网络与信息安全期末实验——

区块链

### 本次实验目的

〉 1. 了解区块链是个什么东西。

〉 2. 搭建一个自己的区块链。

### 实验平台

ubuntu虚拟机

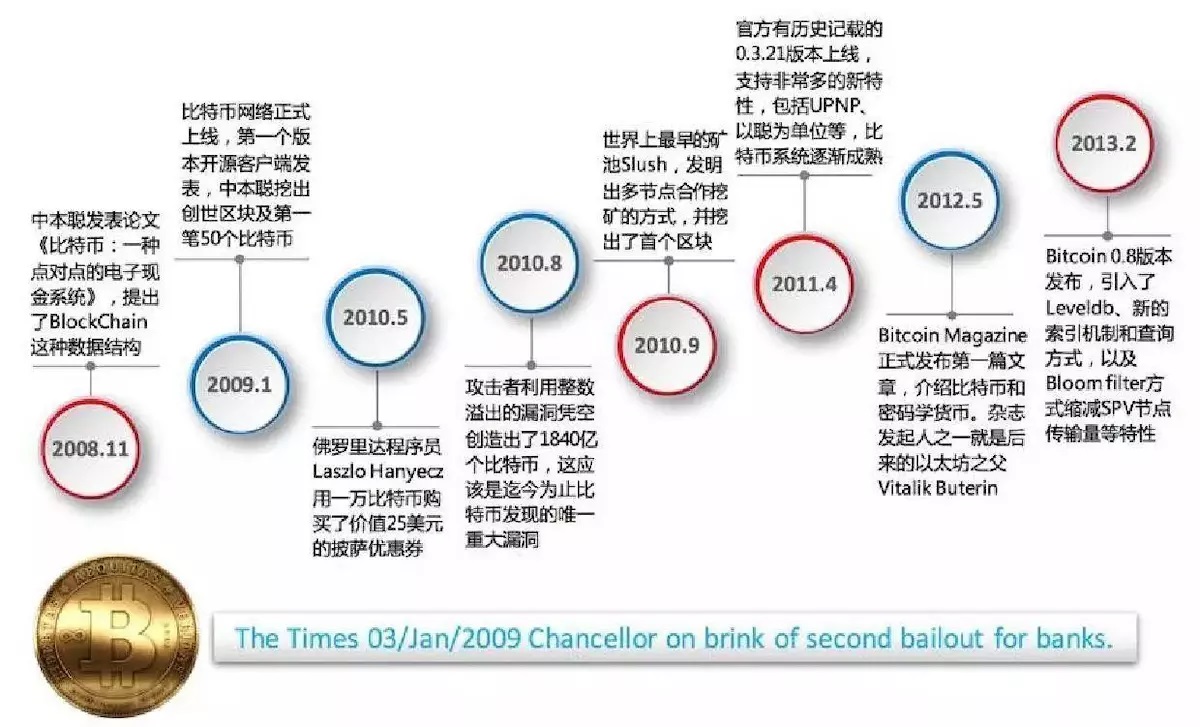
### 实验步骤

###### 1.什么是区块链？

我们或多或少都曾听说过比特币、区块链这类东西。那么，区块链到底是个什么东西？这里推荐一个讲解视频，非常生动形象的讲了这是个什么东西。

<https://www.bilibili.com/video/av12465079>

在知道这是个什么了之后，我们来了解一下区块链的**历史**。

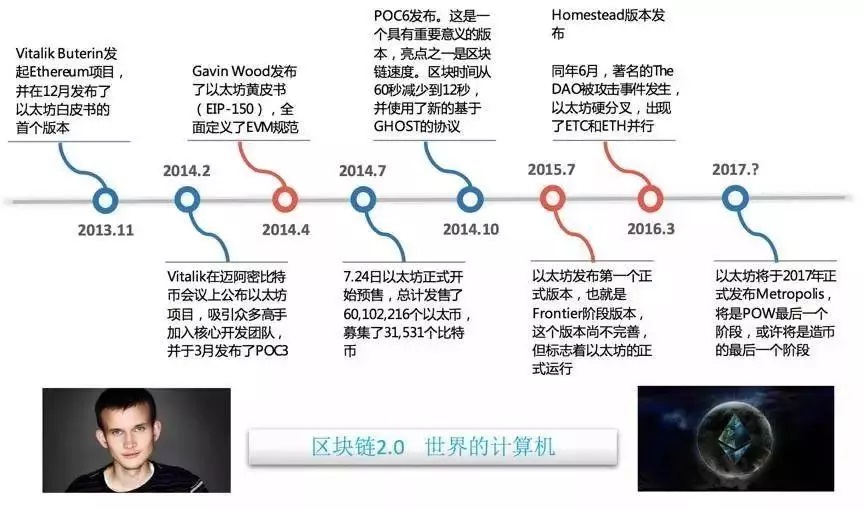


中本聪在2008年11月的时候发表了著名的论文《比特币：点对点的电子现金系统》，2009年1月紧接着用他第一版的软件挖掘出了创始区块，包含着这句：“The Times 03/Jan/2009 Chancellor on brink of second bailout forbanks.”，像魔咒一样开启了比特币的时代。

  2010年9月，第一个矿场Slush发明了多个节点合作挖矿的方式，成为比特币挖矿这个行业的开端。要知道，在此之前的2010年5月，1万比特币才值25美元，如果按照这个价格来计算，全部的比特币（2100万）也就值5万美元，集中投入挖矿显然是没有任何意义的。因此，建立矿池的决定就意味着有人认定比特币未来将成为某种可以与真实世界货币相兑换的，具有无限增长空间的虚拟货币，这无疑是一种远见。

