Bitcoin基础：数据结构

黄华威
副教授

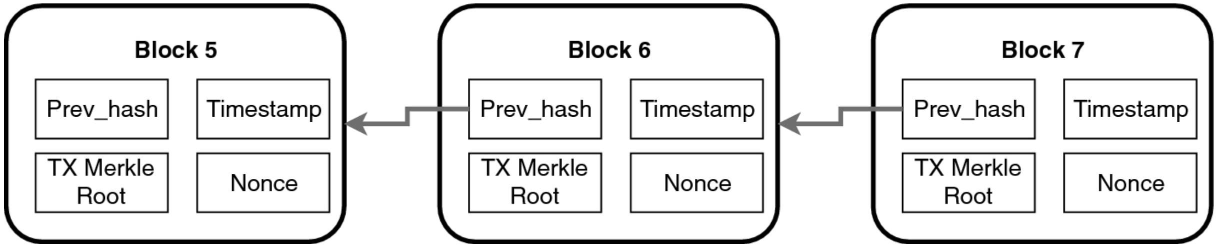
中山大学
数据科学与计算机学院
Http://xintelligence.pro

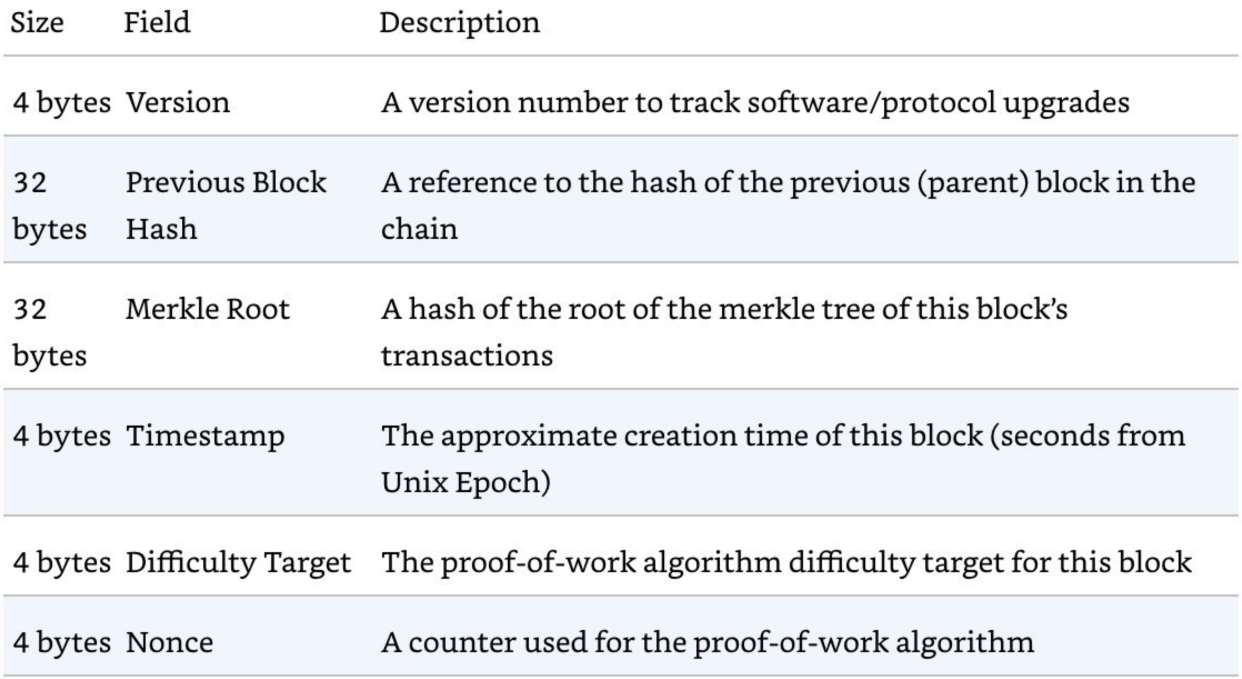
Outline & Keywords of this Class

- Part 1: Hash Pointer
- Part 2: Merkel Tree
- Part 3: A Bitcoin Block

Intro: What are there inside a block?

