

Batch: A2 Roll No.: 1911020

Experiment No. 3

**Title:** Implementation of PoS and PoW in block chain

**Objective:**

To understand and implement Proof of stake and Proof of work concepts in blockchain and to understand differences between them.

**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://repository.psau.edu.sa/jspui/retrieve/4eeae047-64d2-4d01-938c-82059aa39e0b/PoSvsPoW.pdf>
2. <https://www.investopedia.com/terms/p/proof-work.asp>
3. <https://www.investopedia.com/terms/p/proof-stake-pos.asp>
4. <https://www.geeksforgeeks.org/blockchain-proof-of-work-pow/>
5. <https://www.geeksforgeeks.org/difference-between-proof-of-work-pow-and-proof-of-stake-pos-in-blockchain/>
6. <https://www.geeksforgeeks.org/proof-of-stake-pos-in-blockchain/>

**Abstract: -**

Blockchain is a pioneering technology which has brought huge popularity for new virtual currencies where in transactions are recorded after being verified. The transactions are then verified by many clients or validators, to solve the reliability issue among several nodes implemented in digital currency's peer-to-peer network. Though there are different alternate consensus algorithms available, the most commonly implemented algorithm are Proof of Work (PoW) algorithm and the Proof of Stake (PoS) algorithm.

Proof of work (PoW) is a decentralized consensus mechanism that requires members of a network to expend effort solving an arbitrary mathematical puzzle to prevent anybody from gaming the system. Proof of work is used widely in cryptocurrency mining, for validating transactions and mining new tokens. Due to proof of work, Bitcoin and other cryptocurrency transactions can be processed peer-to-peer in a secure manner without the need for a trusted third party.

Proof-of-stake (POS), cryptocurrency owners validate block transactions based on the number of staked coins. Proof-of-stake (POS) was created as an alternative to Proof-of-work (POW), the original consensus mechanism used to validate a blockchain and add new blocks. While PoW mechanisms require miners to solve cryptographic puzzles, PoS mechanisms require validators to hold and stake tokens for the privilege of earning transaction fees. Proof-of-stake (POS) is seen as less risky regarding the potential for an attack on the network, as it structures compensation in a way that makes an attack less advantageous. The next block writer on the blockchain is selected at random, with higher odds being assigned to nodes with larger stake positions.

**Related Theory: -**

The blockchain is a resourceful invention by Satoshi Nakamoto where the information keeping in the shared database which are easily verifiable and not specified to any single location. It is a decentralized technology and there is no way for the hacker to corrupt the information in any transaction connected to the process of identity verification. Blockchain eliminates risks of data located centrally and has no single point of failure by various identical block across the network. Blockchain has been operable without failure since the invention of a cryptocurrency, Bitcoin, in 2008 on the basis of public and private “keys”. Satoshi Nakamoto has introduced proof of work (PoW) to build a distributed trust less consensus and resolve the double-spend problem. Blockchain technology is disrupting almost every industry for its improvement in efficiency and security. There are two primary algorithms, PoW (Proof-of-work) and PoS (Proof-of-stake) through which Blockchain operates and are required to decide whether to invest in a cryptocurrency using some decisive factors. Some significant

features to understand in Blockchain include speed, applications as well as the consensus algorithms. In this paper, we will compare to know about; PoW (Proof-of-work), POS (Proof-of-stake).

### **Proof of work:**

PoW: Proof-of-Work or PoW, is the original consensus algorithm in a Blockchain networks, where user sends a digital token to each other, verifies the transactions and create new blocks to the chain. In this algorithm, all miners or validators participate to validate and confirm the transactions carefully on the network to get rewarded. All the verified transactions in the network is collected into blocks by the distributed ledger and arranged accordingly. This process is called mining. Proof of work is a protocol that prevents cyber threat as in distributed denial-of-service attack (DDoS) which intends to drain computer resources by sending numerous false requests.

