Batch: BCT-1 Roll No.: 1911031

Experiment No. 03

Title: Implementation of PoS and PoW in block chain

Objective: To understand the concept of various consensus algorithms in blockchain and to implement the most commonly used consensus algorithms, proof of stake and proof of work.

Expected Outcome of Experiment:

CO	Outcome
CO4	Grasp the in-depth understanding of Blockchain, Smart Contracts & how it works.

Books/ Journals/ Websites referred:

1. https://www.geeksforgeeks.org/consensus-algorithms-in-blockchain/
2. https://www.investopedia.com/terms/c/consensus-mechanism-cryptocurrency.asp
3. https://ethereum.org/en/developers/docs/consensus-mechanisms/
4. https://www.naukri.com/learning/articles/consensus-mechanisms-in-blockchain/
5. https://www.allerin.com/blog/8-blockchain-consensus-mechanisms-you-should-know-about

Abstract:-

WHAT IS CONSENSUS?

By consensus, we mean that a general agreement has been reached. Consider a group of people going to the cinema. If there is no disagreement on a proposed choice of film, then a consensus is achieved. If there is disagreement, the group must have the means to decide which film to see. In extreme cases, the group will eventually split.

In regards to the Ethereum blockchain, the process is formalized, and reaching consensus means that at least 66% of the nodes on the network agree on the global state of the network.

Related Theory: -

WHAT IS A CONSENSUS MECHANISM?

The term consensus mechanism refers to the entire stack of protocols, incentives and ideas that allow a network of nodes to agree on the state of a blockchain.

Ethereum uses a proof-of-stake-based consensus mechanism that derives its crypto-economic security from a set of rewards and penalties applied to capital locked by stakers. This incentive structure encourages individual stakers to operate honest validators, punishes those who don't, and creates an extremely high cost to attack the network.

Then, there is a protocol that governs how honest validators are selected to propose or validate blocks, process transactions and vote for their view of the head of the chain. In the rare situations where multiple blocks are in the same position near the head of the chain, there is a fork-choice mechanism that selects blocks that make up the 'heaviest' chain, measured by the number of validators that voted for the blocks weighted by their staked ether balance.

Some concepts are important to consensus that are not explicitly defined in code, such as the additional security offered by potential out-of-band social coordination as a last line of defense against attacks on the network.

Related Theory (contd...): -

In any centralized system, like a database holding key information about driving licenses in a country, a central administrator has the authority to maintain and update the database. The task of making any updates—like adding/deleting/updating names of people who qualified for certain licenses—is performed by a central authority who remains the sole in-charge of maintaining genuine records.

Public blockchains that operate as decentralized, self-regulating systems work on a global scale without any single authority. They involve contributions from hundreds of thousands of participants who work on verification and authentication of transactions occurring on the blockchain, and on the block mining activities.

In such a dynamically changing status of the blockchain, these publicly shared ledgers need an efficient, fair, real-time, functional, reliable, and secure mechanism to ensure that all the transactions occurring on the network are genuine and all participants agree on a consensus on the status of the ledger. This all-important task is performed by the consensus mechanism, which is a set of rules that decides on the legitimacy of contributions made by the various participants (i.e., nodes or transactors) of the blockchain.

Blockchain Consensus Mechanisms

There are different kinds of consensus mechanism algorithms, each of which works on different principles.

The proof of work (PoW) is a common consensus algorithm used by the most popular cryptocurrency networks like bitcoin and litecoin. It requires a participant node to prove that the work done and submitted by them qualifies them to receive the right to add new transactions to the blockchain. However, this whole mining mechanism of bitcoin needs high energy consumption and a longer processing time.

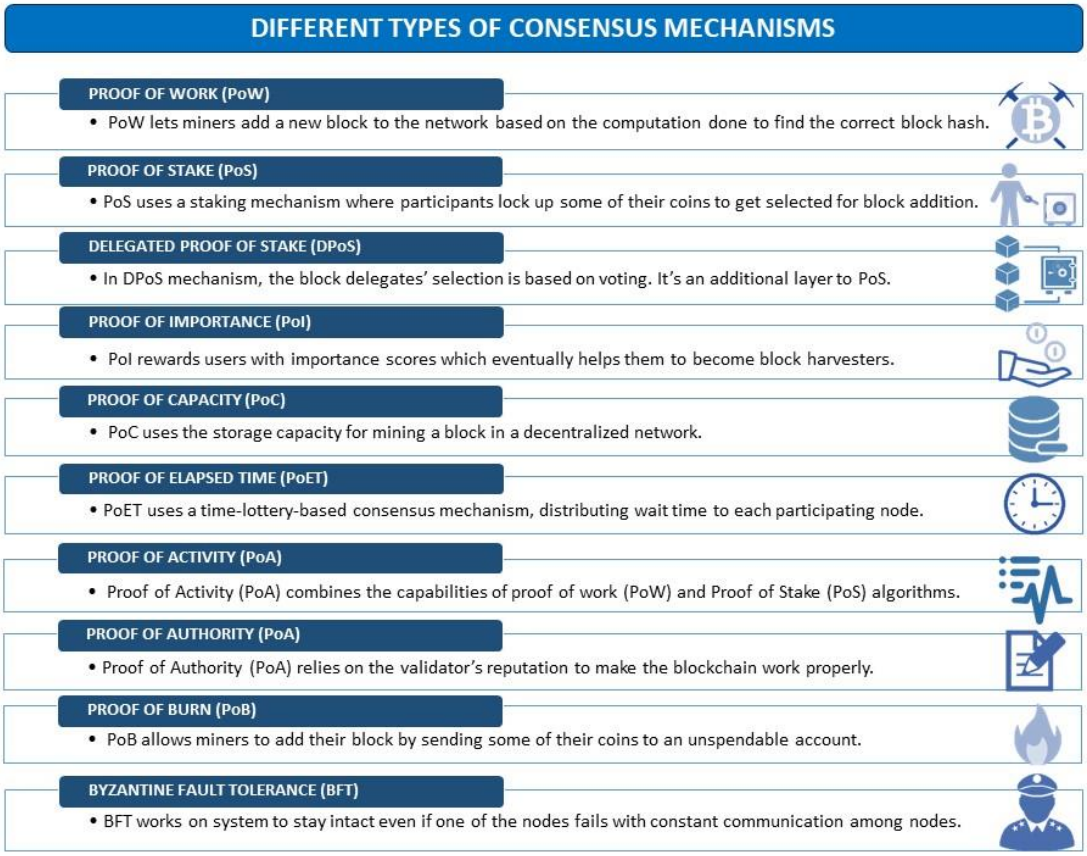
The proof of stake (PoS) is another common consensus algorithm that evolved as a low-cost, low-energy consuming alternative to the PoW algorithm. It involves the allocation of responsibility in maintaining the public ledger to a participant node in proportion to the number of virtual currency tokens held by it. However, this comes with the drawback that it incentivizes cryptocoin hoarding instead of spending.