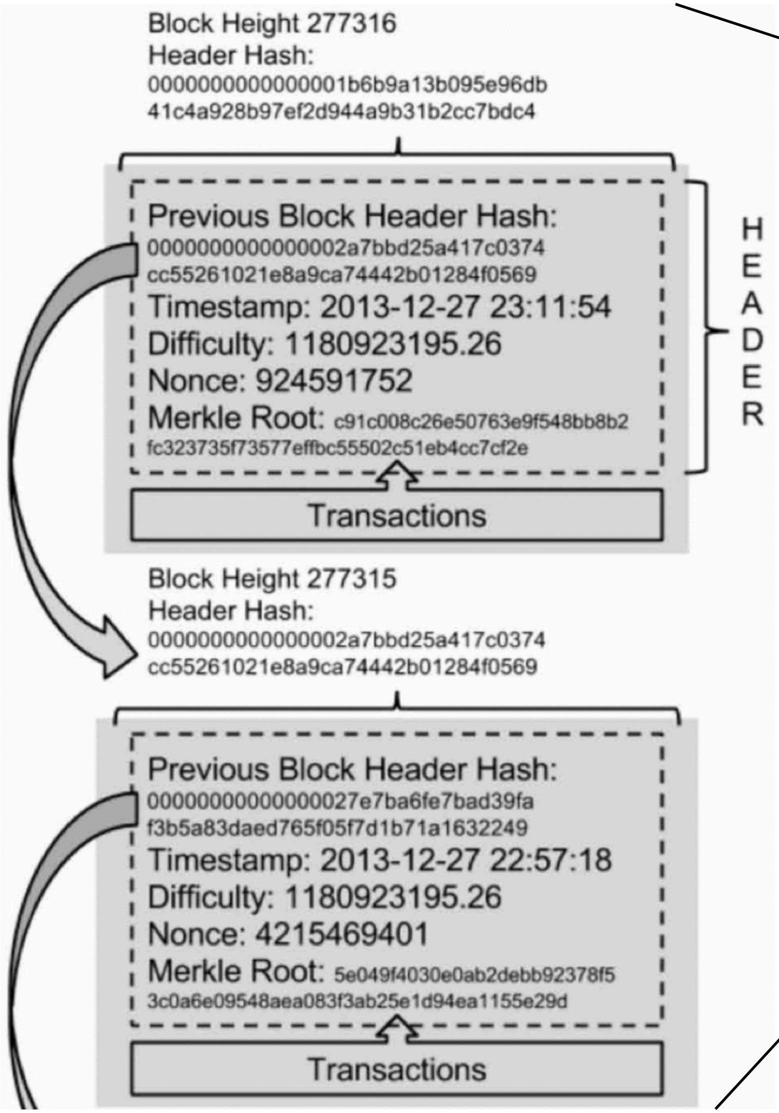
- Data Structure of a Bitcoin Block





Size	Field	Description
4 bytes	Version	A version number to track software/protocol upgrades
32 bytes	Previous Block Hash	A reference to the hash of the previous (parent) block in the chain
32 bytes	Merkle Root	A hash of the root of the merkle tree of this block's transactions
4 bytes	Timestamp	The approximate creation time of this block (seconds from Unix Epoch)
4 bytes	Difficulty Target	The proof-of-work algorithm difficulty target for this block
4 bytes	Nonce	A counter used for the proof-of-work algorithm

Intro: Blockchain structure (Antonopoulos, 2014)



Block Height 277316
Header Hash:
0000000000000001b6b9a13b095e96db
41c4a928b97ef2d944a9b31b2cc7bdc4

Previous Block Header Hash:
0000000000000002a7bbd25a417c0374
cc55261021e8a9ca74442b01284f0569
Timestamp: 2013-12-27 23:11:54
Difficulty: 1180923195.26
Nonce: 924591752
Merkle Root: c91c008c26e50763e9f548bb8b2
fc323735f73577effbc55502c51eb4cc7cf2e

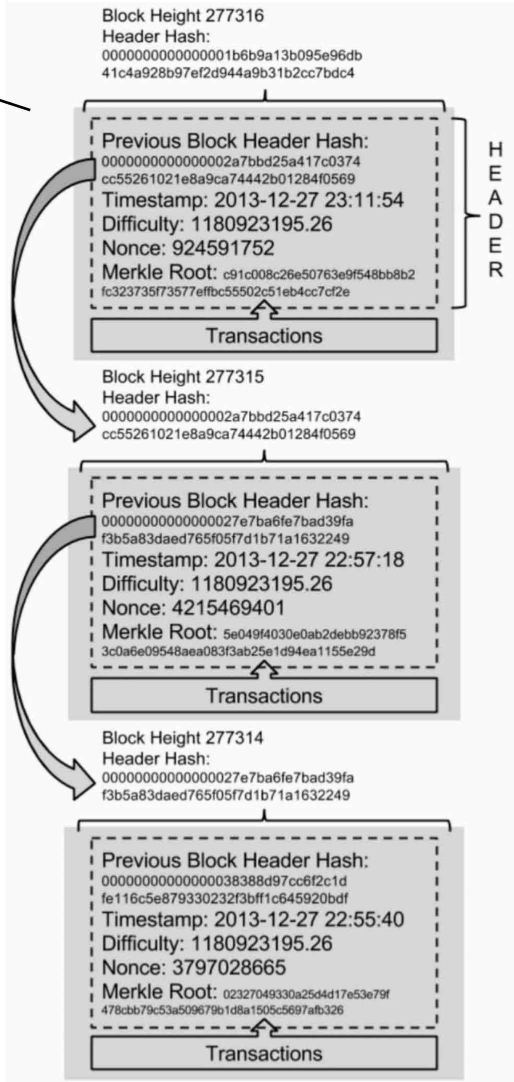
HEADER

Transactions

Block Height 277315
Header Hash:
0000000000000002a7bbd25a417c0374
cc55261021e8a9ca74442b01284f0569

Previous Block Header Hash:
00000000000000027e7ba6fe7bad39fa
f3b5a83daed765f05f7d1b71a1632249
Timestamp: 2013-12-27 22:57:18
Difficulty: 1180923195.26
Nonce: 4215469401
Merkle Root: 5e049f4030e0ab2debb92378f5
3c0a6e09548aea083f3ab25e1d94ea1155e29d

Transactions



Block Height 277316
Header Hash:
0000000000000001b6b9a13b095e96db
41c4a928b97ef2d944a9b31b2cc7bdc4

Previous Block Header Hash:
0000000000000002a7bbd25a417c0374
cc55261021e8a9ca74442b01284f0569
Timestamp: 2013-12-27 23:11:54
Difficulty: 1180923195.26
Nonce: 924591752
Merkle Root: c91c008c26e50763e9f548bb8b2
fc323735f73577effbc55502c51eb4cc7cf2e

HEADER

Transactions

Block Height 277315
Header Hash:
0000000000000002a7bbd25a417c0374
cc55261021e8a9ca74442b01284f0569

Previous Block Header Hash:
00000000000000027e7ba6fe7bad39fa
f3b5a83daed765f05f7d1b71a1632249
Timestamp: 2013-12-27 22:57:18
Difficulty: 1180923195.26
Nonce: 4215469401
Merkle Root: 5e049f4030e0ab2debb92378f5
3c0a6e09548aea083f3ab25e1d94ea1155e29d