### **What is Proof of Work (PoW)**

The term “proof of work” was coined by Markus Jakobsson and Ari Juels during a document published in 1999. It is related to [bitcoin](#). Proof of Work (PoW) may be a protocol designed to form digital transactions secure without having to believe a 3rd party. This work builds on previous puzzle solutions. PoW may be a way of verifying current and past transactions. The work that goes into solving puzzle generates rewards for whoever solves it called it as mining. In other words, this is often an algorithm that’s designed to verify transactions and obtain new blocks added to [blockchain](#). With Proof of Work, miners are competing to be primary to finish a complex mathematical puzzle which will generate this new block, meaning that they’ll be ready to collect some new Bitcoins as a rewards.

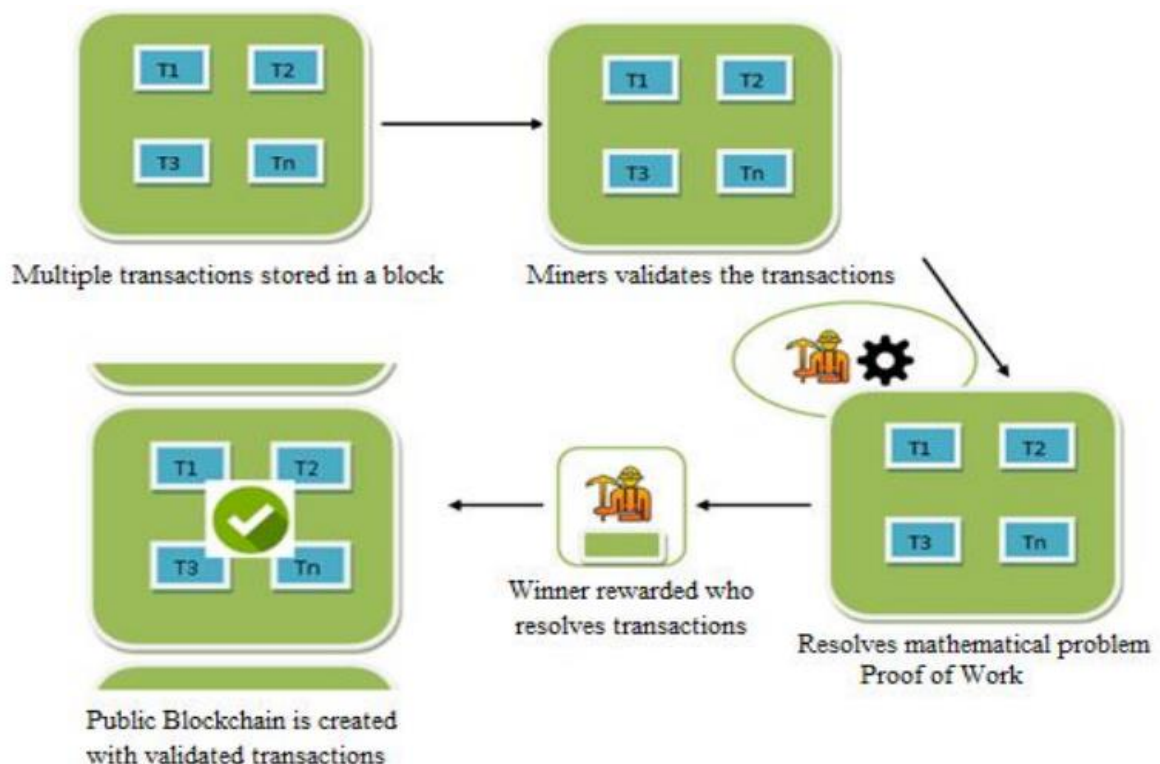
- PoW reduces risk of a 51% attack because it’s very hard to do work.
- No miner will be able to control bitcoin network single handedly Based on Hashcash PoW system.
- The miners need to give a proof that they have done some work, before proposing a new block.
- At the same time, each solution is easy for community to verify. This makes it easy to check all transactions for trustworthiness.
- PoW also sets a limit on how many new blocks of data can be generated. For example, miners can only create a Bitcoin (BTC) block every 10 minutes.
- It doesn’t rely on a single third party transactor. This builds a “trustless” and transparent network.
- Monopoly can increase over time.