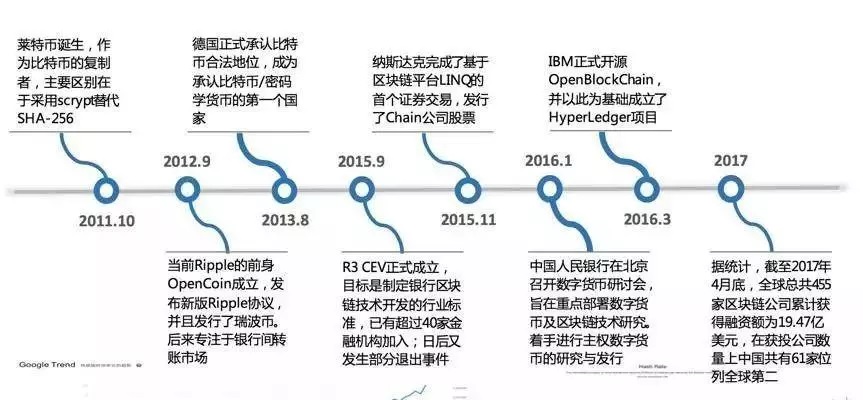
  2011年4月，比特币官方有正式记载的（https://bitcoin.org/en/version-history）第一个版本：0.3.21发布，这个版本非常初级，然而意义重大。首先，由于他支持uPNP，实现了了我们日常使用的P2P软件的能力，比特币才真正能登堂入室，进入寻常百姓家，让任何人都可以参与交易。其次，在此之前比特币节点最小单位只支持0.01比特币，相当于“分”，而这个版本真正支持了“聪”。可以说从这个版本之后，比特币才成为了现在的样子，真正形成了市场，在此之前基本上是技术人员的玩物。

  2013年，比特币发布了0.8的版本，这是比特币历史上最重要的版本，它整个完善了比特币节点本身的内部管理、网络通讯的优化。也就是在这个时间点以后，比特币才真正支持全网的大规模交易，成为中本聪设想的电子现金，真正产生了全球影响力。



事情总是没有那么一帆风顺，在最重要的0.8版本，比特币引入了一个大bug，所以这个版本发布以后比特币短时间就出现了硬分叉，导致整个比特币最后不得不回退到旧的版本，这个也导致了比特币价格产生大幅下跌。

  比特币后面的发展被越来越多的人所熟知，例如：世界各国对它的态度、算力的增长——2016年1月达到1EH/S，以及在Github上超过了1万个相关的开源项目，都证明比特币生态环境已经完全成熟了。