Transactions

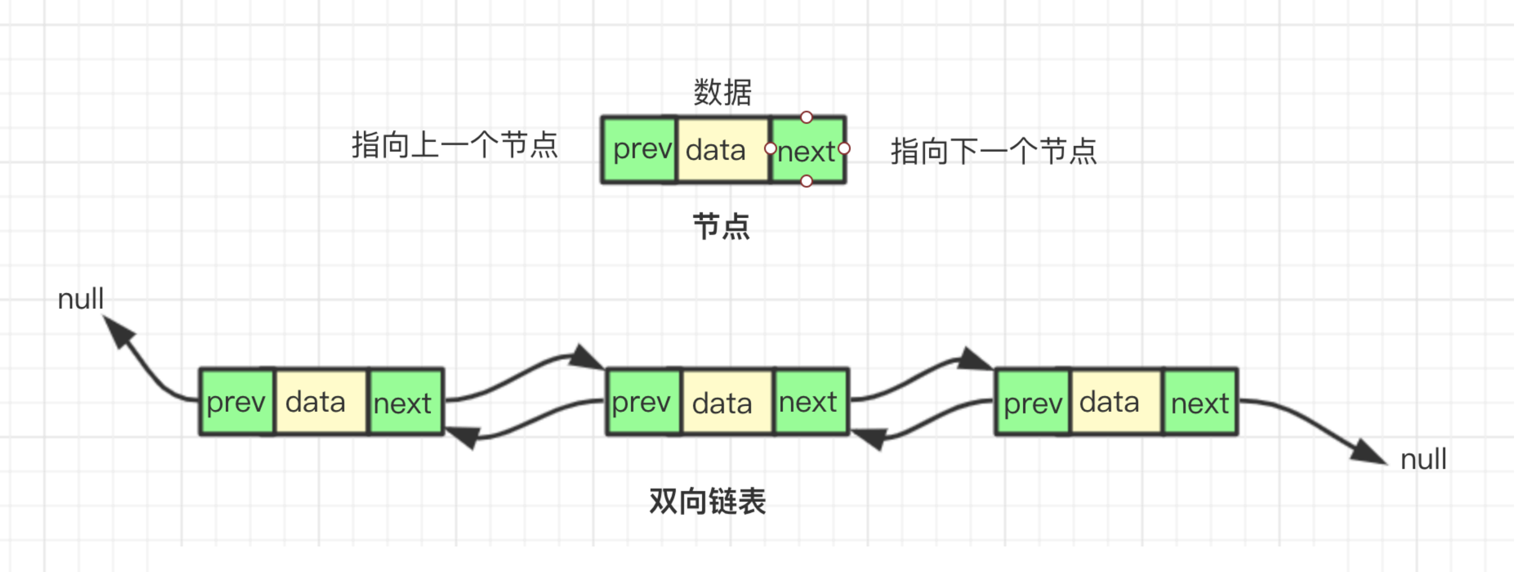
Block Height 277314
Header Hash:
00000000000000027e7ba6fe7bad39fa
f3b5a83daed765f05f7d1b71a1632249

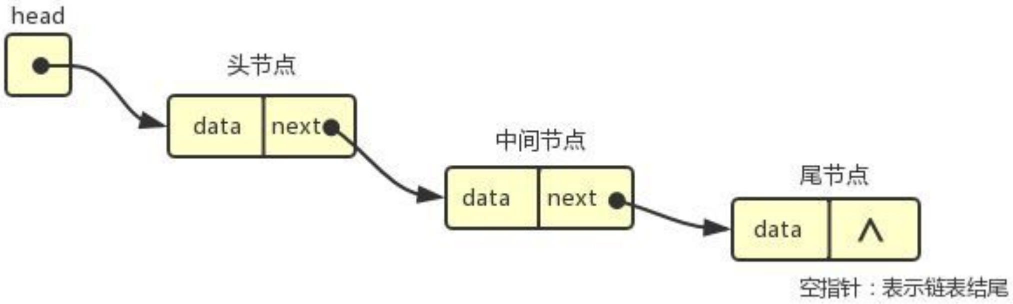
Previous Block Header Hash:
00000000000000038388d97cc6f2c1d
fe116c5e879330232f3bff1c645920bdf
Timestamp: 2013-12-27 22:55:40
Difficulty: 1180923195.26
Nonce: 3797028665
Merkle Root: 02327049330a25d4d17e53e79f
478cbb79c53a509679b1d8a1505c5697afb326

