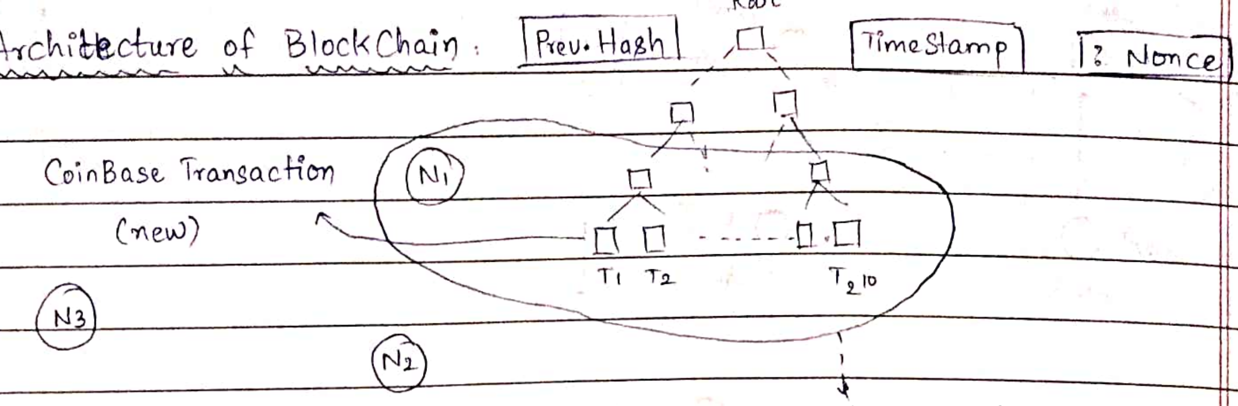
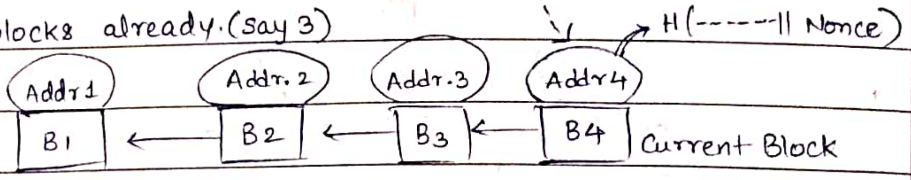
27/2/24

Architecture of BlockChain :



Sps N1 has some blocks already.(say 3)



Prev. Hash for B4 = Addr. 3

$B_4 = T_1 \| T_2 \| \cdots \| T_{1024} \| MR \| TimeStamp \| Prev.Hash \| Nonce$

MR ↓ Merkel Root

Nonce - is a solution to a puzzle

↳ (Puzzle 2) ⟹ $H(T_1 \| T_2 \| \cdots \| T_{1024} \| Prev.Hash \| x)$

Can you compute x such that first 100 bits are 0? $= \boxed{\underbrace{000\cdots0}_{100 \text{ bits}}\cdots}$

$H(11110011 \| x) = \underbrace{00\cdots00}_{256 \text{ bits}}$ ⟹ Avg. no:of trials needed to compute $x = 2^{256}$ (very huge)

Formation of Block is called as "Mining".

Once you mine a block, you'll be rewarded with money.

Coinbase Transaction doesn't have previous transaction.

All transactions other than Coinbase Transaction are old.

With Coinbase Transaction, new money comes into circulation.

Presently, Value of coinbase Transaction is 12.5 BTC.

CoinBase Transaction

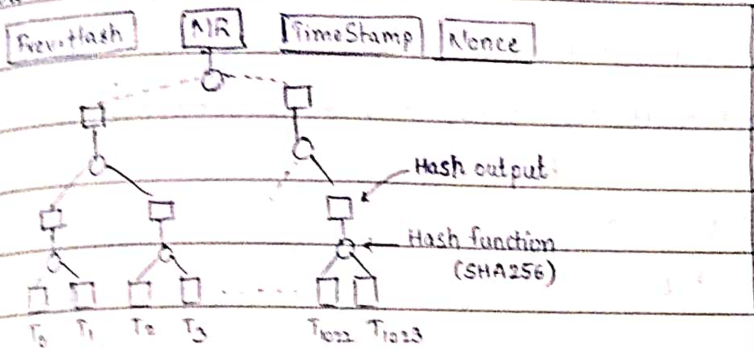
MD	H(-)	TL=0
IN	00----0	
	n	
	COINBASE	
OUT		V=125
	$Ver(Pub_{N_1}, ?) \overset{?}{=} 1$	

