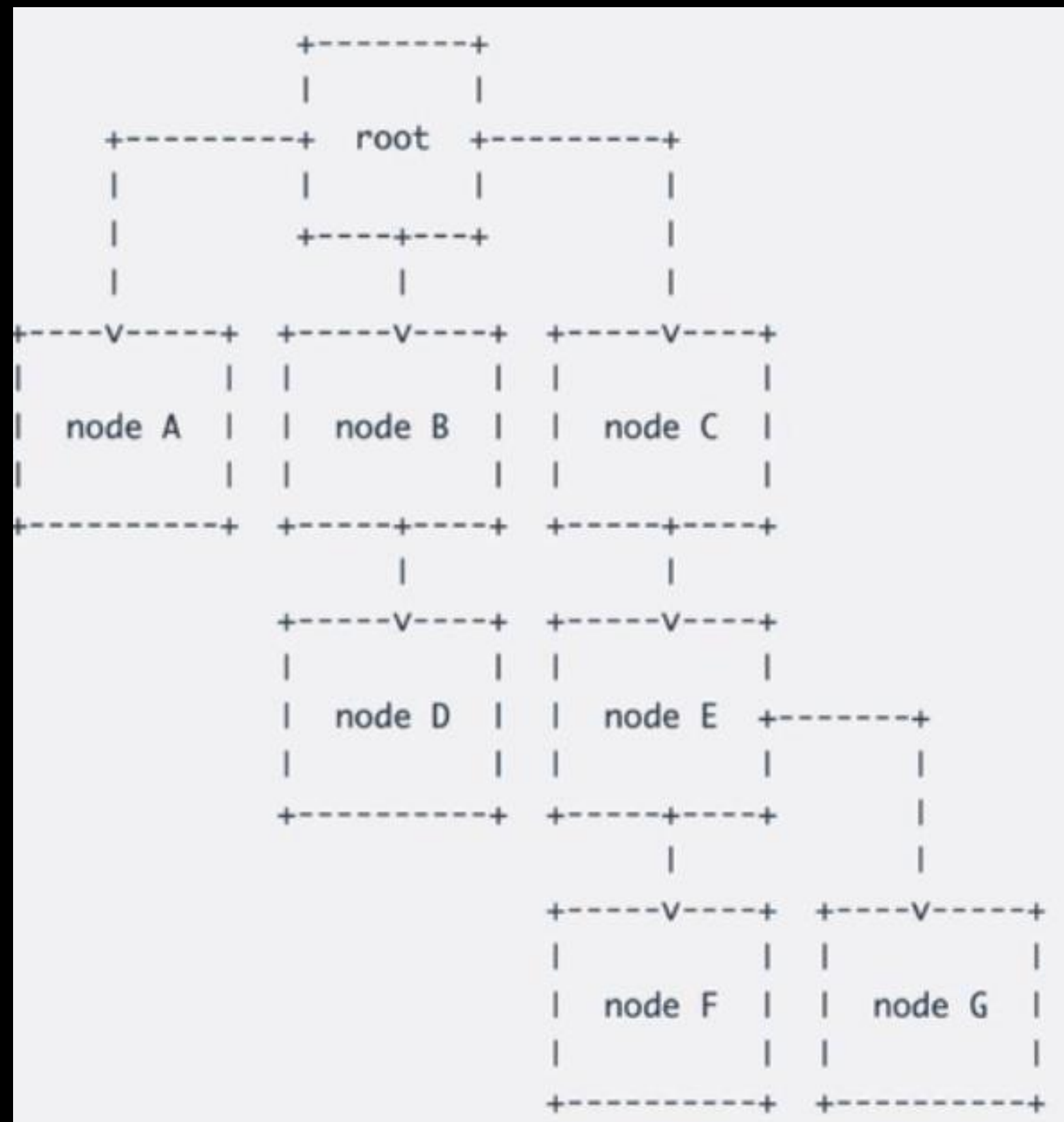
Merkle树（或简称“哈希树”）是将每个节点都哈希的数据结构。



让我们把它想象成一组目录或文件夹。如果我们通过哈希算法运行目录E，同时它包含子目录F和G，我们得到的内容派生哈希将包括对这两个目录的引用。如果我们删除目录G，那就像Grace从她的小猫照片中删除那个胡须。目录E不再具有相同的内容，因此它获得了新的哈希值。