Transactions

Revisit: Data Link Tables

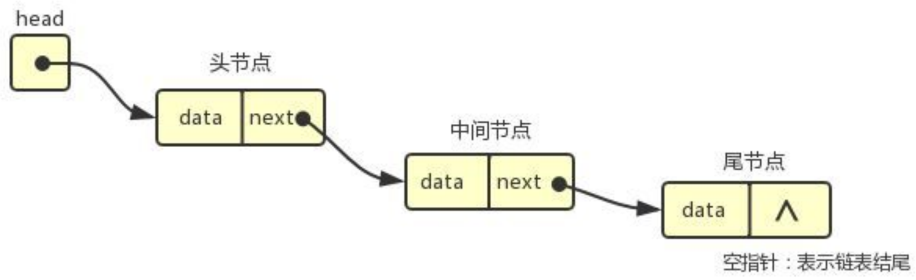
- 链表



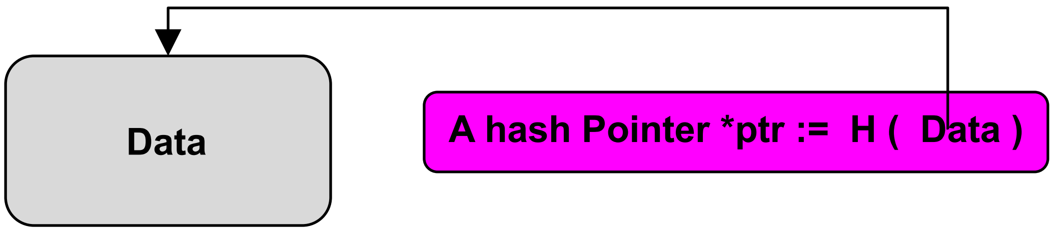


Part 1: Hash Pointer

- A normal pointer: *ptr = & data
  - Tells you the position where a data is

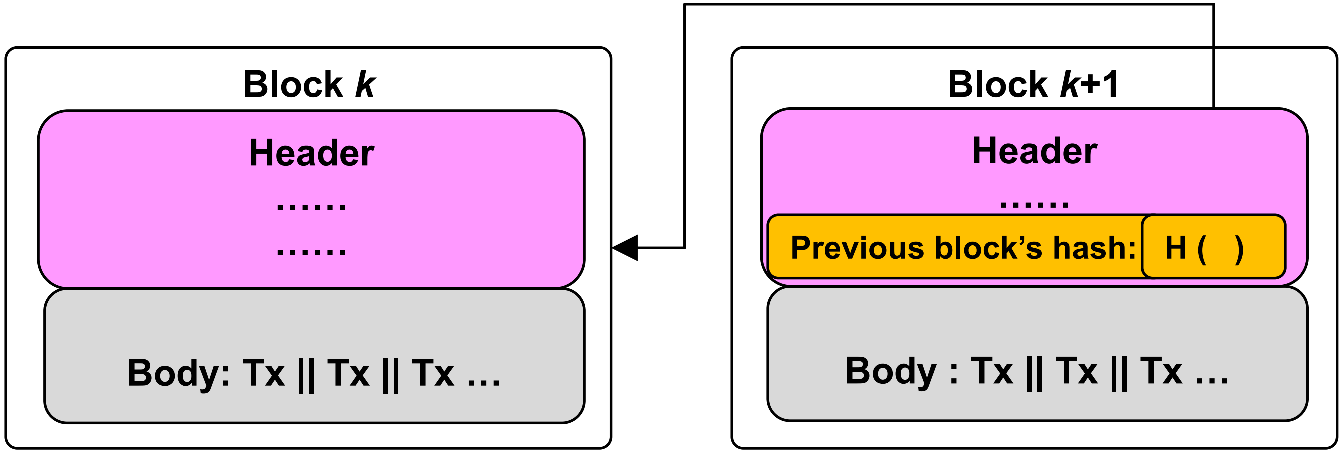


- A Hash Pointer: *ptr = H(Data)
  - Not only tells you where a data is
  - But also enables you to verify whether such data has been tampered or not



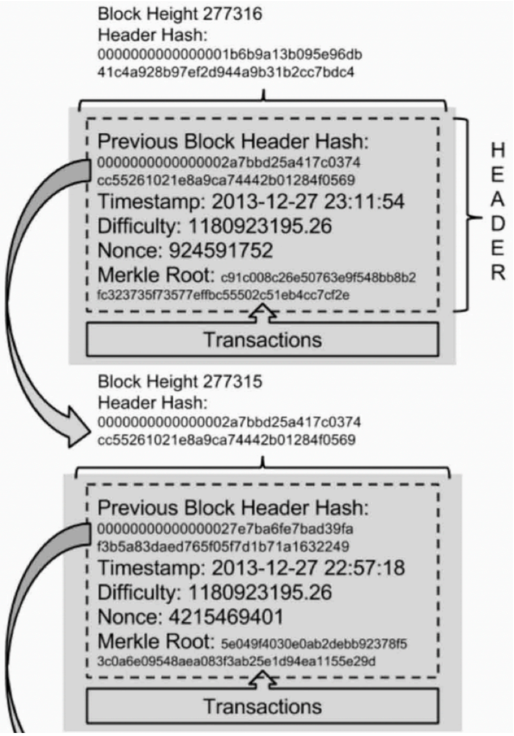
A blockchain == blocks + chains

- How to compute a hash pointer in bitcoin?
  - hash pointer := H(header || body) ?
  - hash pointer := H(header) ?



The previous block *k*

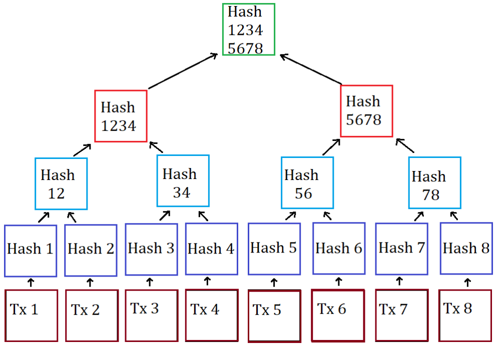
The current block *k*+1


Block Height 277316
Header Hash:
0000000000000001b6b9a13b095e96db
41c4a928b97ef2d944a9b31b2cc7bdc4
Previous Block Header Hash:
0000000000000002a7bbd25a417c0374
cc55261021e8a9ca74442b01284f0569
Timestamp: 2013-12-27 23:11:54
Difficulty: 1180923195.26
Nonce: 924591752
Merkle Root: c91c008c26e50763e9f548bb8b2
fc323735f73577effbc55502c51eb4cc7cf2e
Transactions
HEADER
Block Height 277315
Header Hash:
0000000000000002a7bbd25a417c0374
cc55261021e8a9ca74442b01284f0569
Previous Block Header Hash:
0000000000000027e7ba6fe7bad39fa
f3b5a83daed765f05f7d1b71a1632249
Timestamp: 2013-12-27 22:57:18
Difficulty: 1180923195.26
Nonce: 4215469401
Merkle Root: 5e049f4030e0ab2debb92378f5
3c0a6e09548aea083f3ab25e1d94ea1155e29d
Transactions