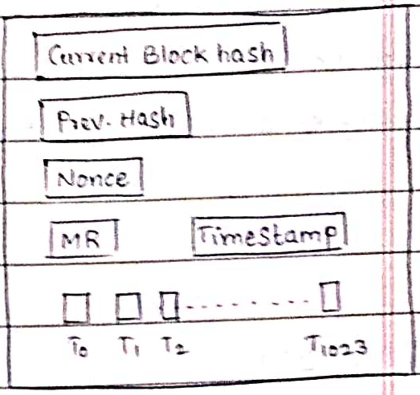
Input has no meaning.

[Q] If 1000 blocks are created each by 1 node (1000 nodes), which of the ~~Bl~~ block to be taken ?

24/2/24

How B4 is generated?





Block B4

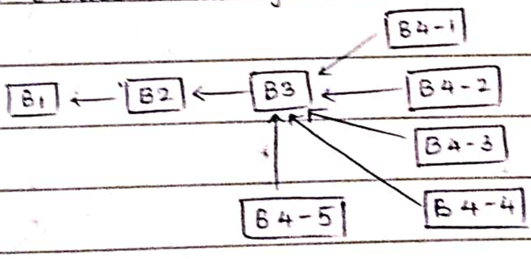
T_0 is COINBASE transaction in B4 which is created by N_1

Computation of Nonce is hard. So creation of Block is difficult. Creation of block is done to get Bitcoins. (12.5 BTC) - Economic incentive.

After N_1 generates legitimate Block B4, it broadcasts this information to all the nodes. Any individual node is suspicious. Verification is easy.

Each node verifies the new block and then updates Blockchain.

If 5 nodes creates 5 new legitimate blocks



Every node stores all the blocks in their storages.

Longest chain (which block further grows linearly faster) is Main Chain and are valid. Transactions which are not on longest chain are not valid.

29/1/24

Every node has Transaction and Block. Block creation is difficult.

[Q] Computation of Nonce has to be difficult. why?.

ans: This brings Immutability property to Block chain.

Blockchain → Transaction Creation, Transaction Broadcast, Block Creation, Block Broadcast

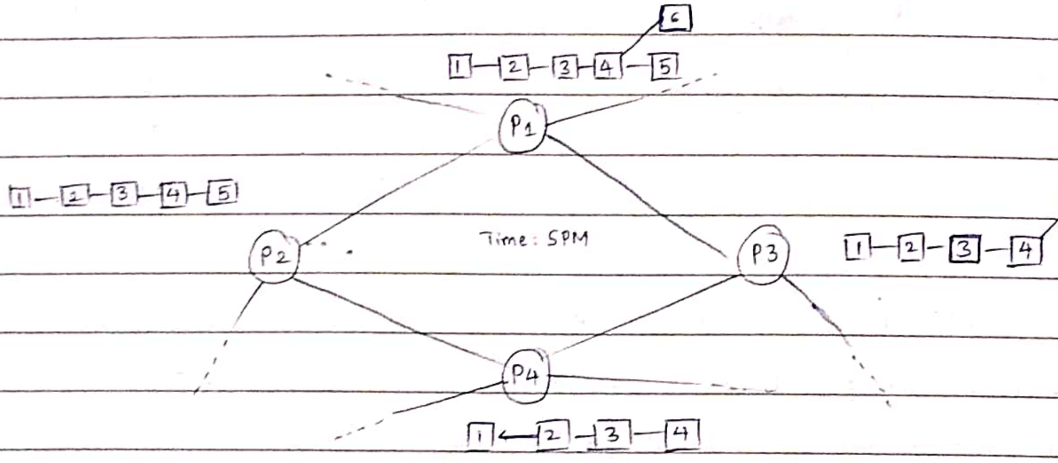
20 conditions are checked by a node for transaction validation (critically signature)

Creation of Block is called "Mining".

