**Lesson Plan – Implementing Proof of Work**

**Learning objectives -**

1. Revise concepts of Nonce and Proof of Work from previous class
2. Implementing Proof Of Work

**Materials required -**

repl.it login credentials

**Proof of work (PoW)** : As we have already seen Proof of work (PoW) is a consensus mechanism used in blockchain networks to validate transactions and create new blocks.

**Here's a general overview of how PoW can be implemented in a blockchain:**

1. Each transaction in the blockchain network is broadcast to all nodes in the network.
2. Miners in the network collect these transactions and add them to a candidate block.
3. The miner then computes a cryptographic hash of the block header, which includes a hash of the previous block, a timestamp, and a nonce (a random value).
4. The miner then checks if the hash meets a certain difficulty requirement set by the network. This requirement is designed to ensure that the block creation process is slow enough to prevent malicious actors from taking control of the network.
5. If the hash meets the difficulty requirement, the miner broadcasts the block to the network.
6. Other nodes in the network receive the new block and verify that the hash meets the difficulty requirement and that the transactions are valid.
7. If the block is accepted by the network, the miner is rewarded with cryptocurrency.

**Diving Deeper into Proof-of-Work**

The blockchain participants always consider the longest chain to be the correct one. If someone is able to create the longest chain of blocks (even if the blocks are fake), the network is forced to accept the new chain.

The reason for this is simple — the blockchain network assumes that the longest chain has the most amount of computational work done in finding the Proof-of-Work for each block. Therefore, it is reasonable for the network to think that the longest chain contains the most proven record of transactions.

If a dishonest participant decides to tamper with a block, they would have to solve the Proof-of-Work for each subsequent block in order to introduce the tampered block into the network. This is computationally infeasible and almost impossible!

Furthermore, while the participant is busy finding the Proof-of-Work for each block, newer blocks are being added to the blockchain at a faster rate. The participant soon finds out that they are playing a losing battle against the entire network.

What is the key takeaway from all this? A block gets increasingly more tamper-proof as newer blocks are added next to it. Proof-of-Work makes it hard to get through the entire blockchain and allow someone to introduce a fake transaction.

The majority decision is represented by the longest chain, which has the greatest proof-of-work effort invested in it. – [Satoshi Nakamoto](https://nakamotoinstitute.org/bitcoin/)

## 

## **How do you calculate the longest chain?**

Here are the steps to calculate the longest chain in a blockchain network:

1. Identify the genesis block: The genesis block is the first block in the blockchain and has a unique identifier. You can usually find the genesis block by looking at the configuration or source code of the blockchain software.
2. Traverse the blockchain: Starting from the genesis block, traverse the blockchain by following the "previous block" pointer in each block. This will create a chain of blocks.
3. Calculate the cumulative work: For each block in the chain, calculate its cumulative work or proof-of-work. This value represents the amount of computational effort that was expended to mine the block. The cumulative work is usually computed using a hash function and a difficulty target.
4. Compare the chains: Compare the cumulative work of the chains that have been discovered. The longest chain is the chain with the most cumulative work. If multiple chains have the same cumulative work, the tie is usually broken by the chain with the most recent block.
5. Update the blockchain: Once the longest chain has been identified, update the blockchain to include any transactions or blocks that were not part of the longest chain. This is usually done by "reorganizing" the blockchain to follow the longest chain.

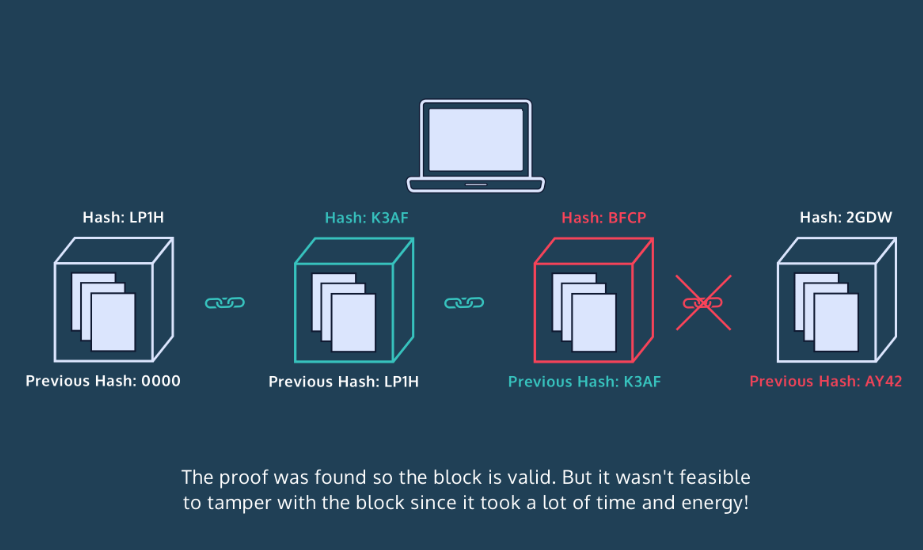
**Key Terms:**

* **Longest Chain:** The most trusted chain with the largest amount of computational work done in calculating the Proof-of-Work. In a blockchain network, the longest chain refers to the chain of blocks with the most cumulative work or proof-of-work. This is also known as the "heaviest chain" or "strongest chain"

## **Chainwork** is the total number of hashes that are expected to have been necessary to produce the current chain. Chainwork is also sometimes used as a measure of a miner's contribution to the network. The more computational work a miner contributes, the more likely they are to receive mining rewards.

– Pieter Wuille

To work out chainwork, you just need to work out how many hashes you would have needed to perform to mine each block in the chain, then add them up.



Now that we’ve seen a simple example of Proof-of-Work, let’s integrate it into our blockchain! Complete the proof\_of\_work() method inside the Blockchain class.

Introducing the lesson/project including the concepts (Time - 45 min)

Link to Repl it Project :

https://replit.com/@PriyankaJetLea1/Lesson10ImplementingProofOfWork#main.py

**Homework –**

# Write a program to find the sum of all items in a dictionary

<https://www.geeksforgeeks.org/python-program-to-find-the-sum-of-all-items-in-a-dictionary/>

Write a program to

* [Get Current Timestamp](https://pynative.com/python-timestamp/#h-get-current-timestamp) using
  + [Datetime to Timestamp](https://pynative.com/python-timestamp/#h-datetime-to-timestamp)
  + [Get Timestamp Using time Module](https://pynative.com/python-timestamp/#h-get-timestamp-using-time-module)
  + [Get Timestamp Using calendar Module](https://pynative.com/python-timestamp/#h-get-timestamp-using-calendar-module)

Also convert timestamp to string and datetime (format)

https://pynative.com/python-timestamp/