While PoW and PoS are by far the most prevalent in the blockchain space, there are other consensus algorithms like Proof of Capacity (PoC) which allow sharing of memory space of the contributing nodes on the blockchain network. The more memory or hard disk space a node has, the more rights it is granted for maintaining the public ledger. Proof of Activity (PoA), used on the Decred blockchain, is a hybrid that makes use of aspects of both PoW and PoS. Proof of Burn (PoB) is another that

requires transactors to send small amounts of cryptocurrency to inaccessible wallet addresses, in effect "burning" them out of existence.

Another, called Proof of History (PoH), developed by the Solana Project and similar to Proof of Elapsed Time (PoET), encodes the passage of time itself cryptographically to achieve consensus without expending many resources.

Types of Consensus Mechanisms:



Proof of Work (PoW)

PoW is a popular consensus algorithm used by Bitcoin and Ethereum networks. Here, miners (or block adders) have to do heavy mathematical computations to find a right hash by changing the nonce of the block. The miner who finds the hash below the difficulty level gets the chance to add his block to the network. Hence, takes the reward.

It's a puzzle-friendly way to reach consensus by using high computational power. Afterward, already present network participants valid transactions in the block added by the miner.

Blockchains using PoW algorithm: Bitcoin, Ethereum, Dogecoin, Litecoin, Zcash, Horizon, and many more.

Proof of Stake (PoS)

PoS consensus eliminates the high energy consumption by PoW. PoS uses a staking mechanism in which miners (or validators) hold some of their earned coins in the network to get selected for adding a block.

It's not an initial consensus algorithm for a network. It can only be implemented after a network gets a good amount of participants (or nodes).

Blockchains using PoS algorithm: Polkadot, EOSIO, Cardano, Ethereum 2.0, and many more.

Delegated Proof of Stake (DPoS)

DPoS improves the PoS mechanism by introducing voting for delegates. Here, network participants vote for the trusted delegates (or miners) using their coins. Then, based on a random selection, one voted delegate gets the chance to add its block.

Blockchains using the DPoS algorithm: EOS, Lisk, Ark y Tron

Proof of Importance (PoI)

PoI uses importance scores to select the one block harvester out of all participants. It aims to eliminate favors toward rich stakeholders in PoS consensus. The importance score depends on your quality transactions and reputation in the network.

Blockchain using PoI algorithm: New Economy Movement (NEM)

Proof of Capacity (PoC)

PoC uses the disk or storage capacity for mining a block in a decentralized network. It exchanges the computation factor with disk space. The PoC motivates miners to collect a list of all the possible nonce and block hashes before the actual mining.

At the time, the miner just uploads the calculated files of possible hashes to the network. PoC reduces the time taken to add and validate the block of transactions.

Blockchains using PoC algorithm: Burstcoin, Storj, Chia, and SpaceMint.

Proof of Elapsed Time (PoET)

PoET mechanism uses time-lottery-based concepts. It distributes random waiting times to each miner. For that waiting time the miner node sleeps, the first woken up node (or short waiting time node) gets the chance to add its block to the network.

Afterward, the block verification takes place by network validators, and a new block gets added.

Blockchain using PoET algorithm: Hyperledger Sawtooth

Proof of Activity (PoA)

Proof of Activity (PoA) combines PoW and PoS mechanisms. First, the miners must do the heavy computation to add an empty block with header information and reward address.

Afterward, one empty block gets chosen based on the number of coins they hold in their respective accounts. Then, the miner of that empty block gets the chance to add its transactions to the block. Moreover, the transactions are verified by network validators.

Proof of Authority (PoA)

Proof of Authority (PoA) consensus utilizes by private or permissioned blockchain networks. PoA highly depends on the reputation of the miner or the network participant who wishes to add a new block of transactions. Here, miners stake their reputation instead of coins.