Part 2: Merkle Tree

- NOT Angela Merkel + tree

Ralph Merkle



Hash 1234 5678

Hash 1234

Hash 5678

Hash 12

Hash 34

Hash 56

Hash 78

Hash 1

Hash 2

Hash 3

Hash 4

Hash 5

Hash 6

Hash 7

Hash 8

Tx 1

Tx 2

Tx 3

Tx 4

Tx 5

Tx 6

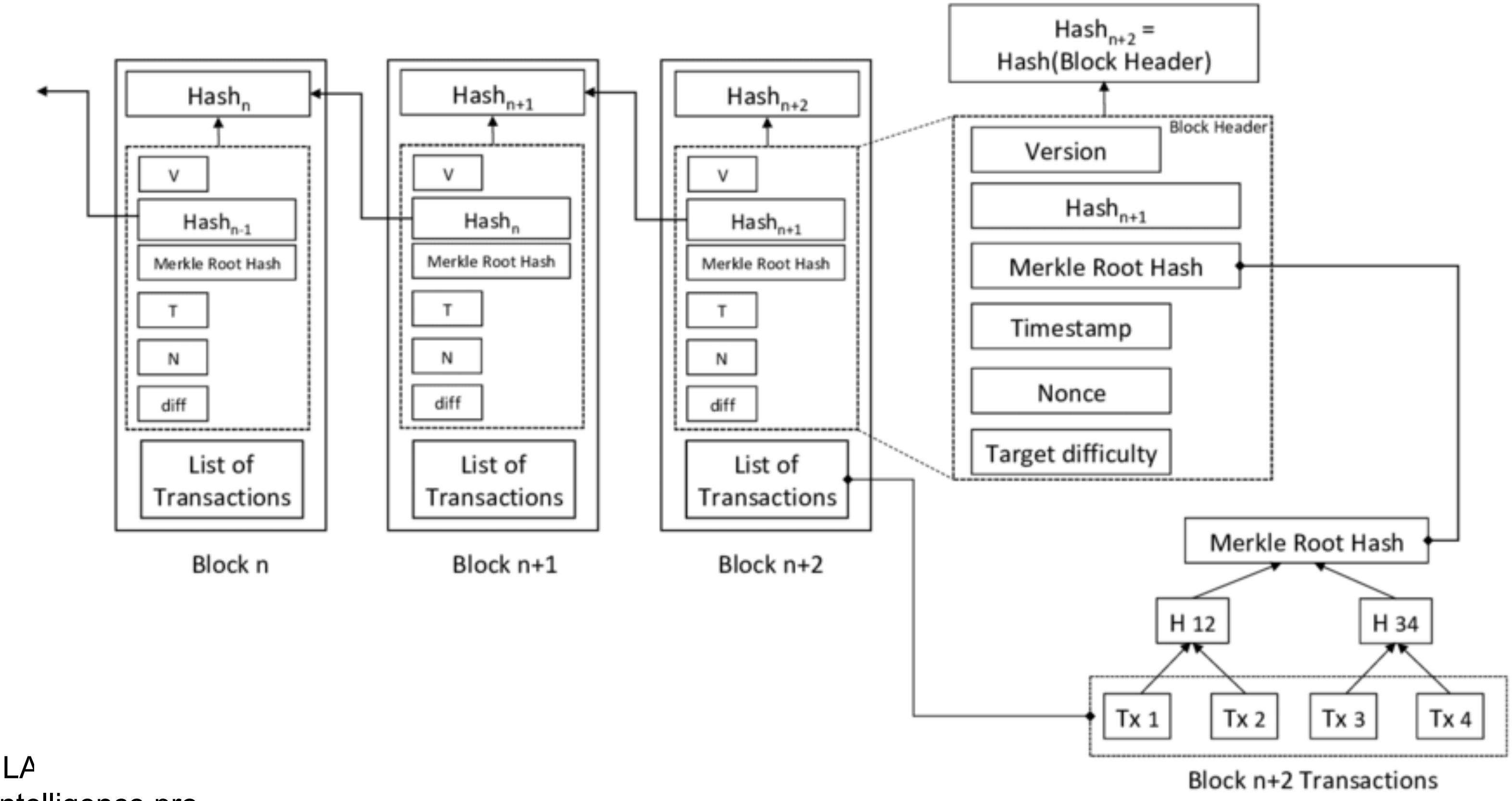
Tx 7

Tx 8

Proposed in 1979, by Ralph Merkle

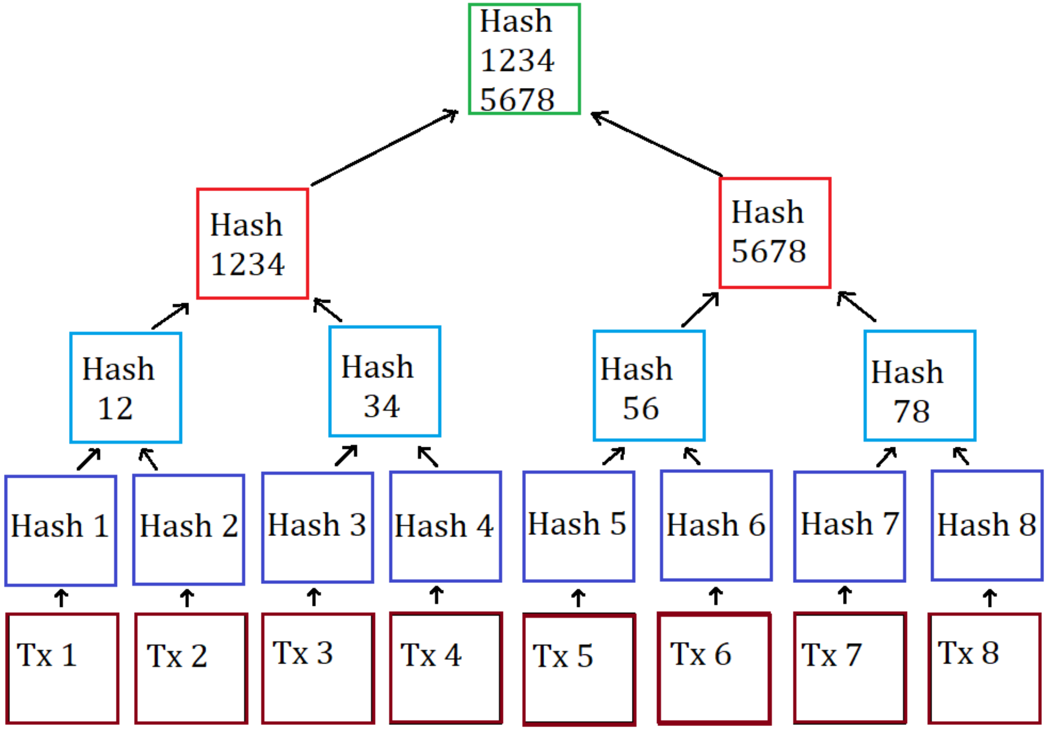
Where is a Merkle Tree in Bitcoin?

- 使用哈希指针的二叉树 —— Merkle Tree
- The Merkle tree is a way of structuring large amounts of data in the form of hashes, and representing that data with a single hash.