 比特币逐渐成熟之后，密码学货币的概念逐渐被人们所认知和接受。区块链也作为一个技术领域登堂入室。从2011年开始的几年内，莱特币、Ripple、R3等数字货币和区块链技术竞相出现。同一时期，德国正式承认比特币，纳斯达克通过自身的区块链平台完成交易，中国人民银行虽然它否定了比特币的地位，但是它却是全球唯一的一个立刻宣布要做自己的密码学货币/数字货币的银行。据统计，到今年4月份全球已经有455家区块链公司获得了将近20亿美元的投资，其中中国可统计的已经有61家。总体上看，在比特币、以太坊这样一些巨头的带动下，全球已经开始了一轮数字货币和区块链的热潮。

更多的详细资料可参考：<http://c.biancheng.net/blockchain/>

###### 2.创造一个自己的区块链

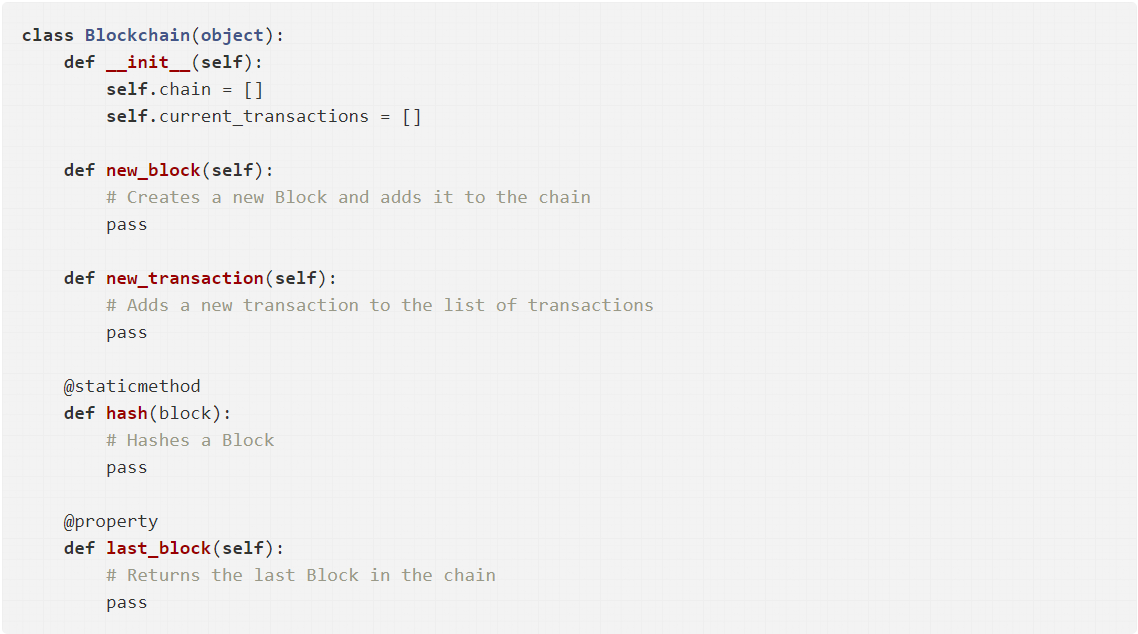
目前搭建区块链的方法有很多，包括用c++、java、python、ruby等等均有开源实现。这次，我们参考<https://segmentfault.com/a/1190000014483104>这篇文档，尝试搭建一个区块链。

先观察一个区块（block）的格式：

一个block中有5个变量：index、timestamp（时间）、transactions（交易）、proof（工作量证明）、previous\_hash。



一个类区块链的实例如下：



按照链接中的说明进行填充之后，再加入工作量证明机制。

工作量证明系统主要特征是客户端需要做一定难度的工作得出一个结果，验证方却很容易通过结果来检查出客户端是不是做了相应的工作。这种方案的一个核心特征是不对称性：工作对于请求方是适中的，对于验证方则是易于验证的。它与验证码不同，验证码的设计出发点是易于被人类解决而不易被计算机解决。

\*\*\*这里查看实验报告要求2.