### How does PoW work?

The miner or validator has to perform complex mathematical calculation to find digital coins. The successfully verified transactions are then stored in the new block and thus create a new group of blocks in blockchain, a public distributed ledger. There are two important aspects of mining; one is to check the validity of a transaction, and another is to create new cryptocurrency mined by the rewarded validators for their previous work. The miner will be rewarded with new cryptocurrency if they resolve the task first and this way, it interests new more miners. Furthermore, the mining process enhances the network computing power and computation to make a coin to escalate, which makes the mining process for the coin more difficult and expensive for the single miner.

Following occurrences happen while creating a transaction:

1. All transactions are stored in a block.
2. Miners validate the transactions in each block.
3. Miners/validators resolve mathematical problem called proof-of-work
4. The miner/validator is rewarded as first winner who resolves transaction running in each block.
5. Finally, the public blockchain is created with validated transactions.



Validation of transaction process

This mathematical computation or much CPU function is asymmetric and the work required is complex. Mining follows inverse hashing, where it finds number (nonce) so that the hash algorithm of block information is to be more lesser than the provided threshold (complexity). The threshold concludes the effort, computations, and energy required for mining to create a new block. That also makes the miner to be efficient for mining. This update occurs almost every 14 days, and a new block is generated in every 10 minutes. Miners put all the effort and get rewarded to make the node and blockchain more secure in the network. Proof-of work is more complex computational process to prevent modification of the old blocks in blockchain.

### 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.

### PoW algorithm:

The Proof of Work was invented by Markus Jacobson in 1999. All blockchain transactions are collected in groups called as mempool, where miners verify every transactions. Bitcoin users' requests transaction, which is then verified by the miner and add it to the next block using cryptographic hash value of the previous block. The hash value of the previous block is hidden for which miner has to keep trying a number after another. Once the miner finds the hash of the previous block, he declares it to the network to verify and create a new block. The first miner will be rewarded with bitcoin once he resolves this mathematical problem using massive computing power.

The miners have to resolve various problems with following characteristics:

- a) Asymmetric problems are tricky to decipher but the resolution is easily approved by the network.
- b) The puzzles with no skill involved, they require brute force, which needs massive computation energy.
- c) The parameters are timely updated; the mining will become more complex if it exceeds average block time.

## **Proof of stake**

PoS: Proof-of-stake is another consensus algorithm which possesses same motive as proof-of-work except the process to validate transactions in the distributed network. Proof-of-stake depends on its wealth called stake. A stake is a sum of currency locked up for some definite time period. Unlike proof-of-work, there is no reward of cryptocurrency unit for validating and confirming transaction within a block, instead, miners achieve transaction fees for the achieved task as reward. PoS works based on stake and emphasizes on number of cryptocurrencies in the blockchain to create new blocks rather than spending too many resources, energy or computational power as in PoW.

### **What is Proof of Stake (PoS) ?**

Proof-of-stake is a consensus algorithm that decides on who validate next block, according to how many coins you hold, instead of miners cracking cryptographic puzzles using computing power to verify transactions like they do with traditional Proof-of-Work.

- The probability of validating a new block is determined by how large a stake of a person.
- The validator do not receive a block reward, instead they collect network fees as their reward.
- Peercoin was first cryptocurrency to implement a full-scale PoS consensus model.
- Handling Monopoly and Power Consumption.