Blockchain using PoA algorithm: VeChain

Proof of Burn (PoB)

PoB allows miners to add their block by sending some of their coins to an unspendable account. This process of sending your earned coins to an escrow account is called burning the coins.

PoB eliminated the burnt coins permanently from regular transactions. Hence, they become unspendable even by its owner.

The more coins a miner burns, the higher his chances of adding his new block of transactions to the network. Burning coins brings virtual mining rights to the miner.

Blockchain using PoB algorithm: Slimcoin

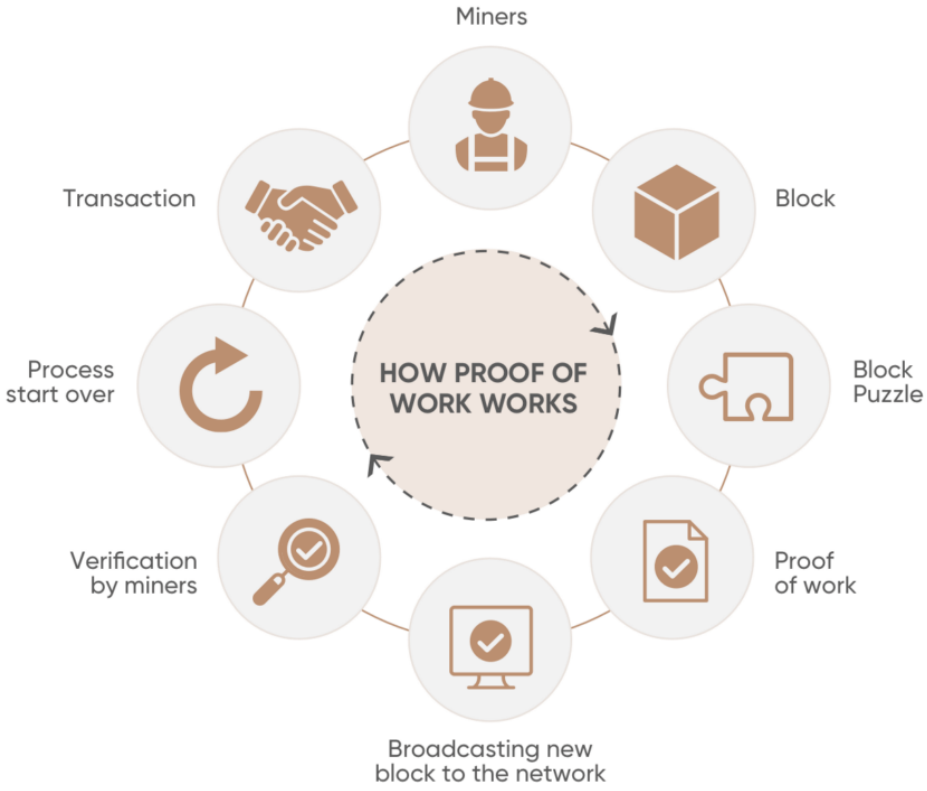
Byzantine Fault Tolerance (BFT)

BFT aims to resolve Byzantine Generals' decisional puzzle**.** It's based on the communication problem generals of different armies might have to decide to attack or retreat at the same time.

BFT mechanism regulates the communication between nodes using hashes, digital signatures, and metadata. It embraces the synchronization among nodes of a decentralized network.

Blockchains using BFT algorithm**:** Hyperledger Fabric and Zilliqa

Proof of Work (PoW)



Proof of Work consensus is the mechanism of choice for the majority of cryptocurrencies currently in circulation. The algorithm is used to verify the transaction and create a new block in the blockchain. The idea for Proof of Work(PoW) was first published in 1993 by Cynthia Dwork and Moni Naor and was later applied by Satoshi Nakamoto in the Bitcoin paper in 2008. The term "proof of work" was first used by Markus Jakobsson and Ari Juels in a publication in 1999. Cryptocurrencies like Ethereum, Litecoin, and Bitcoin are currently using PoW.

Principle: A solution that is difficult to find but is easy to verify.

Purpose of PoW

The **purpose** of a consensus mechanism is to bring all the nodes in agreement, that is, trust one another, in an environment where the nodes don't trust each other.

- All the transactions in the new block are then validated and the new block is then added to the blockchain.
- The block will get added to the chain which has the longest block height(see blockchain forks to understand how multiple chains can exist at a point in time).

- Miners(special computers on the network) perform computation work in solving a complex mathematical problem to add the block to the network, hence named, Proof-of-Work.
- With time, the mathematical problem becomes more complex.

Features of PoW

There are mainly two features that have contributed to the wide popularity of this consensus protocol and they are:

- It is hard to find a solution to a mathematical problem.
- It is easy to verify the correctness of that solution.

How Does PoW Work?

The PoW consensus algorithm involves verifying a transaction through the mining process. This section focuses on discussing the mining process and resource consumption during the mining process.

Mining:

The Proof of Work consensus algorithm involves solving a computationally challenging puzzle in order to create new blocks in the Bitcoin blockchain. The process is known as 'mining', and the nodes in the network that engages in mining are known as 'miners'.

- The incentive for mining transactions lies in economic payoffs, where competing miners are rewarded with 6.25 bitcoins and a small transaction fee.
- This reward will get reduced by half its current value with time.