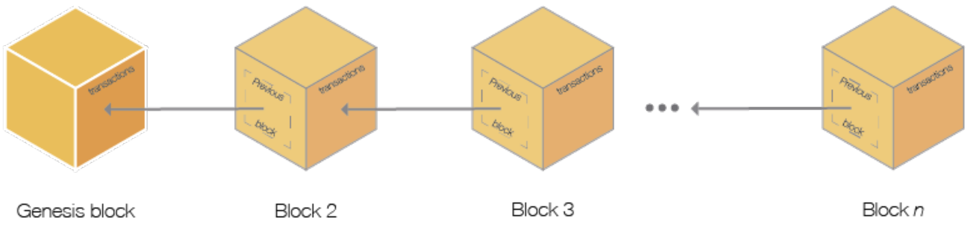
What can Merkel Tree do in Bitcoin?

- Merkle root hash (32 Bytes)
  - The hash of the Merkle Tree root of all transactions in the block.
  - If any transaction is changed, removed, or reordered, it will change the merkle root hash.
  - This is what locks all of the transactions in the block.



A Question: 为何区块链可以防篡改？

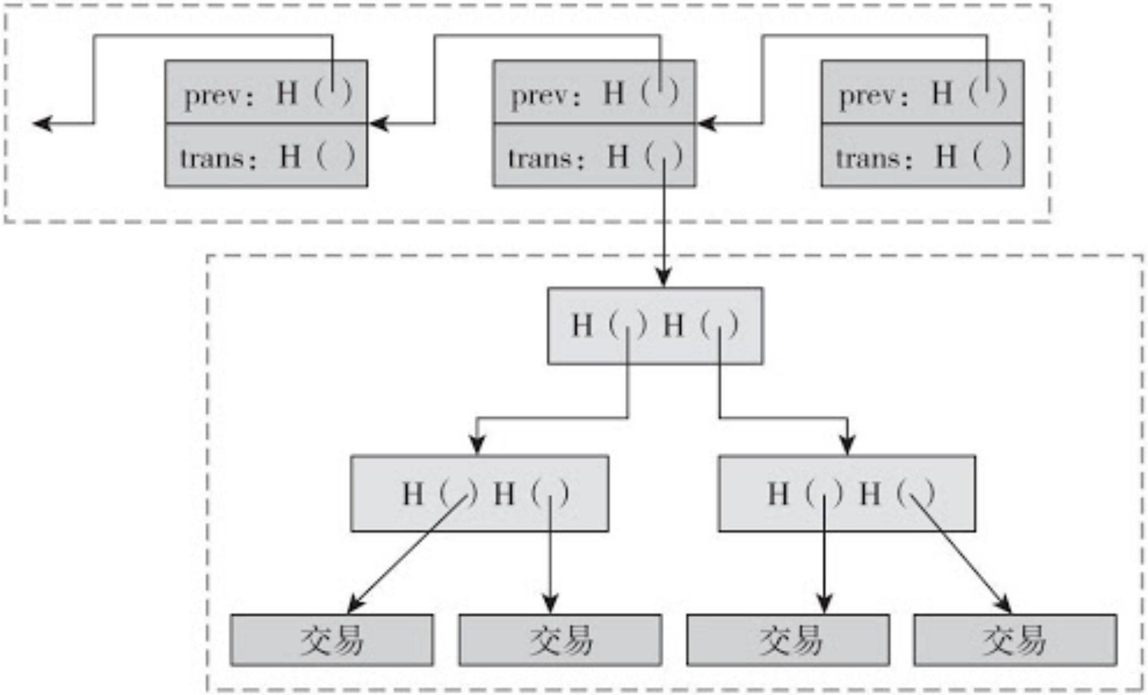
- Why would we say that blockchain can prevent the ledger data from tampering?
  - Ledger is with the *append-only* property
  - If someone modifies any part in previous blocks, we know it immediately. Why?
  - 篡改会顺着链传导 (图示: 2个维度)