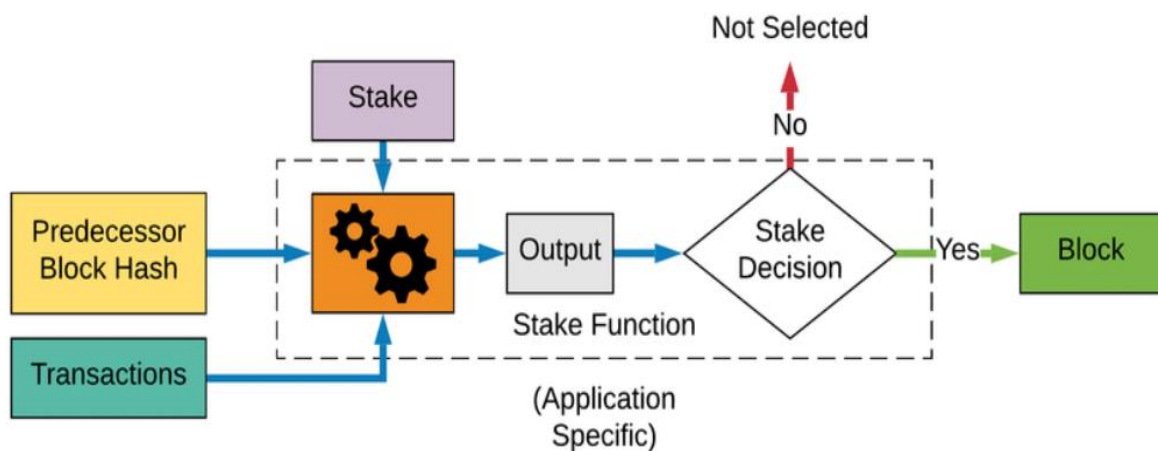
### **How does PoS work**

Proof-of-Stake is indirectly proportional to the network size and number of people staking the digital currency. If there is many people staking the coin then there will be fewer rewards. Furthermore, if the users have the possession of more cryptocurrencies for a longer period, achieve more transaction fees as reward, however, the process

should be shared in the network so that the control of the coin is prevented from one person. This concept works similarly as in banks fixed deposit wherein the customer earns more interest for keeping significant amount of money for a longer period. In PoS, the selection of a creator depends on the wealth, which is the number of coins or stake. Here, the user who validates the transaction and adds new block is called as forger. The forger puts their coins at stake to create a new block and validates the transactions to add a new block. They can also lose their stake and authority for further proceedings, if validator confirms any fraud transaction.

Here, it is required to select the forger to forge the next block. Following is the proper way to select the forger:

1. Randomized Selection- The user having the lowest hash value and the size of their stake will get the opportunity to select the next block.
2. Coin Age-Based Selection- The age of coin selects the forger.
3. Voting Based Selection - The user having the highest votes will get the opportunity to select the next block.



Proof-of-Stake flow.

### PoS algorithm:

PoS algorithm has a different way of processing than PoW. In this case, set of nodes stake their own digital coins for transaction confirmation. The staker can have a better opportunity to own the transaction validation if the amount and the deposit time of the stake are longer. In PoS, mining is not required as done in PoW, instead the digital currencies are already created in the network avoiding loss of computation power and complex work. Also, the validators can add blocks more frequently if they have more stakes in the blockchain. The participant or validator will be selected based on the



number of stake they possess. PoS implemented a technology called 'sharding'. It is the process of storing horizontal part of network in separate groups of nodes.

The limitation of PoS depends on its monopoly and 'Nothing at Stake'. Monopoly is the main disadvantage of PoS algorithms by the major stakeholders of the network. In case of 'Nothing at stake' there is more conflict occur if there are multiple unique chain in the blockchain, there can also have more forks which can create more confusion. Such problems does not occur in PoW algorithm.

#### **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.

PoW and PoS are the most recent and implemented blockchain consensus mechanism. PoW is strongly verified and implemented in various cryptocurrency schemes. The blockchain implies PoW algorithm can hardly encounters with DDoS attacks on any technologies. However, the huge energy consumption, expensive computational power, rising centralization, and small transaction throughput will make it difficult to adopt in the future. On the contrary, PoS system does not consumption of computing power is lesser and the reward depends on the amount and the duration of keeping the stake longer. The PoS algorithm presents a scalable blockchain with major transaction throughput, however, it is not much securer than the decentralized PoW algorithm.

The blockchain has been made more secure concerning attacks if there is rise in number of coins. The coin becomes more expensive preventing to take possession and buy huge amount of coins. In PoS, the protocol called Casper prevents from invalid transactions performed by staker. The protocol instantly removes the staker to validate the chain if found guilty and also stops for further staking. Though these consensus algorithms have a major role in cryptocurrency transaction, however, their differences in an approach becomes a controversial subject. The comparisons among consensus algorithms prove its competitive nature for adoption.