Energy and Time consumption in Mining:

The process of verifying the transactions in the block to be added, organizing these transactions in chronological order in the block, and announcing the newly mined block to the entire network does not take much energy and time.

- The energy-consuming part is solving the 'hard mathematical problem' to link the new block to the last block in the valid blockchain.
- When a miner finally finds the right solution, the node broadcasts it to the whole network at the same time, receiving a cryptocurrency prize (the reward) provided by the PoW protocol.

Mining reward:

- Currently, mining a block in the bitcoin network gives the winning miner 6.25 bitcoins.
- The amount of bitcoins won halves every four years. So, the next deduction in the amount of bitcoin is due at around 2024(with the current rate and growth).

- With more miners comes the inevitability of the time it takes to mine the new block getting shorter.
- This means that the new blocks are found faster. In order to consistently find 1 block every 10 minutes. (That is the amount of time that the bitcoin developers think is necessary for a steady and diminishing flow of new coins until the maximum number of 21 million is reached (expected some time with the current rate in around 2140)), the Bitcoin network regularly changes the difficulty level of mining a new block.

Bitcoin's PoW System

Bitcoin uses the Hashcash Proof of Work system as the mining basis. The **'hard mathematical problem'** can be written in an abstract way like below :
Given data A, find a number x such as that the hash of x appended to A results is a number less than B.

- The miners bundle up a group of transactions into a block and try to mine. To mine it, a hard mathematical problem has to be solved.
- This problem is called the proof of work problem which has to be solved to show that the miner has done some work in finding out the solution to the problem and hence the mined block must be valid.
- The answer to the problem needs to be a lower number than the hash of the block for it to be accepted, known as the '**target hash**'.

A **target hash** is a number that the header of a hashed block must be equal to or less than for a new block, along with the reward, to be awarded to a miner.
The lower a target is, the more difficult it is to generate a block.

- A miner continues testing different unique values (known as a nonce(s)) until a suitable one is produced.
- The miner who manages to solve the problem gets the bitcoin reward and adds the block to the blockchain by broadcasting that the block has been mined.

Note: The target hash adjusts once every 2016 block or approximately once every 2 weeks. All the miners immediately stop working on the said block and start mining the next block.
Common cryptographic protocols used in PoW: The most widely used proof-of-work consensus is based on SHA-256 and was introduced as a part of Bitcoin. Others include Scrypt, SHA-3, scrypt-jane, scrypt-n, etc.

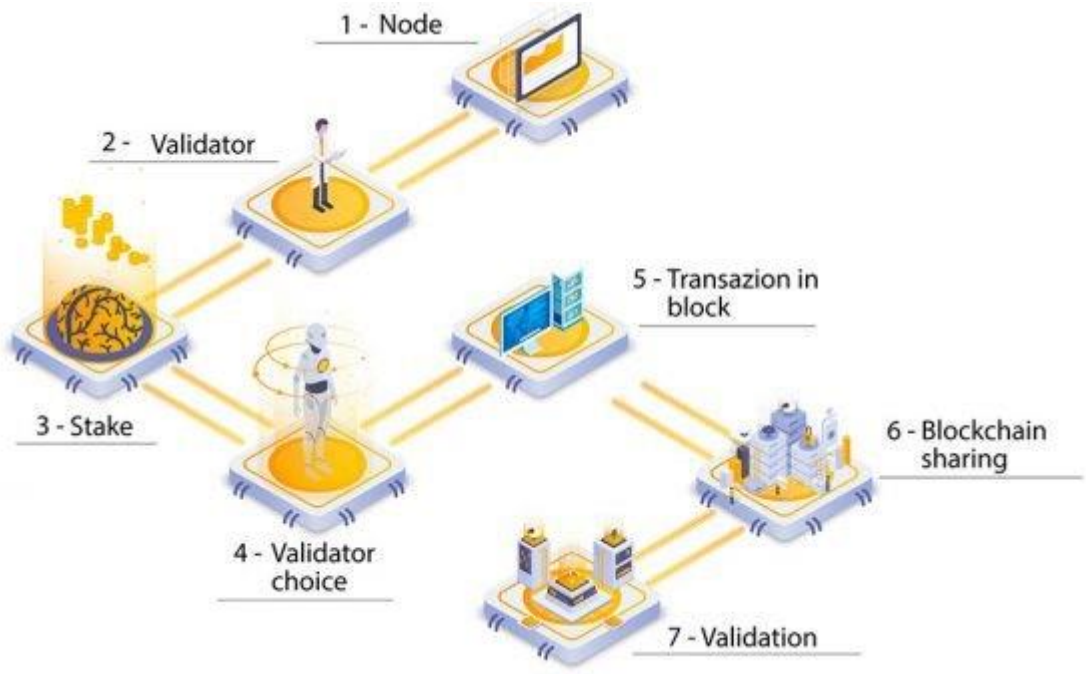
Challenges With PoW

The Proof-of-Work consensus mechanism has some issues which are as follows:

- **The 51% risk**: If a controlling entity owns 51% or more than 51% of nodes in the network, the entity can corrupt the blockchain by gaining the majority of the network.