Two types of hash in Bitcoin Blockchain

- #1: hash chains
- #2: hashes in Merkle Tree inside each block body

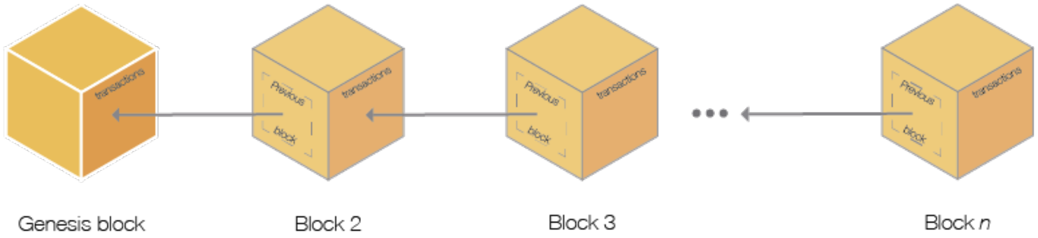
区块的哈希链



每个区块中各笔交易的哈希树（梅克尔树）

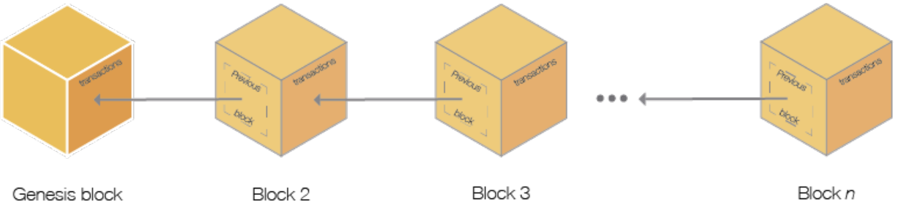
A Question: 为何区块链可以防篡改？

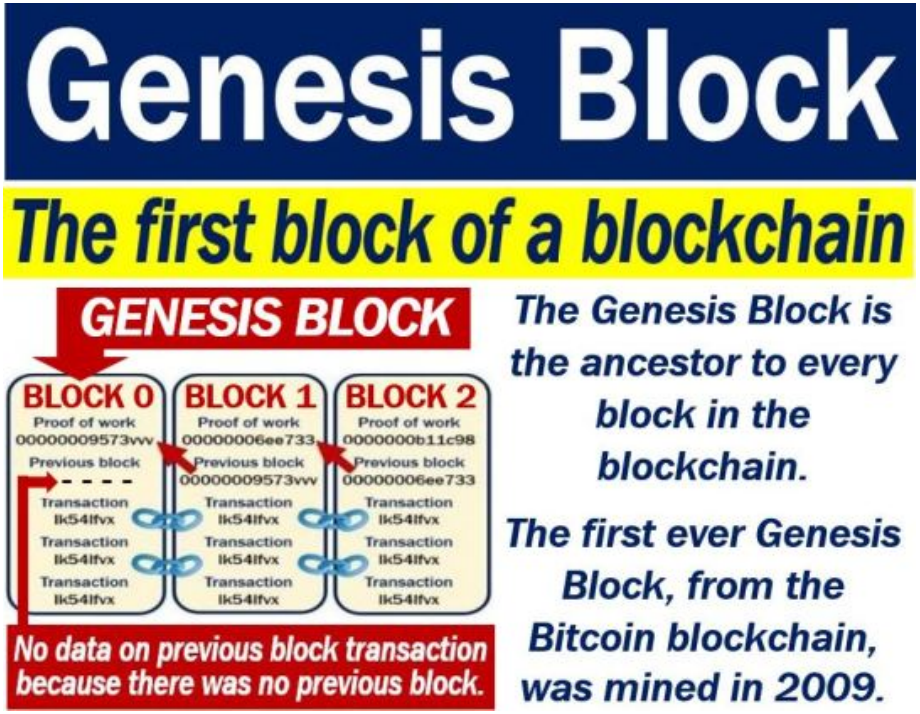
- When an attacker tries to tamper a block data (e.g., a Tx)
  - He may try to keep changing the previous hash pointers
  - Can he make it?
    - No, because we have the Genesis block



Genesis Block

- The first block in any blockchain is termed the genesis block.
- If you start at any block and follow the chain backwards chronologically, you will arrive at the genesis block.
- The genesis block is statically encoded within the client software, that it cannot be changed.




Genesis Block

The first block of a blockchain

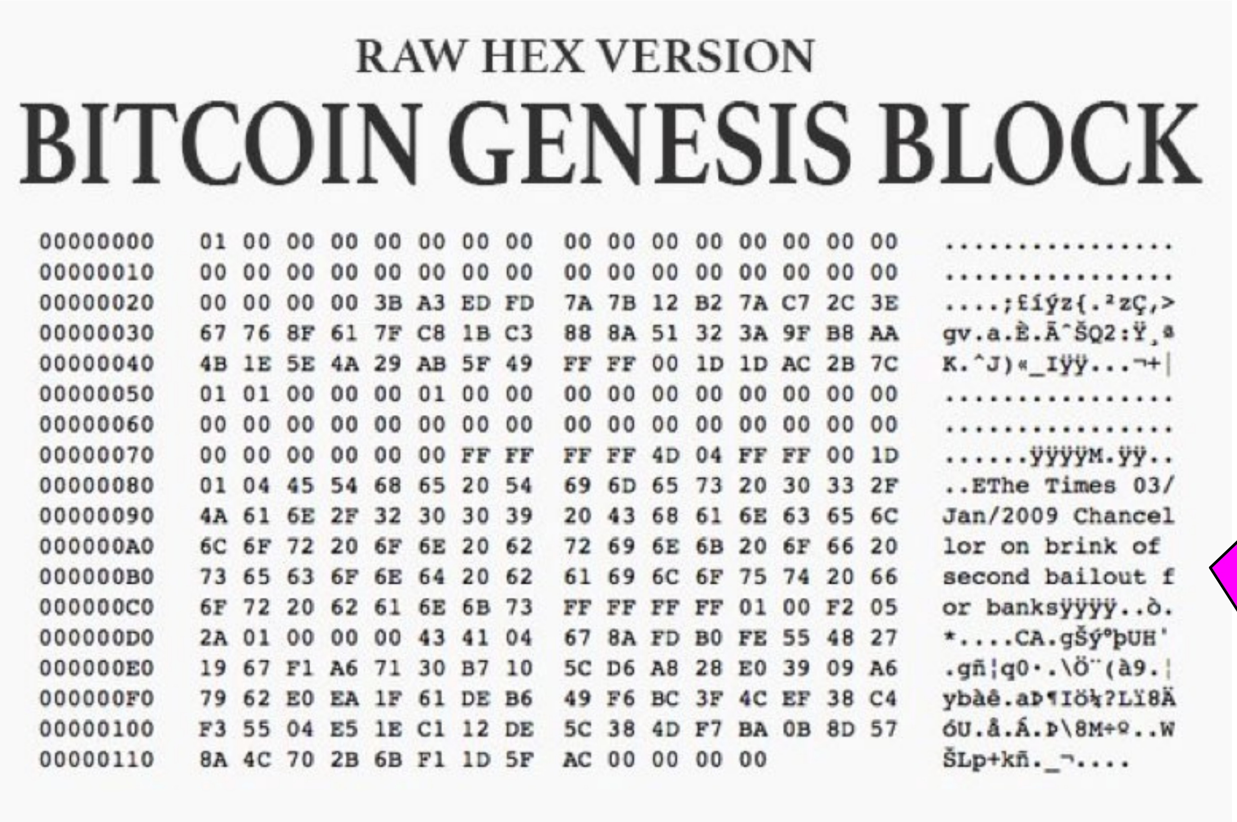
The Genesis Block is the ancestor to every block in the blockchain.

The first ever Genesis Block, from the Bitcoin blockchain, was mined in 2009.

No data on previous block transaction because there was no previous block.

Genesis Block

- Every node can identify the genesis block’s hash and structure, the fixed time of creation, and the single transaction within it.
- Thus, every node has a secure “root”, from which it is possible to build a trusted blockchain.