### Difference between Proof of Work (PoW) and Proof of Stake (PoS):







Proof of Work	Proof of Stake
 Computational work done by the miner	 Validating a new block is determined by how large a stake a person holds
 Reward is given to the first miner	 Collects network fees as their reward
 Network miners compete with one another, miner communities become more centralized over time	 Proof of stake systems are much more cost and energy efficient

Fig. 3 PoW Vs. PoS

Proof of Work (PoW)	Proof of Stake (PoS)
The probability of mining a block is determined by how much computational work is done by miner.	The probability of validating a new block is determined by how large of a stake a person holds (how many coins they possess).
A reward is given to first miner to solve cryptographic puzzle of each block.	The validator donot receive a block reward instead they collect network fee as their reward.
To add each block to chain, miners must compete to solve difficult puzzles using their computer process power	There is no competition as block creator is chosen by an algorithm based on user stake.
Hackers would need to have 51% of computation power to add malicious block.	Hackers would need to own 51% of all cryptocurrency on network, which is practically impossible.
Proof of work systems are less energy efficient and are less costly but more proven.	Proof of Stake systems are much more cost and energy efficient than POW systems but less proven.
Specialized equipment to optimize processing power.	Standard server grade unit is more than enough.
Initial investment to buy hardware.	Initial investment to buy stake and build reputation.
Bitcoin is most well-known crypto with a Proof-of-Work consensus building algorithm which uses most well-known proof-of-work function is called SHA256.	Some of cryptocurrencies that use different variants of proof-of-stake consensus are: EOS (EOS), Tezos (XTZ), Cardano (ADA), Cosmos (ATOM), Lisk (LSK).

**Implementation Details:**

**1. Enlist all the Steps followed and various options explored**

**Proof of work :**

**Steps:**

- The data and the difficulty limit are taken as input from the user.
- For each difficulty the nonce is calculated which gives the hash of the data and nonce combined with leading zeros equal to the difficulty level.
- The time taken for calculating nonce for the data for each difficulty is stored and plotted.

**Options explored:**

Random choice of nonce was explored but rejected due to fuzzy nature and different hash functions were explored for implementing Proof of work concept but SHA256 was chosen.

**Proof of stake :**

**Random Selection:**

**Steps:**

- The stakes and the number of cycles is taken as input from the user.
- For each cycle there would be as many rounds as the number of users and for each round for the remaining nodes random numbers are generated.
- The generated random numbers are multiplied by the stake of the corresponding node. The node with the maximum value after multiplication with stakes is chosen removed from further rounds in the cycle.
- The steps are repeated for every cycle and the nodes are put again after every cycle.

**Options explored:**

In case of a tie the first node chosen was explored but the node with the highest stake is chosen and implemented. Implementation for single cycle was explored but not used to simulate and depict randomness for the proof of stake concept.

**Coin Age:**

**Steps:**

- The time waited by each node and the number of rounds is taken as input from the user.
- For each round the node with the maximum wait time is chosen and then its wait time is set to 0.
- For the remaining nodes the wait times is incremented by one.
- The steps are repeated till all rounds are exhausted.

**Options explored:**

In case of a tie the first node chosen was explored and implemented. Implementation for user defined number of rounds was done.

**Voting:**

**Steps:**

- The stakes and the number of cycles is taken as input from the user.
- For each cycle there would be as many rounds as the number of users and for each round the nodes vote for any one of the remaining nodes except themselves.
- The node which receives the maximum votes is chosen and removed from further rounds in the cycle.
- The steps are repeated for every cycle and the nodes are put again after every cycle.

**Options explored:**

In case of a tie the first node chosen was explored but the node with the highest stake is chosen and implemented. Implementation for single cycle was explored but not used to simulate and depict randomness for the proof of stake concept.