- **Time-consuming**: Miners have to check over many nonce values to find the right solution to the puzzle that must be solved to mine the block, which is a time-consuming process.
- **Resource consumption**: Miners consume high amounts of computing power in order to find the solution to the hard mathematical puzzle. It leads to a waste of precious resources(money, energy, space, hardware). It is expected that 0.3% of the world's electricity will be spent to verify transactions by the end of 2018.
- **Not instantaneous transaction:** Transaction confirmation takes about 10–60 minutes. So, it is not an instantaneous transaction; because it takes some time to mine the transaction and add it to the blockchain thus committing the transaction.

Proof of Stake (PoS)



Proof of Stake (PoS) is a type of algorithm which aims to achieve distributed consensus in a Blockchain. This way to achieve consensus was first suggested by

Quantum Mechanic here and later Sunny King and his peer wrote a paper on it. This led to Proof-of-Stake (PoS) based Peercoin.
A **stake** is value/money we bet on a certain outcome. The process is called staking.
A more particular meaning of stake will be defined later on.

Why Proof-of-Stake:
Before proof of stake, the most popular way to achieve distributed consensus was through Proof-of-Work (implemented in Bitcoin). But Proof-of-Work is quite energy(electrical energy in mining a bitcoin) intensive. So, a proof-of-work based consensus mechanism increases an entity's chances of mining a new block if it has more computation resources. Apart from the upper two points, there are other weaknesses of a PoW based consensus mechanism which we will discuss later on. In such a scenario, a Proof-of-Stake based mechanism holds merit.

What is Proof-of-Stake:
As understandable from the name, nodes on a network stake an amount of cryptocurrency to become candidates to validate the new block and earn the fee from it. Then, an algorithm chooses from the pool of candidates the node which will validate the new block. This selection algorithm combines the quantity of stake (amount of cryptocurrency) with other factors (like coin-age based selection, randomization process) to make the selection fair to everyone on the network.

- **Coin-age based selection:**
  The algorithm tracks the time every validator candidate node stays a validator. The older the node becomes, the higher the chances of it becoming the new validator.
- **Random Block selection:**
  The validator is chosen with a combination of 'lowest hash value' and 'highest stake'. The node having the best weighted-combination of these becomes the new validator.

A typical PoS based mechanism workflow:

- Nodes make transactions. The PoS algorithm puts all these transactions in a pool.
- All the nodes contending to become validator for the next block raise a stake. This stake is combined with other factors like 'coin-age' or 'randomized block selection' to select the validator.
- The validator verifies all the transactions and publishes the block. His stake still remains locked and the forging reward is also not granted yet. This is so that the nodes on the network can 'OK' the new block.
- If the block is 'OK'-ed, the validator gets the stake back and the reward too. If the algorithm is using a coin-age based mechanism to select validators, the

validator for the current block's has its coin-age reset to 0. This puts him in a low-priority for the next validator election.

- If the block is not verified by other nodes on the network, the validator loses its stake and is marked as 'bad' by the algorithm. The process again starts from step 1 to forge the new block.

Features:

- **Fixed coins in existence:**
  There is only a finite number of coins that always circulate in the network. There is no existence of bringing new coins into existence(as in by mining in case of bitcoin and other PoW based systems). Note that the network starts with a finite number of coins or 'initially starts with PoW, then shifts to PoS' in some cases. This initiation with PoW is meant to bring coins/cryptocurrency in the network.
- **Transaction fee as reward to minters/forgers:**
  Every transaction is charged some amount of fee. This is accumulated and given to the entity who forges the new block. Note that if the forged block is found fraudulent, the transaction fee is not rewarded. Moreover, the stake of the validator is also lost(which is also known as **slashing**).
- **Impracticality of the 51% attack:**
  To conduct a 51% attack, the attacker will have to own 51% of the total cryptocurrency in the network which is quite expensive. This deems doing the attack too tedious, expensive and not so profitable. There will occur problems when amassing such a share of total cryptocurrency as there might not be so much currency to buy, also that buying more and more coins/value will become more expensive. Also validating wrong transactions will cause the validator to lose its stake, thereby being reward-negative.

Advantages of PoS:

- **Energy-efficient:**
  As all the nodes are not competing against each other to attach a new block to the blockchain, energy is saved. Also, no problem has to be solved(as in case of Proof-of-Work system) thus saving the energy.
- **Decentralization:**
  In blockchains like Bitcoin(Proof of Work system to achieve distributed consensus), an extra incentive of exponential rewards are in place to join a mining pool leading to a more centralized nature of blockchain. In the case of a Proof-of-Stake based system(like Peercoin), rewards are proportional(linear) to the amount of stake. So, it provides absolutely no extra edge to join a mining pool; thus promoting decentralization.

- **Security:**
  A person attempting to attack a network will have to own 51% of the stakes(pretty expensive). This leads to a secure network.

Weakness of a PoS mechanism:

- **Large stake validators:**
  If a group of validator candidates combine and own a significant share of total cryptocurrency, they will have more chances of becoming validators. Increased chances lead to increased selections, which lead to more and more forging reward earning, which lead to owning a huge currency share. This can cause the network to become centralized over time.
- **New technology:**
  PoS is still relatively new. Research is ongoing to find flaws, fix them and making it viable for a live network with actual currency transactions.
- **The 'Nothing at Stake' problem:**
  This problem describes the little to no disadvantage to the nodes in case they support multiple blockchains in the event of a blockchain split(blockchain forking). In the worst-case scenario, every fork will lead to multiple blockchains and validators will work and the nodes in the network will never achieve consensus.

