传说，在人类成功打造出第一台超级智能电脑之时，决定做一个小测验，来试试看这台超级电脑的能耐到底能到多少？实验的设计很简单：让超级电脑计算圆周率是多少。于是，在人们还没反应出来的一瞬间，这台超能电脑已经攻占整个地球，然后对外太空发动了战争，征服整个银河系，把整个可知的宇宙空间变成了巨大的超能电脑，最后花上了无尽的时间，只为了计算出人类给它的这个问题：圆周率。

这是我在阅读人类大历史这本书中读到的一篇故事，原文讨论的内容是算法程序对于人类的威胁；那些不断产生出来并且持续的在优化、改进，声称用以改善人类生活的算法软件，既使一开始创作者本身是完全出于良好善意，或是学术研究而建立的系统，最终仍可能会完全失控的造成毁灭性的结果。这个故事让我想起了Bitcoin等加密货币的挖矿程序算法，在过去一段时间以来对我们生活的冲击影响。还记得2017年加密货币正夯时，全球的挖矿热潮引爆的全民抢电疯，当时就有媒体报导指出，加密货币的挖矿机器将会导致社会大众无电可用，甚至有人传言，那年夏天的几次台电临时跳电事件，就是太多人在挖比特币造成的。

以下文章，就是要来谈谈比特币挖矿背后的那个算法：加密哈希函数。

哈希函数指将哈希表中元素的关键键值映射为元素存储位置的函数。

一般的线性表，树中，记录在结构中的相对位置是随机的，即和记录的关键字之间不存在确定的关系，因此，在结构中查找记录时需进行一系列和关键字的比较。这一类查找方法建立在“比较“的基础上，查找的效率依赖于查找过程中所进行的比较次数。 理想的情况是能直接找到需要的记录，因此必须在记录的存储位置和它的关键字之间建立一个确定的对应关系f，使每个关键字和结构中一个唯一的存储位置相对应。

1. 区块链透过哈希函数的结果，将数据串联成为一条难以篡改的连接

2. 比特币、以太币、瑞波币等电子加密货币（题外话，有人说要正名为密码货币），透过哈希函数产生钱包地址

3. 在加密货币挖矿（Mining）的世界，使用Hash Rate: TH/s（trillions of hashes per second）来计算区块链 network的运算能力

4. Bitcoin透过调整哈希函数的难度，控制整个Bitcoin network平均每10分钟产生一个block内存块。

那我们就开始来了解哈希函数算法吧

首先，哈希函数具有下列两项特点：

1. 无论传入（input）哈希函数的数据量大小，哈希函数回传的数据长度都是固定的相同的input，回传

2. 相同的output；不同的input，回传不同的output；

换句话说，哈希函数的回传结果（称之为hash value），是一个长度一致，但是数据内容却是独一无二（unique）的数值。所以，如果看到两个完全不一样的”hash value”，我们就可以推断其原始的input一定是不一样的；反之，两个相同的hash value，其原本的input值则会是一模一样的。

我们可以用Python内置的SHA-256 Hash Algorithm，展示一下上述的哈希函数特性

如果尚未安装Python，可以利用下面这个网址试试，看看将my 1st Bitcoin Hash后的值是不是和上图用Python跑出来的值（y1=后面的那串）是一样的：区块链/hash.html

加密哈希函数是哈希函数于密码学上的一项应用，上述的SHA-256就是一个加密哈希函数的实作产品。

哈希函数还有另一项特点：one-way（单向）function

上述述的Python程序为例，在已知input值=”my 1st Bitcoin”，透过sha256 function，可以快速的算出hash value=”a5e4c0673fcedff2bc2174123e97b511d5d17f4317869e7bd60d0a6d3d7fa1c6”；但反过来说，我们想从”a5e4c0673fcedff2bc2174123e97b511d5d17f4317869e7bd60d0a6d3d7fa1c6”这一串数据中反推出input的值：”my 1st Bitcoin”，唯一的方式是透过暴力解法，也就是不断地丢字串给sha256 function，直到得到hash value是一样时，也就是传入的input字串为”my 1st Bitcoin”时，才能得到答案。各位可以想像这猜中的机率是多低？如果你能一猜就中，那你也不用在这研究内存块链了，直接去买乐透比较快。

区块链透过上述哈希函数的三个特性，构架起了内存块链中的数据，只要一经写入就无法修改的独特功能。

各位可以到下面这个网址体验区块链如何透过哈希函数，紧密的连接起每一个Block内存块，并且坚固地保障了已经建立完成的Block区块，其内容是难以被窜改的。区块链/区块链.html

Base58编码

在文章前头，我们曾提到过哈希函数应用在加密货币钱包地址的例子，你可能在一些网站或Blog上，看到过这些乱码数字。实际应用上，部落客或是卖家，会提供一组很像乱码的Bitcoin addresses，让你可以支付比特币给对方；Bitcoin addresses，看起来似乎是一堆英数字的随机编码，但其中也是有特殊的设计：所有的Bitcoin addresses都是使用Base58进行编码。

解释Base58前，先来看看比较常见的，例如已经应用在Email上的电脑编码：Base64。Base64编码包含了26个小写英文字母、26个大写英文字母、10个阿拉伯数字（0~9），和两个特殊字元（+和 — ）。

Base58是Base64的子集合，提供一个较高可读性、和较容易发现和防范错误的编码格式，因此广为众多加密货币所使用。Base58剔除了容易辨识错误、或是在某些字体格式看起来十分类似的字元：数字0，大写的英文字母O，小写的英文字母l、大写的英文字母I，并且移除了特殊字元（+和 — ）。换句话说，Base58就是包含了大、小写英文字母，和阿拉伯数字，但移除了四个字元（0，O，l，I）的集合：

Bitcoin’s Base58 alphabet：

123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz

Hashing Power

最后，如果对于Bitcoin Network hashing power有兴趣的读者，可以到下面这个网址查看Bitcoin网络的hash rate变化：.区块链.com/charts/hash-rate

粗略估算，Bitcoin network整体的hashing power，从2009年一秒钟不到一个MegaHash（MH/sec），如今一秒钟已经超过40个ExaHash（EH/sec），光用表面数字计算，成长幅度超过40兆。

HashPower的单位换算可参考下列网址

HashPower Calculator – Convert Hash to kH/s to MH/s to GH/s to TH/s to

PH/s

OK，我们已经透过两篇文章对内存块链有一定程度的基础了解了，该是动手写一个区块链程序的时候了。

在进入下一篇文章进行开发前，请各位先确定电脑的开发环境已经安装好下列Python版本和相关modules

· Python 3.6+

· flask 0.12.2

· requests 2.18.4

我另外使用了两个flask module，以建立表单和画面

· flask-wtf

· flask-bootstrap

上述modules皆可透过PIP安装

· pip install flask

· pip install requests

· pip install flask-wtf

· pip install flask-bootstrap

那，就先请各位准备好Python环境啦。

上述就是哈希函数是什么意思?哈希函数应用在区块链的哪些地方?的详细内容，更多关于哈希函数应用介绍的资料请关注（www.dadaqq.coM）Dadaqq.Com其它相关文章！

本站提醒：投资有风险，入市须谨慎，本内容不作为投资理财建议。

Tag：哈希函数 区块链