Blockchain 作为 API 接口

我们将使用 Python Flask 框架，这是一个轻量 Web 应用框架，它方便将网络请求映射到 Python 函数，现在我们来让 Blockchain 运行在基于 Flask web 上。

我们将创建三个接口：

/transactions/new 创建一个交易并添加到区块

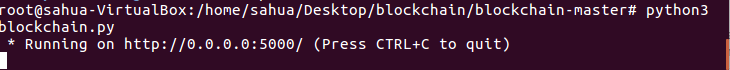
/mine 告诉服务器去挖掘新的区块

/chain 返回整个区块链

一般而言，这一步的目的是让所有人可以访问这个区块链，并不需要将new和mine公开。不过现在为了操作方便，就这样了吧。

运行区块链

运行python3 blockchain.py



错误提示：print(f’’)错误，则需要升级python3到3.6版本以上

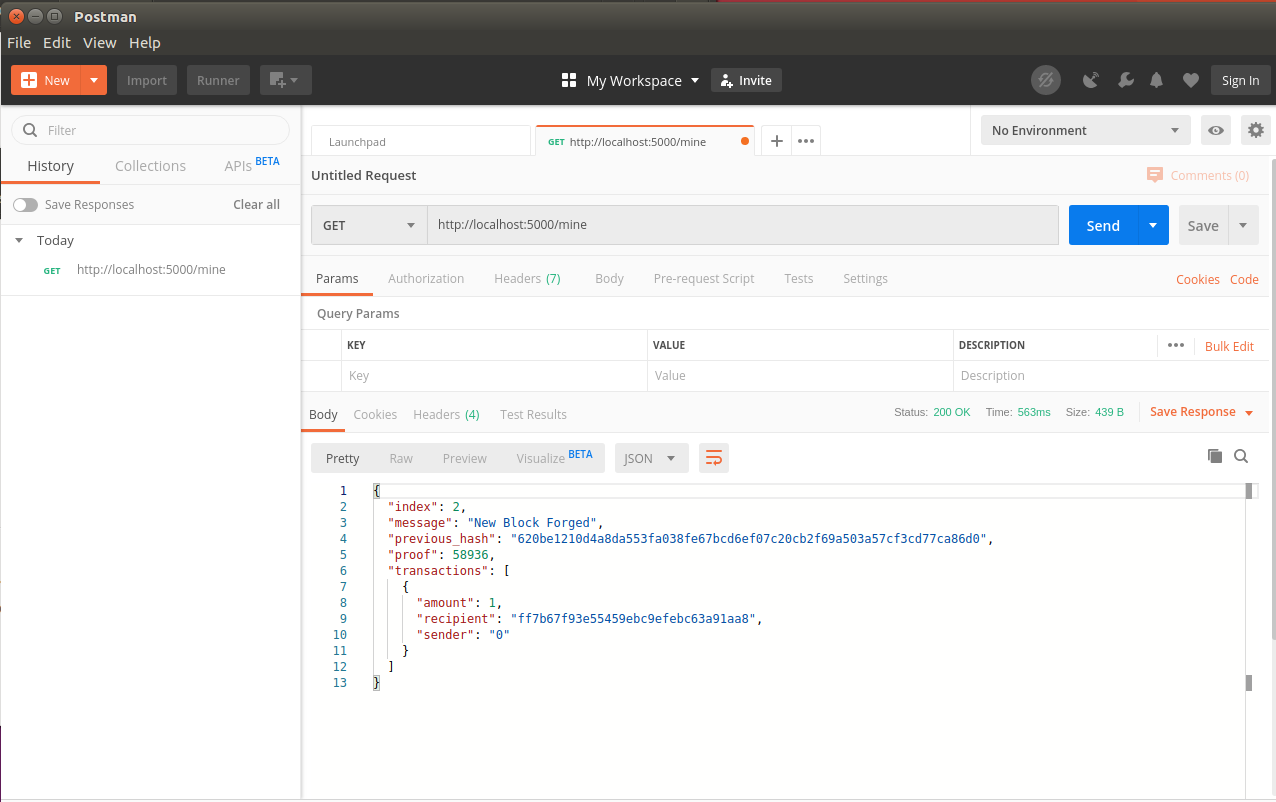