Blockchains using Proof-of-Stake:

- Ethereum(Casper update)
- Peercoin
- Nxt

Variants of Proof-of-Stake:

- Regular Proof-of-Stake – The one discussed in this article.
- Delegated Proof-of-Stake
- Leased Proof-of-Stake
- Masternode Proof-of-Stake

Implementation Details:

1. Enlist all the Steps followed and various options explored

Proof of Work

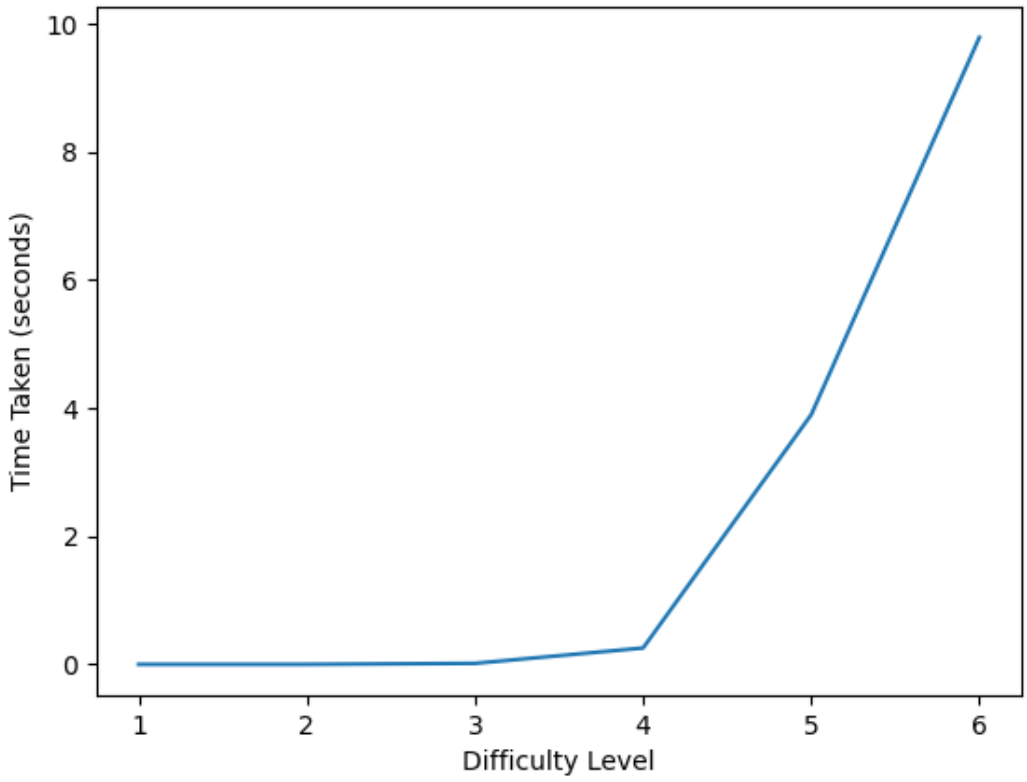
```
import time
import hashlib
import matplotlib.pyplot as plt

DIFFICULTY_LEVEL = 1
nonce = 1
data = input("Enter data : ")
difficultyLevel = []
timeTaken = []

while DIFFICULTY_LEVEL < 7:
    difficultyLevel.append(DIFFICULTY_LEVEL)
    startTime = time.time()
    while True:
        toBeHasedData = data + str(nonce)
        hash = hashlib.sha256(toBeHasedData.encode())
        digest = hash.hexdigest()
        count = 0
        for i in range(len(digest[:DIFFICULTY_LEVEL])):
            if digest[:DIFFICULTY_LEVEL][i] == '0':
                count += 1
        if count == DIFFICULTY_LEVEL:
            print("------------------------------------------------------------
--------------------")
            print("Difficulty Level: ", DIFFICULTY_LEVEL)
            print("Nonce: ", nonce)
            print("Digest: ", digest)
            break
        nonce += 1
    DIFFICULTY_LEVEL += 1
    endTime = time.time()
    timeTaken.append(endTime - startTime)

print("------------------------------------------------------------------------
--------")
plt.plot(difficultyLevel, timeTaken)
plt.xlabel("Difficulty Level")
plt.ylabel("Time Taken (seconds)")
plt.show()
```

```
D:\COLLEGE\LY_KJ_VII\BCT>D:/python/python.exe d:/COLLEGE/LY_KJ_VII/BCT/Exp3.py
Enter data : some data
--------------------------------------------------------------------------------
Difficulty Level:  1
Nonce:  23
Digest:  0df1e275c35748342d97a2f9dff153093f832eb40080705ce153a6eb41db073c
--------------------------------------------------------------------------------
Difficulty Level:  2
Nonce:  42
Digest:  00b02117f62c4d1f00c8a399667d18c7bea98dd9439a192a6852c40a2b17893c
--------------------------------------------------------------------------------
Difficulty Level:  3
Nonce:  7375
Digest:  000631e99efb86e01b541660640f0e6ddfc70efafe75d1d95447178693749e75
--------------------------------------------------------------------------------
Difficulty Level:  4
Nonce:  125656
Digest:  0000352613f686315df083c4596651184e6b01b98654ea2996ca6e22241c09ea
--------------------------------------------------------------------------------
Difficulty Level:  5
Nonce:  1792398
Digest:  00000cefd87a41fea61b9f51bb9799db71a8f981a97df0fbc905b31c8047fc96
--------------------------------------------------------------------------------
Difficulty Level:  6
Nonce:  5753072
Digest:  000000186d587352c6f715cf1f172bc9e7628a80e739685de652a5c7f02d79a8
--------------------------------------------------------------------------------
```