19 Conditions are checked by node for Block validation(Nonce checking is critical)

Double Spending doesn't means "Discarding of Block". Double Spending concerns Legal Transactio

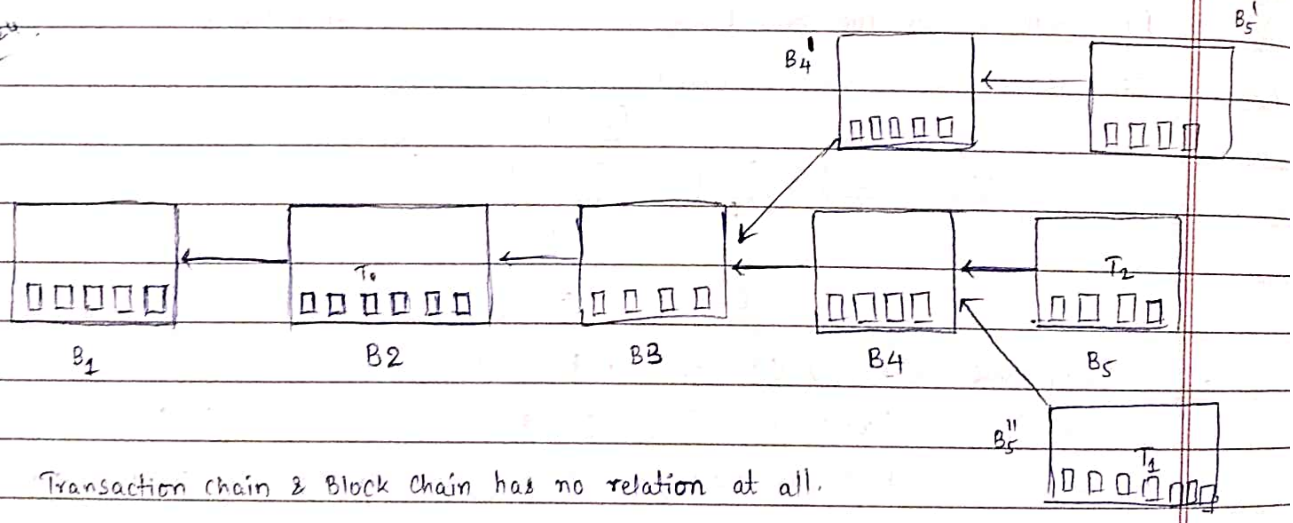
Asymmetry of Blockchain - due to network disruptions, delay or power outage.



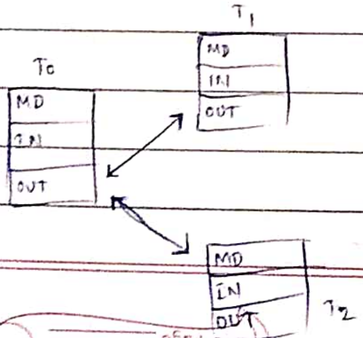
Take Longest Chain if there are branches. (after a while, only 1 chain is valid)

Block 5 isn't illegal (It's called Orphan Block). Orphan Blocks are legal.

6/2/24



Transaction chain & Block Chain has no relation at all.



So, there are 25 BTC pledged to Alice by T0

Alice creates 2 transactions (T1 & T2) for redeeming

25 BTC twice (Double Spending)

One for Flipkart & One for Amazon

B_5', B_5'' are created by two diff. miners.

T_1 & T_2 can never be in same block.

T_1 and T_2 can not be in same chain even though they are created by different blocks.

7th block from Top Block(Recent Block)

How do I know my block is in Longest chain.?

After fetching raw data, we can create our own dashboard

Derived Data-

12/3/24

13/3/24

Hash function, $h : \{0,1\}^* \rightarrow \{0,1\}^{256}$

Block_Header ⇒ (Nonce, MR, B, Time, V, PB)

B (Bits), Time, V, PB (Previous Block Hash), Transactions are fixed

Choose a nonce x

{ Difficulty = Prev_Difficulty × 2016 × 10 / (Time to mine 2016 last blocks)

Initial Difficulty = 1.

First 2016 blocks have Difficulty as 1.

{ Target : 2^{224} / Difficulty.