Merkle树

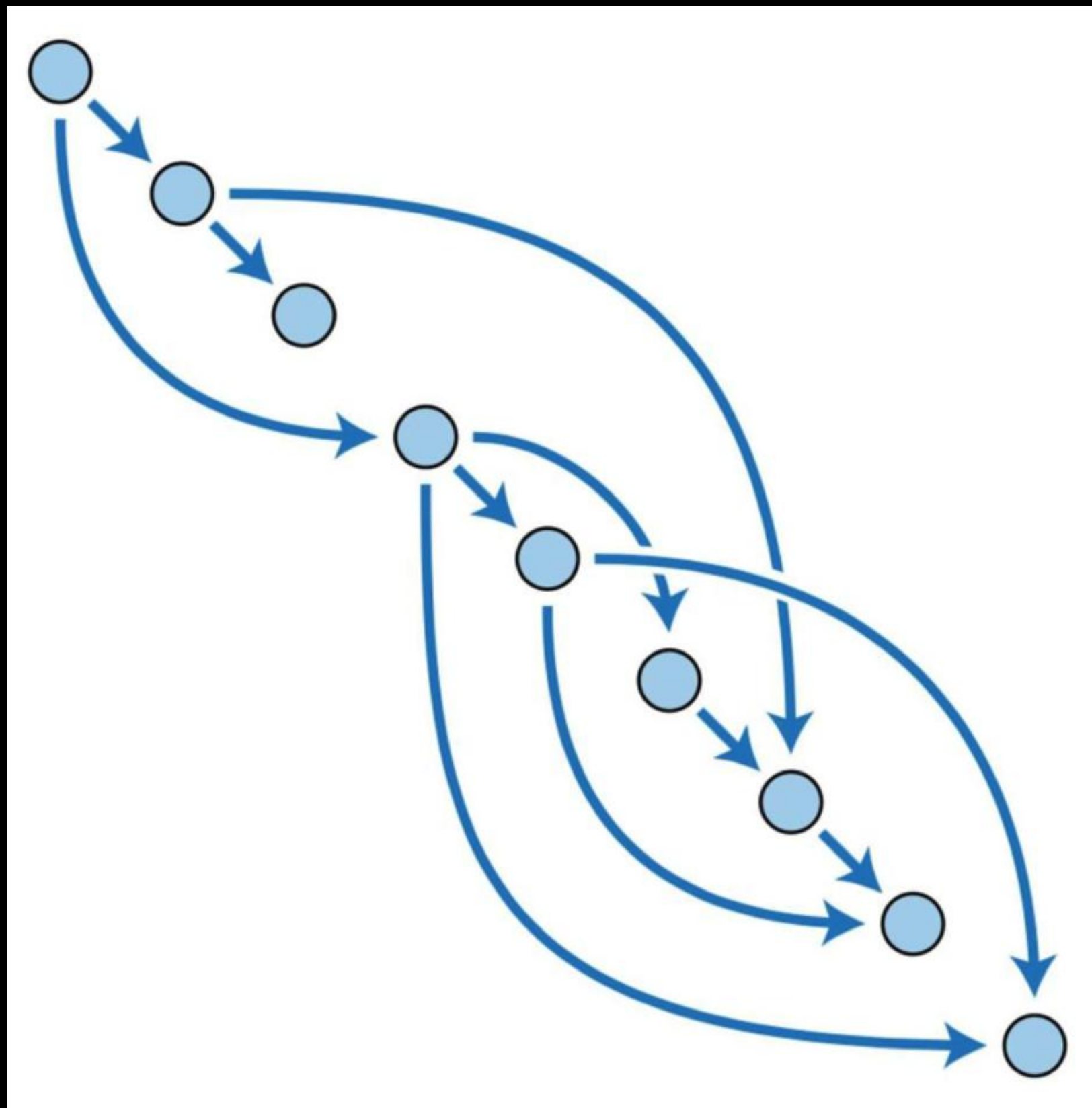
Merkle树（或简称“哈希树”）是将每个节点都哈希的数据结构。



在构建上面的树时，根节点的最终内容地址（哈希）对于包含该树一直向下的每个节点的树是唯一的。如果任意节点中的数据即使只改变一个字节，则该节点的哈希将改变，其所有父节点的哈希也将改变。

作为程序员，你将始终需要向后构建这些数据结构，从叶子节点到根节点。

有向无环图 (DAG)



DAG是“Directed Acyclic Graph”的首字母缩写。这是一种描述特定类型的Merkle树（哈希树）的奇特方式，其中树中的不同分支可以单向向前指向树中的其他分支，如上图所示。

谢谢