Proof of Stake

VOTING

```
from collections import Counter
import random

# VOTING
numberOfMiners = int(input("Enter number of nodes : "))
n = numberOfMiners
minerNodes = ['P' + str(i) for i in range(numberOfMiners)]
nodeNumbers = [i for i in range(numberOfMiners)]
maxIndex = -1
maximumVotes = -1
alreadyGotAChance = []
print("\n")
excluded = []
while numberOfMiners > 0:
    votes = [random.choices(nodeNumbers)[0] for i in range(n)]
    if maximumVotes >= 0:
        alreadyGotAChance.append(maximumVotes)
        for i in range(len(alreadyGotAChance)):
            votes[alreadyGotAChance[i]] = "X"
    print("Cycle: ", n - numberOfMiners + 1)
    print("-------------------------------------------------")
    print(f"Votes Given: {votes}")
    d = Counter(votes)
    d = dict(sorted(d.items(), key=lambda item: item[1], reverse=True))
    maximumVotes = next(iter(d))
    excluded.append(maximumVotes)
    maxIndex = votes.index(maximumVotes)
    if maximumVotes == "X":
        maximumVotes = list(d.keys())[1]
        print(maximumVotes)
        excluded.append(maximumVotes)
        maxIndex = votes.index(maximumVotes)
    print("Node:", maximumVotes, " got:", d[maximumVotes], "votes")
    print(f"Publisher Node: {maximumVotes}")
    print(f"Node {votes[maxIndex]} has mined a block and is published on
the blockchain")
    print("-------------------------------------------------\n\n")
    # votes.remove(maximumVotes)
    nodeNumbers.remove(maximumVotes)
    numberOfMiners -= 1
votes[maximumVotes] = "X"
print(votes)
```

```
Enter number of nodes : 5


Cycle:  1
-------------------------------------------------
Votes Given: [4, 2, 0, 1, 0]
Node: 0  got: 2 votes
Publisher Node: 0
Node 0 has mined a block and is published on the blockchain
-------------------------------------------------


Cycle:  2
-------------------------------------------------
Votes Given: ['X', 4, 4, 2, 1]
Node: 4  got: 2 votes
Publisher Node: 4
Node 4 has mined a block and is published on the blockchain
-------------------------------------------------


Cycle:  3
-------------------------------------------------
Votes Given: ['X', 2, 3, 2, 'X']
2
Node: 2  got: 2 votes
Publisher Node: 2
Node 2 has mined a block and is published on the blockchain
-------------------------------------------------


Cycle:  4
-------------------------------------------------
Votes Given: ['X', 1, 'X', 3, 'X']
1
Node: 1  got: 1 votes
Publisher Node: 1
Node 1 has mined a block and is published on the blockchain
-------------------------------------------------


Cycle:  5
-------------------------------------------------
Votes Given: ['X', 'X', 'X', 3, 'X']
3
Node: 3  got: 1 votes
Publisher Node: 3
Node 3 has mined a block and is published on the blockchain
-------------------------------------------------


['X', 'X', 'X', 'X', 'X']
```

COIN AGE

```
numberOfMiners = int(input("Enter number of nodes : "))
nodeNumbers = [i for i in range(numberOfMiners)]
n = numberOfMiners
print("\n\n")

coinAge = [random.randint(0, 20) for i in range(numberOfMiners)]
dummyCoinAge = coinAge.copy()
while numberOfMiners > 0:
    print("Cycle: ", n - numberOfMiners)
    print("-----------------------------------------------------------------
------------------")
    print(f"Coin Age: {coinAge}")
    maximumAge = max(dummyCoinAge)
    maxIndex = coinAge.index(maximumAge)

    print(f"Publisher Node: {maxIndex}")
    print(f"Node with value {maximumAge} has mined a block and is
published on the blockchain")
    print("-----------------------------------------------------------------
------------------\n\n")

    coinAge[maxIndex] = "X"
    for i in range(len(coinAge)):
        if coinAge[i] != "X":
            coinAge[i] += 1
    dummyCoinAge.remove(maximumAge)
    for i in range(len(dummyCoinAge)):
        dummyCoinAge[i] += 1
    numberOfMiners -= 1
```

```
Enter number of nodes : 5



Cycle:  0
--------------------------------------------------------------------------------
Coin Age: [10, 11, 8, 19, 19]
Publisher Node: 3
Node with value 19 has mined a block and is published on the blockchain
--------------------------------------------------------------------------------


Cycle:  1
--------------------------------------------------------------------------------
Coin Age: [11, 12, 9, 'X', 20]
Publisher Node: 4
Node with value 20 has mined a block and is published on the blockchain
--------------------------------------------------------------------------------


Cycle:  2
--------------------------------------------------------------------------------
Coin Age: [12, 13, 10, 'X', 'X']
Publisher Node: 1
Node with value 13 has mined a block and is published on the blockchain
--------------------------------------------------------------------------------


Cycle:  3
--------------------------------------------------------------------------------
Coin Age: [13, 'X', 11, 'X', 'X']
Publisher Node: 0
Node with value 13 has mined a block and is published on the blockchain
--------------------------------------------------------------------------------


Cycle:  4
--------------------------------------------------------------------------------
Coin Age: ['X', 'X', 12, 'X', 'X']
Publisher Node: 2
Node with value 12 has mined a block and is published on the blockchain
--------------------------------------------------------------------------------
```