错误提示：import错误，安装所需的库。

现在我们用postman来进行操作。

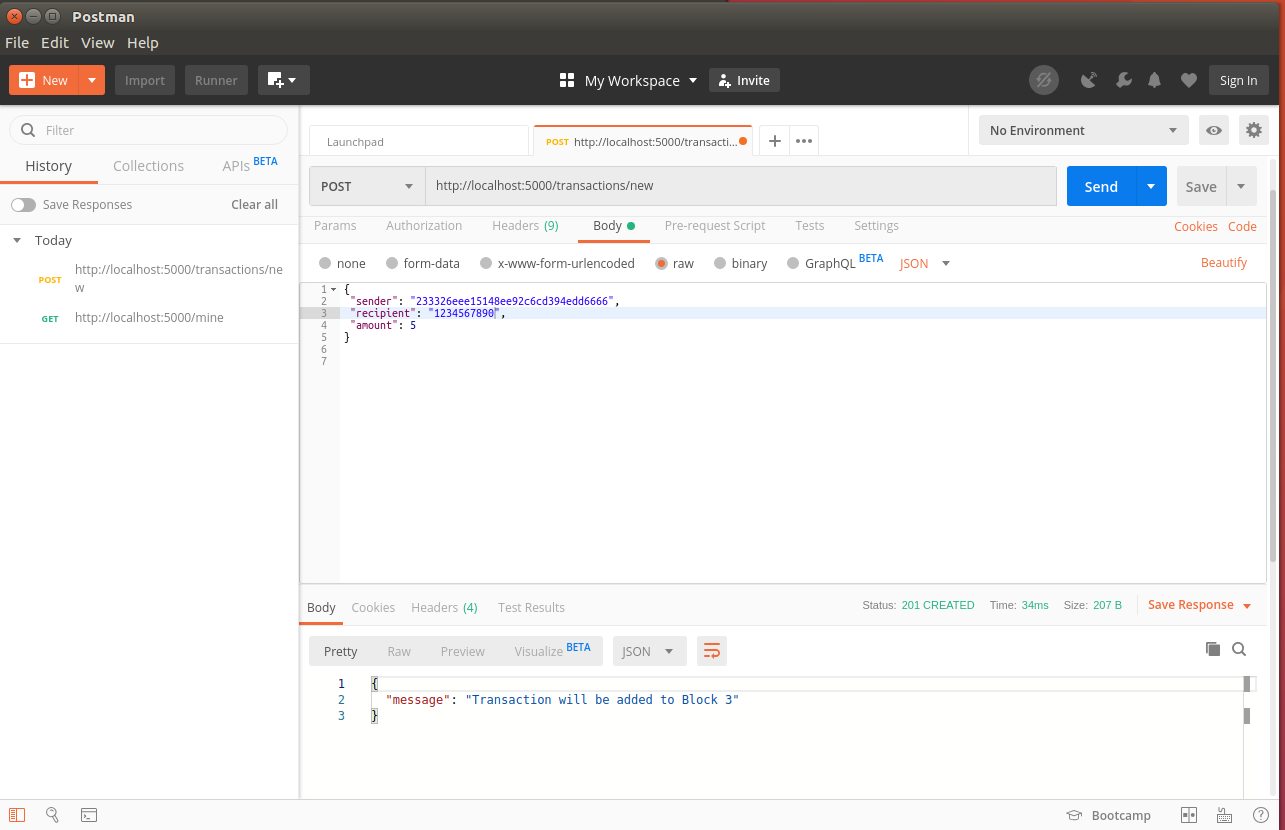
附：postman Ubuntu 安装方法<https://blog.csdn.net/xiamoyanyulrq/article/details/80334900>

附2：不需要注册，弹出界面直接叉掉。

再postman中输入新建一个请求，并访问http://localhost:5000/mine，如图所示。



同理，post请求也能正常运行。



节点一致性（共识）

我们已经有了一个基本的区块链可以接受交易和挖矿。但是区块链系统应该是分布式的。既然是分布式的，那么我们究竟拿什么保证所有节点有同样的链呢？这就是一致性问题，我们要想在网络上有多个节点，就必须实现一个一致性的算法。