We reserve 32 bits outof 256 bits for somethingelse.

Bits = Index (8 bits) || Coefficient (24 bits).

Compute Bits such that

$$\text{Target} = \text{Coefficient} \times 2^{(8 \times \text{Index} - 3)}$$

9/3/24

[Q] What is the "STATE" of Bitcoin?

STATE is like "Books of Accounts".

Cash Book | General ledger | Loan Register | Deposit Register

⇧

Transactions with Customers (Cheques, Currency Notes, FDS, Share certificates, bond Cer.)

Like Transactions - Raw data

State - Account by account log books.

↳ Derived information as ledger.

Ethereum:

Same working principle as Bitcoin - Transactions, Blocks & Blockchains.

Difference: State of Ethereum has Contract Accounts in addition to User accounts (ext. owned)

State of bitcoin has only User Account.

Question: What How and Where : Contract Account ?

Contract is a "Condition" and could have account balance.

↳ actually a "Piece of program" which you can execute and compile.

Puzzle in bitcoin is generalised in Ethereum (and assigned a value).

State - General ledger kept acc. to account by account

Transaction

User Account

Receiver Account

Value

Signature (...)

State

User A/c : 123
Value : 50 BTC

User A/c : 456
Value : 30 BTC

State is maintained by User in local Machine.,

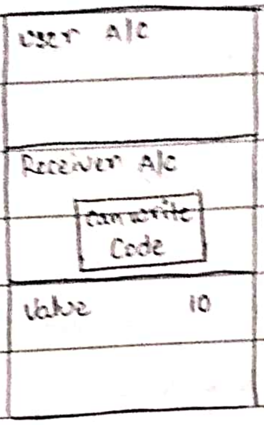
State is never broadcasted.

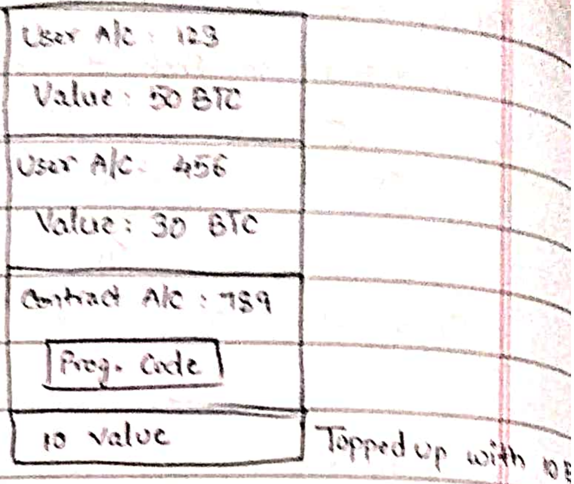
↳ Reorganising data in Transaction. contained.

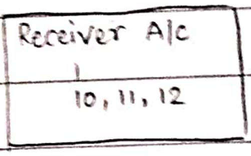
Tsf Money, Contract Creation & Contract Execution

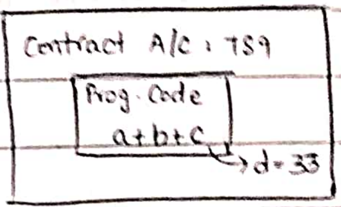
Rather than giving Receiver's Address and Receiver account,

it can give "Programming Code".









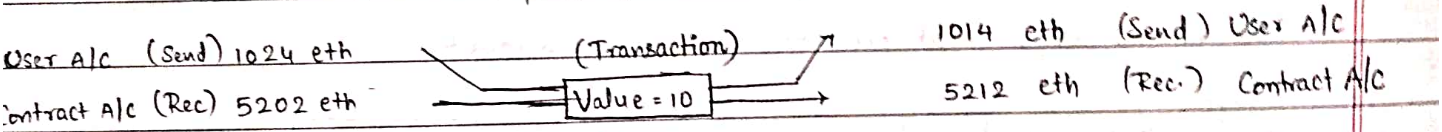
0/3/24 tx → transaction

Ethereum is quite efficient than Bitcoin. (can write any code easily)

Signature (who signs?)

Using Transactions, I have run a program.

Contract is most imp. (dispute stuff) for Business Transaction.

