**Lesson Plan – Generating Block Hashes**

**Learning objectives -**

1. **Revise concepts of hashing and sha-256 from previous class**
2. **Block hash generation**

**Materials required -**

**repl.it login credentials**

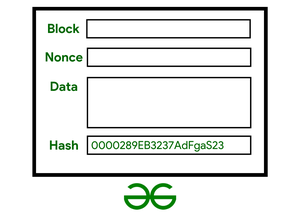
**What is Hash?**

A [hash](https://www.geeksforgeeks.org/hashing-set-3-open-addressing/)is a mathematical function that transforms an arbitrary length input into a fixed-length encrypted output. This consensus algorithm is a collection of rules that regulates the operation of a blockchain network. Aside from cryptocurrency, the most prevalent application of hash functions is password storage.

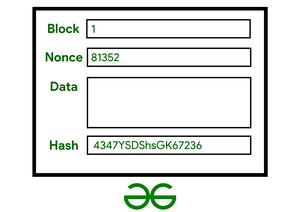
(*A consensus mechanism is any method used to achieve agreement, trust, and security across a decentralized computer network. In the context of blockchains and cryptocurrencies, proof-of-work (PoW) and proof-of-stake (PoS) are two of the most prevalent consensus mechanisms*.)

The cryptographic functions have the traditional functions along with some security traits, making them difficult to predict and determine the underlying content of the text or the transaction.

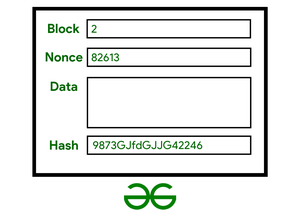
Consider the following picture, which depicts a collection of transaction data blocks.



*Figure 1. The Blockchain Transaction #1*



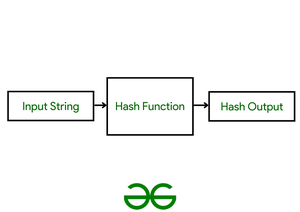
*Figure 2. The Blockchain Transaction #2*



*Figure 3. The Blockchain Transaction #3*

**How Do Block Hashes Actually Work?**

Putting in simple words, just take any length input string and then end up with a string of a fixed length through some work every time the hashing process needs to be done. That work is hashing. To understand it better, look at the diagram below:



*Figure 4. Understanding the Block Hash.*

**Example:**Let’s say, for example, there is a hashing algorithm that takes an input string and generates an output hash value.

**Input** **String:** Wow, this is a great Geeks for Geeks Tutorial

**Output:** tVP4UguDYLYf7BoyRPLMVpnuVGIMYJkmcn5KOnXmkwdxt8AGU5

**Note:**Even the slightest change in the structure of anything could have a huge impact on the output charset generated by the hashing block.

Even if there is a change in the input string like this-

**Input String:** wow, this is a great geeks for geeks tutorial

The output would then turn out to be-

**Output:** rVuSuWYq3oE1z0ROjBPjunQ7SJbMSPTgnj7slb2Uvo9Td4Tgay

**Properties Of Hash Blocks:**

**1. Property #1: The definiteness:**This means that no matter how many times a given input is parsed using a hash function, the result will always be the same. This is essential since it will be hard to keep track of the input if different hashes are obtained every time.

**2. Property #2: Easy yet Rapid Generation:**The hash function should be able to rapidly return the hash of input. If the procedure is not rapid enough, the system will be inefficient.

**3. Property #3: Former Image Resistance:**As seen in the example above, the generated hash should have no pre or former image resistance, even a small change should be able to create a different hash block, else it will be easy to decode the transactions, something which is not wanted.

**4. Property #4: Data Integrity Check:**The most typical use of hash functions is data integrity checking. It is used to compute checksums for data files. This program offers the user assurance that the data is correct. The integrity check assists the user in detecting any modifications to the original file. It does not, however, guarantee the originality of the work. Instead of changing file data, the attacker can update the entire file, compute a new hash, and deliver it to the recipient. This integrity testing program is only useful if the user is confident in the file’s authenticity.

**5. Property #5: Password Storage:**Password storage is protected using hash functions. Instead of saving passwords in clear text, most login procedures save password hash values to a file. The Password file is made up of a table of pairs in the form (user ID, h(P)).