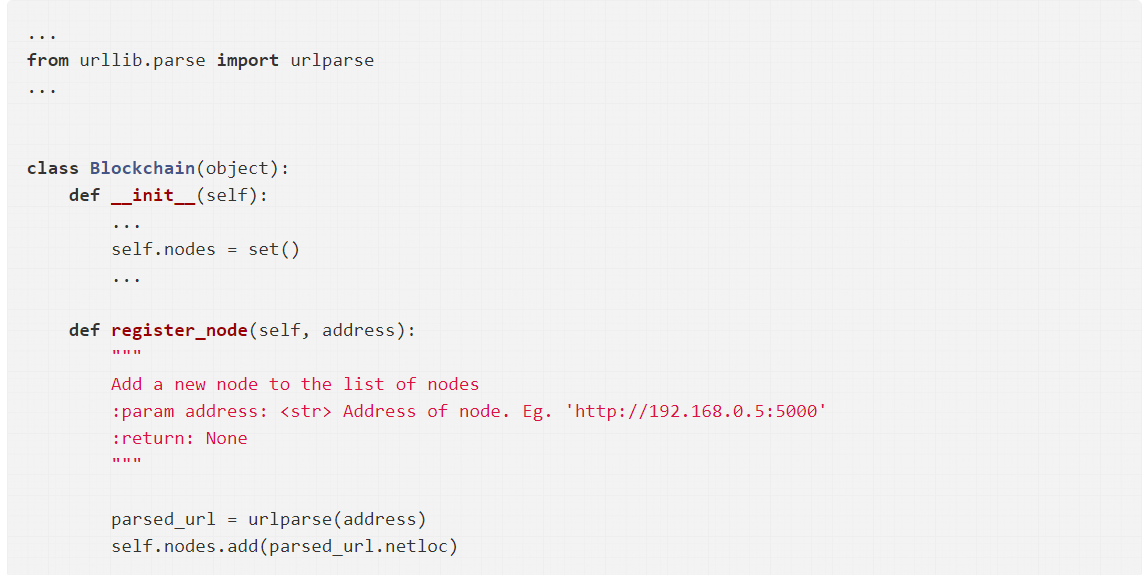
当一个节点与另一个节点有不同的链时，就会产生冲突。 为了解决这个问题，我们将制定最长的有效链条是最权威的规则。换句话说就是：在这个网络里最长的链就是最权威的。 我们将使用这个算法，在网络中的节点之间达成共识。

注册节点

在实现一致性算法之前，我们需要找到一种方式让一个节点知道它相邻的节点。每个节点都需要保存一份包含网络中其它节点的记录。因此让我们新增几个接口：

/nodes/register 接收 URL 形式的新节点列表.

/nodes/resolve 执行一致性算法，解决任何冲突，确保节点拥有正确的链.



此处需要对函数进行填写，并保存测试截图。

如果实验中出现问题，可以下载<https://github.com/dvf/blockchain>文档进行对照。

### 实验报告要求

1. 写出一个block中index、timestamp（时间）、transactions（交易）、proof（工作量证明）、previous\_hash各自的用处。
2. 这篇文章中的proof\_of\_work写的有一点问题，请说明它有什么问题，可能导致什么结构，并进行修正，附在实验报告中。

def proof\_of\_work(self, last\_proof):

"""

Simple Proof of Work Algorithm:

- Find a number p' such that hash(pp') contains leading 4 zeroes, where p is the previous p'

- p is the previous proof, and p' is the new proof

:param last\_proof: <int>

:return: <int>

"""

proof = 0

while self.valid\_proof(last\_proof, proof) is False:

proof += 1

return proof

@staticmethod

def valid\_proof(last\_proof, proof):

"""

Validates the Proof: Does hash(last\_proof, proof) contain 4 leading zeroes?

:param last\_proof: <int> Previous Proof

:param proof: <int> Current Proof

:return: <bool> True if correct, False if not.

"""

guess = f'{last\_proof}{proof}'.encode()

guess\_hash = hashlib.sha256(guess).hexdigest()

return guess\_hash[:4] == "0000"

3. 本次实验为模拟实验，请指出一点未提到的与实际情况不同的地方

4. 写出你对区块链的或者理解感受（100字）