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

### 1. Proof of Work:

#### Code:

```
# Proof of Work

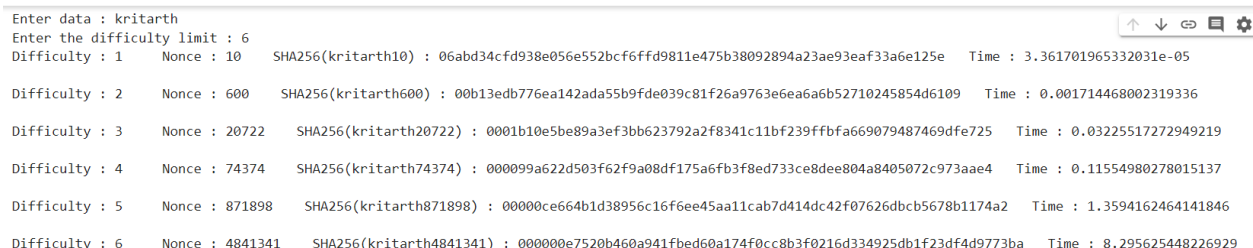
import time
import matplotlib.pyplot as plt
import hashlib

data = input("Enter data : ")
difficulty_limit = int(input("Enter the difficulty limit : "))
time_taken = []

for difficulty in range(1,difficulty_limit+1):
    start_time = time.time()
    nonce = 0
    while str(hashlib.sha256(f"{data}{nonce}".encode('utf-8')).hexdigest())[:difficulty]!="0"*difficulty:
        nonce+=1
    elapsed_time = float(time.time()-start_time)
    time_taken.append(elapsed_time)
    hash = hashlib.sha256(f"{data}{nonce}".encode('utf-8')).hexdigest()
    print(f"Difficulty : {difficulty}      Nonce : {nonce}      SHA256({data}{nonce}) : {hash}      Time : {elapsed_time}\n")

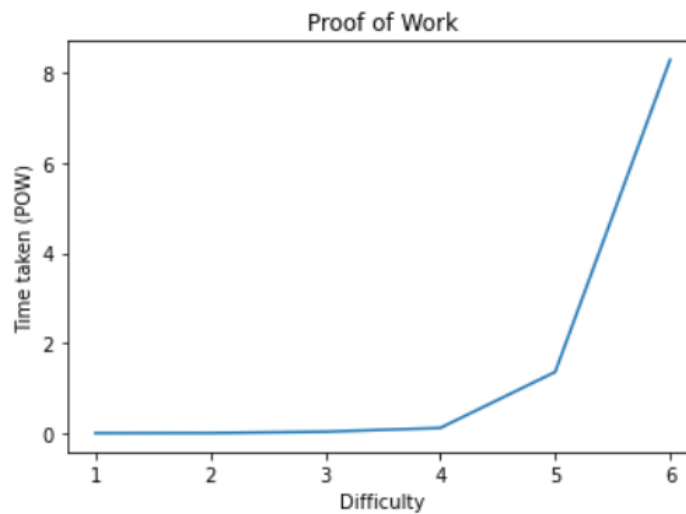
plt.title("Proof of Work")
plt.xlabel("Difficulty")
plt.ylabel("Time taken (POW)")
plt.plot([i for i in range(1,difficulty_limit+1)],time_taken)
plt.show()
```

#### Output:



The screenshot displays the output of the Python program. It shows the input data 'kritarth' and a difficulty limit of 6. For each difficulty level from 1 to 6, it prints the nonce, the SHA256 hash, and the time taken. The time taken increases significantly as the difficulty increases, especially at level 6 where it takes over 8 seconds.

Difficulty	Nonce	SHA256 Hash	Time
1	10	06abd34cfd938e056e552bcf6ff9811e475b38092894a23ae93eaf33a6e125e	3.361701965332031e-05
2	600	00b13edb776ea142ada55b9fde039c81f26a9763e6ea6a6b52710245854d6109	0.001714468002319336
3	20722	0001b10e5be89a3ef3bb623792a2f8341c11bf239ffbf6a669079487469dfe725	0.03225517272949219
4	74374	000099a622d503f62f9a08df175a6fb3f8ed733ce8dee804a8405072c973aae4	0.11554980278015137
5	871898	00000ce664b1d38956c16f6ee45aa11cab7d414dc42f07626dbcb5678b1174a2	1.3594162464141846
6	4841341	000000e7520b460a941fbcd60a174f0cc8b3f0216d334925db1f23df4d9773ba	8.295625448226929