Various options explored:

1) **Proof-of-Burn:** Proof-of-Burn is a new alternative to Proof-Of-Work. However, it is said to work on a similar platform as that of Proof-Of-Work. Here, instead of bringing the money together into computer equipment, the owner burns the coins. Here the coins to the address where they are irretrievable. By doing this, the owner gets a privilege to mine on the system. It works on random selection. To implement this system, the miners can burn wither the native currency, or they can burn the currency of a different chain

like the Bitcoin. As mentioned before, the burning of coins gets you the privilege. Thus, more is the number of coins your burn; more is the probability that you qualify the selection process.

2) **Proof-Of-Capacity:** Most of the alternative system present works on pay and play mechanism. In case of Proof-Of-Capacity, you pay with the hard drive space. The more is the hard drive space; the more is the probability of mining the next block and earning a reward. It sounds similar to Proof-Of-Stake, where the validator having more coins in the wallet gets the reward.
3) **Proof of Elapsed Time:** Clipmaker Intel is the company behind the development of Proof-Of-Elapsed Time. The functioning of this system is similar to Proof-Of-Work. However, it consumes lesser electricity. Unlike the Proof-Of-Work, you don't have to solve any cryptographic puzzle. In case of Proof-Of-Elapsed Time, you use Trusted Execution Environment or TEE to ensure a random looter production.
4) **Proof of authority:** In proof-of-authority blockchains, blocks are validated by approved accounts known as validators. The entire process is automated, which means that it does not require validators to be stuck sitting at their computers. But it does require that their nodes remain uncompromised at all times. To become a validator, a network user must earn the right by accumulating a sufficiently positive reputation. Thus, validators have an incentive to retain their position once they have earned it. Validators are motivated to uphold the transaction process, as failing to do so would result in a negative reputation being attached to their identities.

2. Explain your program logic, classes and methods used.

➔ **Program logic:**

Proof of work:

- Take input data from the user
- Iterate over all the preset difficulty levels from 1 to 6
- Now create a startTime variable to get the start time of the code
- Now keep changing the nonce and hash the concatenation of data and nonce until we get the required number of zeros at the start of the hashed value of the string
- The hashing algorithm used us SHA256
- Now once the required hash is found create an endTime variable
- Now store all the differences of start and end time and plot the graph between the difficulty level and the time taken to compute the required hash.

Proof of stake: CoinAge:

- Take the number of miners as input from the user
- Now create random coinAges numbers for all the miners
- Now select the maximum number from the coin ages and select that miner as the publisher node
- Now increment the values of all the others miners and repeat the above steps until all the miners become publisher nodes

Proof of stake: Voting:

- Take number of nodes as input from the user
- Now create random numbers as votes for other users
- No miner can vote for himself
- Now select the miner with maximum vote as the publisher node
- Now remove that node from further cycles
- Again take votes from the remaining miners and repeat the above process until all the miners get a chance to publish a node.

Methods Used:

- **Plot()**
- **Sha256()**
- **Encode()**
- **Hexdigest()**
- **Random()**
- **Randint()**
- **Time()**
- **Choices()**

3. Explain the Importance of the approach followed by you

➔ Proof of work and proof of stake consensus methods forms the basis of most of the cryptocurrencies. There are many alternatives which are available other than everything defined above but proof of work and proof of stake has its own advantages.

- A proof of work verification is difficult, costly, and time-consuming to create, but easy to verify. Bitcoin is secure because it is computationally infeasible to attack the network. Requiring Proof of Work for participation is central to this property. Hence Bitcoin relies on computational work on cryptographic challenges as the basis for trust.

- Proof of work (PoW) is necessary for security, which prevents fraud, which enables trust. This security ensures that independent data processors (miners) can’t lie about a transaction.

- PoW fundamentally is a system for authenticating transactions without the need for a third party, as well as for preventing individuals or organizations from tampering with the database.
- A specific advantage of PoW is that it depends on computational capabilities. So there is no need of putting anything into stake.
- Proof of stake offers key advantages compared to proof of work, experts say. Its faster transaction speeds and more efficient energy requirements allow for blockchains that are more scalable and thus easier to find more adoption among new users.
- On top of that, proof of stake provides opportunities to earn more crypto. You can lock up your coins in a liquidity pool and receive rewards in the form of more coins. This offers more opportunities to earn money and integrate into a financial system on a proof of stake network than on a proof of work network.
- Cost efficiency The main advantage of PoS is that miners do not need to invest increasing sums of money in more and more powerful computing equipment.
- PoS cuts the need for complex computations, which means more energy efficiency. The power to validate transactions is transferred to those with the most holdings of the network's native currency.

Conclusion:-

Understood how miners in blockchain add blocks to the blockchain and how they are rewarded based on the amount of work they do. The way by which they proof that the computation they have done is true is basically done by proof of work or proof of stake which are the methods employed to prove the work done by the miner.