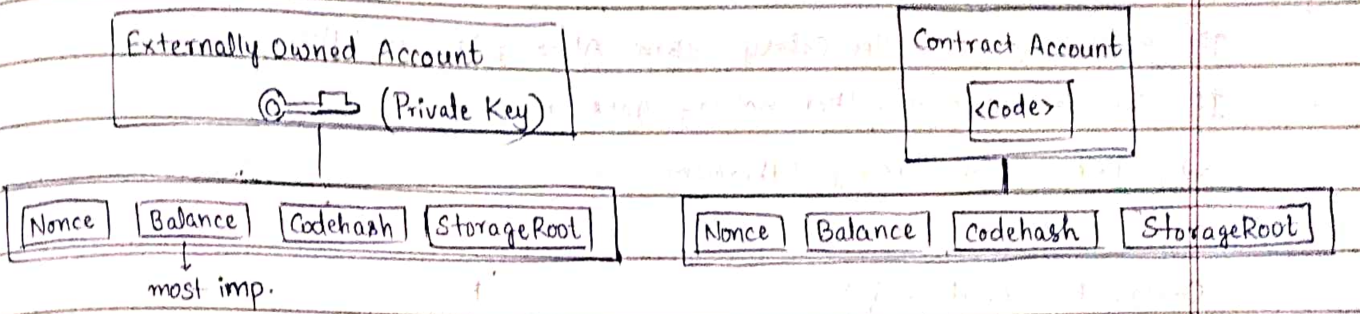


State is NOT Immutable. Transactions are Immutable ⟹ Info. never lost.

Ethereum is Immutable.

1/4/24

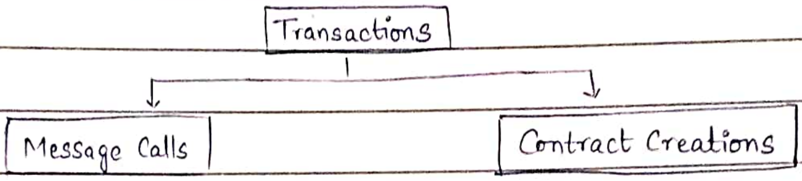
Types of Accounts

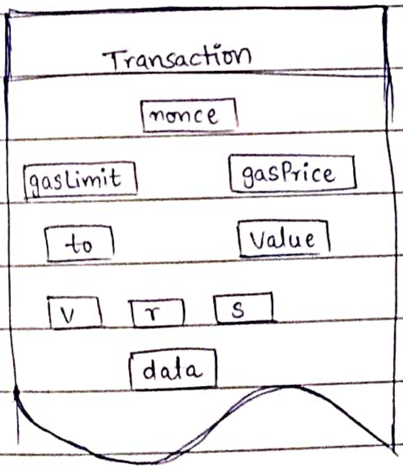


Contract account can be created by any account, not associated with Private Key

Communication b/w Accounts takes place through "State".
(Tsf. data) - using Transaction (money/info)

Transaction is effectively local computation & Communication.
↳ invokes/calls/triggers functions.





Signature verification (v, r, s together)

Internal Transactions are kind of "virtual"
↳ (contract account invokes/calls another contract account).

[5] If Alice salary > Bob Salary then Bob gets 10 Ether.
If Bob Salary > Alice Salary then Alice gets 10 Ether.
If Salaries are equal, then nobody gets reward.
Solve this problem using Ethereum.

```
Contract (Input a, b)
                    b → Salary of Bob
                    a → Salary of Alice
    Value : 20 ether
    If a>b :
        Send 10 ether to Bob.
    if b>a :
        Send 10 ether to Alice.
```

Protocol.

Step1 : Alice calls Contract

Step2 : Bob calls Contract

9/4/24

Protocol.

Alice input -

Bob input -

1. Alice : deposits 10 eth from his account to $\overbrace{\text{DEPOSIT A/c}}^{\text{contract account}}$

2. Bob :

3. Alice generates a binary string x, hashes it, and stores it

$h \leftarrow H(x)$

$\text{STORE}(h)$

4. Bob :

5. Alice - STORE(X)
6. Bob - STORE(Y) → checks whether hashes are Original values/not.
7. Alice - VERIFY_WINNER
8. WINNER- TRANSFER_REWARD