RAW HEX VERSION

BITCOIN GENESIS BLOCK

```
00000000   01 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00000010   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00000020   00 00 00 00 3B A3 ED FD  7A 7B 12 B2 7A C7 2C 3E   ....;£íýz{.²zÇ,>
00000030   67 76 8F 61 7F C8 1B C3  88 8A 51 32 3A 9F B8 AA   gv.a.È.Ã^ŠQ2:Ÿ¸ª
00000040   4B 1E 5E 4A 29 AB 5F 49  FF FF 00 1D 1D AC 2B 7C   K.^J)«_Iÿÿ...¬+|
00000050   01 01 00 00 00 01 00 00  00 00 00 00 00 00 00 00   ................
00000060   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
00000070   00 00 00 00 00 00 FF FF  FF FF 4D 04 FF FF 00 1D   ......ÿÿÿÿM.ÿÿ..
00000080   01 04 45 54 68 65 20 54  69 6D 65 73 20 30 33 2F   ..EThe Times 03/
00000090   4A 61 6E 2F 32 30 30 39  20 43 68 61 6E 63 65 6C   Jan/2009 Chancel
000000A0   6C 6F 72 20 6F 6E 20 62  72 69 6E 6B 20 6F 66 20   lor on brink of 
000000B0   73 65 63 6F 6E 64 20 62  61 69 6C 6F 75 74 20 66   second bailout f
000000C0   6F 72 20 62 61 6E 6B 73  FF FF FF FF 01 00 F2 05   or banksÿÿÿÿ..ò.
000000D0   2A 01 00 00 00 43 41 04  67 8A FD B0 FE 55 48 27   *....CA.gŠý°þUH'
000000E0   19 67 F1 A6 71 30 B7 10  5C D6 A8 28 E0 39 09 A6   .gñ¦q0·.\Ö¨(à9.¦
000000F0   79 62 E0 EA 1F 61 DE B6  49 F6 BC 3F 4C EF 38 C4   ybàê.aÞ¶Iö¼?LÏ8Ä
00000100   F3 55 04 E5 1E C1 12 DE  5C 38 4D F7 BA 0B 8D 57   óU.å.Á.Þ\8M÷º..W
00000110   8A 4C 70 2B 6B F1 1D 5F  AC 00 00 00 00            ŠLp+kñ._¬....
```

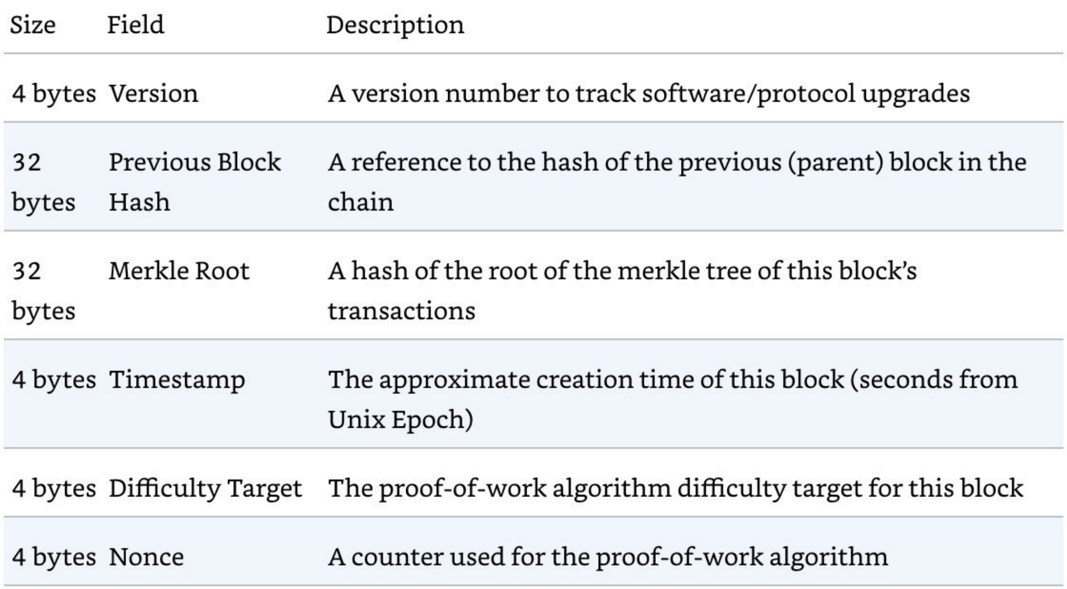
The Times 03/Jan/2009 Chancellor on brink of second bailout (拯救) for banks.

Part 3: A Bitcoin Block

- What does it look exactly?
  - {block header} {transactions}

- What components?
  - Block Header: {version 4B} {previous block hash 32B} {merkle root hash 32B} {time 4B}{bits 4B} {nonce 4B}
  - Transactions: a bunch of TXs

Block Header



Size	Field	Description
4 bytes	Version	A version number to track software/protocol upgrades
32 bytes	Previous Block Hash	A reference to the hash of the previous (parent) block in the chain
32 bytes	Merkle Root	A hash of the root of the merkle tree of this block's transactions
4 bytes	Timestamp	The approximate creation time of this block (seconds from Unix Epoch)
4 bytes	Difficulty Target	The proof-of-work algorithm difficulty target for this block
4 bytes	Nonce	A counter used for the proof-of-work algorithm

- **version** — `01000000` (1)
- **previous block hash** — `0000000000000000000000000000000000000000000000000000000000000000`
- **merkle root hash** — `3ba3edfd7a7b12b27ac72c3e67768f617fc81bc3888a51323a9fb8aa4b1e5e4a`
- **time** — `dae5494d` (1296688602 Wednesday, February 2, 2011 11:16:42 PM GMT)
- **bits** — `ffff7f20` (7fffff0000000000000000000000000000000000000000000000000000000000)
- **nonce** — `02000000` (decimal 2)

总结

- Hash Pointer
- Merkle Tree
- What is inside a Bitcoin’s block

加入我们！

- InplusLab
  - 8位教授/副教授导师，研究方向涉及 区块链、人工智能、数据挖掘、分布式学习，物联网，等等方向
  - 请随时投简历到邮箱：inpluslab@yeah.net

- 如果对我(黄华威)的研究方向感兴趣,
  - 可以提前联系我内推
  - 请邮件联系： huanghw28@mail.sysu.edu.cn
  - HuangLab 主页: http://xintelligence.pro/

InplusLab
实验室公众号