## 2. Proof of Stake:

### a) Random Selection

#### Code:

```
# Proof of Stake

# Random
import random

cycles = int(input("Enter number of cycles : ").strip())
stakes = list(map(float, input("Enter the stakes of nodes : ").strip().split()))
n_nodes = len(stakes)

for c in range(1, cycles+1):
    print("\nCycle : ", c)
    left_nodes = [i for i in range(n_nodes)]

    for round in range(1, n_nodes+1):
        n = len(left_nodes)
        rand = [random.random() for i in left_nodes]
        v = [rand[index]*stakes[i] for index, i in enumerate(left_nodes)]

    chosen_node = left_nodes[v.index(max(v))]
    print(f"\nRound {(c-1)*n_nodes + round} :")
    print("Nodes : ", '\t'.join([f"P{i}" for i in left_nodes]))
    print("Random : ", '\t'.join([str(i) for i in rand]))
    print("Stakes*R : ", '\t'.join([str(i) for i in v]))
```

```
print(f"Chosen node : P{chosen_node}")
left_nodes.remove(chosen_node)
```

### Output:

Enter number of cycles : 2  
Enter the stakes of nodes : 10 15 20 25 30

Cycle : 1

Round 1 :

Nodes :	P0	P1	P2	P3	P4			
Random :	0.3460180982787263	0.1403534718676559	0.2274272444349338	0.6973623590821443	0.8093026767333271			
Stakes*R :	3.4601809827872634	2.1053020780148386	4.548544888869868	17.434058977053606	24.279080301999812			
Chosen node :	P4							

Round 2 :

Nodes :	P0	P1	P2	P3			
Random :	0.5773303764422224	0.960755259928835	0.4694179241413212	0.7361593439569227			
Stakes*R :	5.773303764422224	14.411328898932526	9.388358482826424	18.40398359892307			
Chosen node :	P3						

Round 3 :

Nodes :	P0	P1	P2			
Random :	0.9335577399959499	0.8033410671149276	0.2739060175398985			
Stakes*R :	9.335577399959499	12.050116006723913	5.478120350797971			
Chosen node :	P1					

Round 4 :

Nodes :	P0	P2			
Random :	0.788666497075014	0.12723675384239852			
Stakes*R :	7.88666497075014	2.5447350768479704			
Chosen node :	P0				

Round 5 :

Nodes :	P2			
Random :	0.9126045714943152			
Stakes*R :	18.252091429886303			
Chosen node :	P2			

Cycle : 2

Round 6 :

Nodes :	P0	P1	P2	P3	P4			
Random :	0.6618985682942415	0.5878921950883794	0.5529529722576688	0.4789495355174822	0.7289312820023468			
Stakes*R :	6.618985682942416	8.818382926325691	11.059059445153377	11.973738387937056	21.867938460070405			
Chosen node :	P4							

Round 7 :

Nodes :	P0	P1	P2	P3			
Random :	0.6803335849977955	0.6033867979154335	0.08990411772229157	0.4220115444608812			
Stakes*R :	6.803335849977955	9.050801968731502	1.7980823544458313	10.55028861152203			
Chosen node :	P3						



Round 8 :  
Nodes : P0 P1 P2  
Random : 0.19968542177873805 0.8695112134025031 0.9155868412373366  
Stakes\*R : 1.9968542177873805 13.042668201037547 18.31173682474673  
Chosen node : P2

Round 9 :  
Nodes : P0 P1  
Random : 0.5230150580062816 0.7896806239990175  
Stakes\*R : 5.230150580062816 11.845209359985262  
Chosen node : P1

Round 10 :  
Nodes : P0  
Random : 0.8586064459252931  
Stakes\*R : 8.586064459252931  
Chosen node : P0

## b) Coin Age

### Code:

```
# Proof of Stake

# Coin Age
import random

rounds = int(input("Enter number of rounds : ").strip())
time_waited = list(map(int, input("Enter the time waited for nodes : ").strip().split()))
n_nodes = len(time_waited)

for round in range(1, rounds+1):
    print(f"\nRound {round} :")
    print("Nodes : ", '\t'.join([f"P{i}" for i in range(n_nodes)]))
    print("Time_waited : ", '\t'.join([str(time_waited[i]) for i in range(n_nodes)]))
    chosen_node = time_waited.index(max(time_waited))
    time_waited = [i+1 for i in time_waited]
    print(f"Chosen node : P{chosen_node}")
    time_waited[chosen_node] = 0
```

### Output:

Department of Computer Engineering

Enter number of rounds : 10

Enter the time waited for nodes : 7 6 2 11 5

Round 1 :