**Difficulties Faced In Creating A Hash Block:**The difficulty of discovering a block is built into the blockchain, but it is also linked to the hash output’s four leading zeros. The difficulty here refers to the difficulty of obtaining a hash output that is less or more than the objective. Because any zeroes can be encoded (or hashed) in any number of ways possible.

As more individuals join the network, or rather as hashing power increases, the network’s complexity grows with time. It is, however, modified on a regular basis to guarantee that the block is mined within a certain time frame.

**Purpose of Hash Function:**Cryptographic hash functions, like speed bumps on a racing motorcycle, create obstacles for attackers. But it’s important to realize that the motorbike will ultimately make it down the street. These hurdles, however, will slow down the defenders—normal people and the server. If the speed bump is set too high, then there is a risk of irritating the user and overburdening the server. The continuous issue is to slow down attackers while balancing the users’ requirements and pleasure.

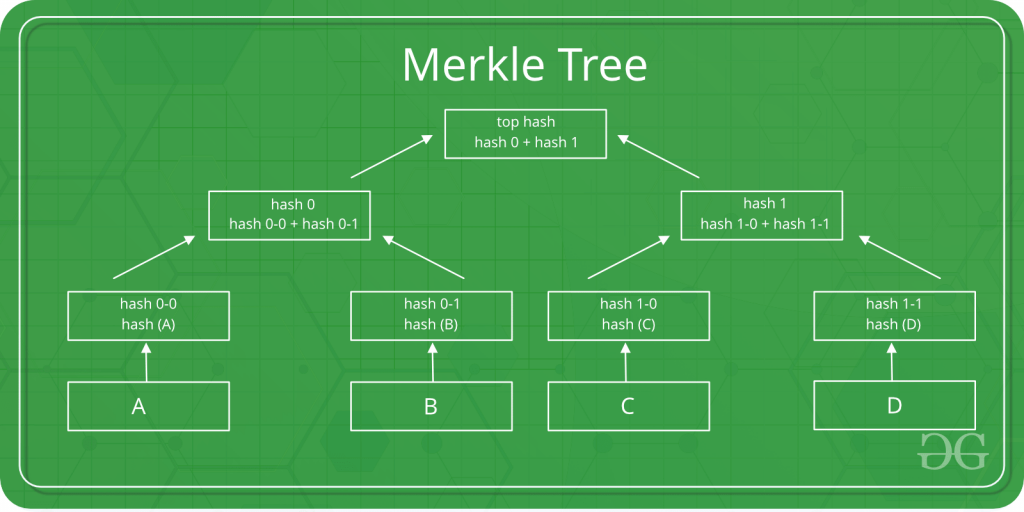
An adaptive one-way function is any function that is intended to iterate on its inner workings, feeding the output back as input, causing it to take longer to run in the end. It is adaptable since the developer may change the number of iterations that occur. Adaptive design has been used to design hash functions (such as PBKDF2) and encryption methods to secure saved passwords (such as [bcrypt](https://www.npmjs.com/package/bcrypt)).

[**Introduction to Merkle tree**](https://www.geeksforgeeks.org/introduction-to-merkle-tree/)

Merkle tree also known as hash tree is a data structure used for data verification and synchronization.

* Merkle trees are useful in distributed systems where same data should exist in multiple places.
* Merkle trees can be used to check inconsistencies.
* It is used in bitcoin and blockchain.

**Let’s look into one example :**



This is a binary merkel tree, the top hash is a hash of the entire tree -

* This structure of the tree allows efficient mapping of huge data and small changes made to the data can be easily identified.
* If we want to know where data change has occurred then we can check if data is consistent with root hash and we will not have to traverse the whole structure but only a small part of the structure.
* The root hash is used as the fingerprint for the entire data.
* The Merkle tree is useful because it allows users to verify a specific transaction without downloading the whole blockchain

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

Activity link:

[lesson 4](https://docs.google.com/document/d/1S6ZBM0CuXlqoFNNZgH_zxUTult8YMW7SeVnECbblk_0/edit?usp=sharing)

**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/