Nodes :	P0	P1	P2	P3	P4
Time_waited :	7		6	2	11
Chosen node :	P3				

Round 2 :

Nodes :	P0	P1	P2	P3	P4
Time_waited :	8		7	3	0
Chosen node :	P0				

Round 3 :

Nodes :	P0	P1	P2	P3	P4
Time_waited :	0		8	4	1
Chosen node :	P1				

Round 4 :

Nodes :	P0	P1	P2	P3	P4
Time_waited :	1		0	5	2
Chosen node :	P4				

Round 5 :

Nodes :	P0	P1	P2	P3	P4
Time_waited :	2		1	6	3
Chosen node :	P2				

Round 6 :

Nodes :	P0	P1	P2	P3	P4
Time_waited :	3		2	0	4
Chosen node :	P3				

Round 7 :

Nodes :	P0	P1	P2	P3	P4
Time_waited :	4		3	1	0
Chosen node :	P0				

Round 8 :

Nodes :	P0	P1	P2	P3	P4
Time_waited :	0		4	2	1
Chosen node :	P1				

Round 9 :

Nodes :	P0	P1	P2	P3	P4
Time_waited :	1		0	3	2
Chosen node :	P4				

Round 10 :

Nodes :	P0	P1	P2	P3	P4	
Time_waited :	2		1	4	3	0
Chosen node :	P2					

### c) Voting

**Code:**

```
# Proof of stake

# voting
import random

cycles = int(input("Enter number of cycles : ").strip())
stakes = list(map(float, input("Enter the stakes of nodes : ").strip()
().split()))
n_nodes = len(stakes)

for c in range(1, cycles+1):
    print("\nCycle : ", c)
    left_nodes = [i for i in range(n_nodes)]

    for round in range(1, n_nodes+1):
        n = len(left_nodes)
        if n!=1:
            votes = [left_nodes[(random.randint(1, n-
1)+i)%n] for i in range(n)]
            chosen_node = max(set(votes), key = lambda x: votes.count(x)+
stakes[x]/100)

        else:
            votes = [left_nodes[0]]
            chosen_node = left_nodes[0]

    print(f"\nRound {(c-1)*n_nodes + round} :")
    print("Nodes : ", '\t'.join([f"P{i}" for i in left_nodes]))
    print("Votes : ", '\t'.join([str(i) for i in votes]))
    print(f"Chosen node : P{chosen_node}")
    left_nodes.remove(chosen_node)
```

**Output:**

```
Enter number of cycles : 2
Enter the stakes of nodes : 10 15 25 20 30
```

Cycle : 1

Round 1 :

```
Nodes : P0      P1      P2      P3      P4
Votes : 4       3       4       2       2
Chosen node : P4
```

Round 2 :

```
Nodes : P0      P1      P2      P3
Votes : 3       2       0       2
Chosen node : P2
```

Round 3 :

```
Nodes : P0      P1      P3
Votes : 1       0       1
Chosen node : P1
```

Round 4 :

```
Nodes : P0      P3
Votes : 3       0
Chosen node : P3
```

Round 5 :

```
Nodes : P0
Votes : 0
Chosen node : P0
```

Cycle : 2

Round 6 :

```
Nodes : P0      P1      P2      P3      P4
Votes : 1       3       3       4       0
Chosen node : P3
```

Round 7 :

```
Nodes : P0      P1      P2      P4
Votes : 2       0       1       0
Chosen node : P0
```

Round 8 :

```
Nodes : P1      P2      P4
Votes : 2       4       1
Chosen node : P4
```

Round 9 :  
Nodes : P1        P2  
Votes : 2        1  
Chosen node : P2

Round 10 :  
Nodes : P1  
Votes : 1  
Chosen node : P1

### 3. Explain the Importance of the approach followed by you.

The approach followed indicates the strength and weakness of both consensus algorithms proof of work and proof of stake. It indicates that the time taken by proof of work approach increases exponentially with increase in difficulty level. The proof of stake approach is linear with the number of nodes participating or having stake in the blockchain.

### Conclusion:-

Thus we have understood and implemented proof of work and proof of stake consensus algorithms and simulated it for toy examples. We have also understood the distinctions between the PoW and PoS approaches.