Merkle Trees and Start of Finance

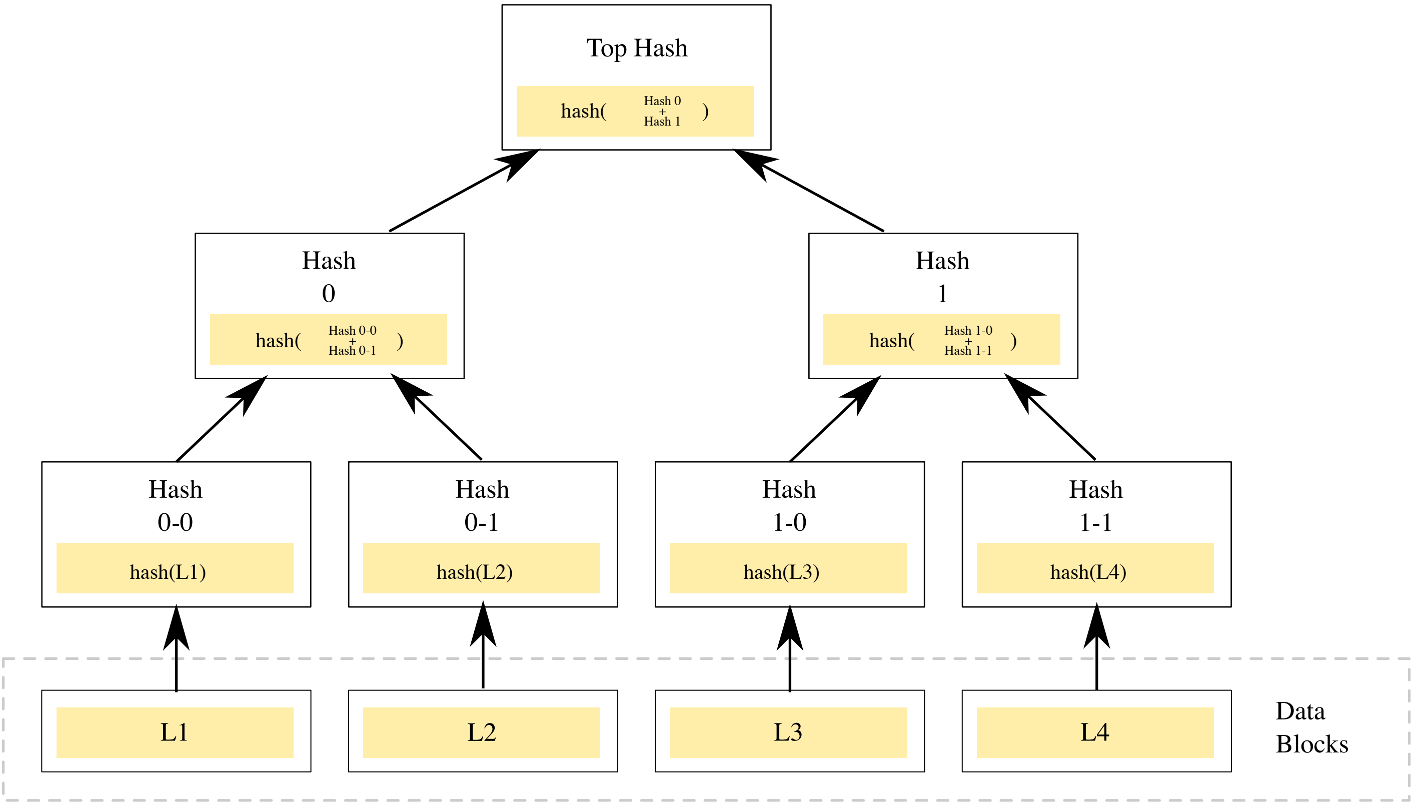
News

1. We have a number of successfully funded companies based on Crypto Offerings. “Pocket Full of Quarters” for example.
2. Kenya, Nigeria, Uganda and South Africa - A year ago they were starting to use “crypto” based coins. Today it is over 10% of all e-commerce in these conuntries.
3. Nairobi, South Africa, Malta, Reunion have implemented crypto-friendly laws.

Merkle Trees

Pseudo Code

1. Create a slice to hold the hashes of the leaves. Each leaf hash is a `[]byte` . So make the data type `[][]byte` . Make this slice of slice of byte then length of the data. That would be `len(data)` . Let’s call this `hTmp` .
2. For each data block
    1. Calculate a hash for the data block using `hash.HashOf()` .
    2. Save this in the slice created in (1) above.
3. Create a `[][]byte` slice to hold the intermediate hashes in the tree. This will need to be no more than `len(data)/2+1` in length. The plus 1 is so that 0 blocks of hashing or an odd number of blocks will have enough space. Let’s call this `hMid` .
4. Declare a variable `ln` , and set it to `len(data)/2+1`
5. While `ln >= 1` (Hint: the language only has `for` loops with lots of different ways of doing it)
    1. For each pair of hashes (if you have an odd number just use the single hash)
        - Calculate the hash of the pair using `hash.Keccak256()` . It takes a variable number of arguments so you can pass 1 or 2 arguments to it.
        - Append this to `hMid` .
    2. Replace hTmp with hMid
    3. Recalculate `ln` set it to `len(hTmp)/2`
    4. Generate a new empty hMid of allocated space of `len(hTmp)/2` .
6. Return `hTmp[0]`



Block Data	Hash
L1	21
L2	8
L3	10
L4	40
21+8	5
10+40	72
5+72	14

Finance

1. Stock
2. Bond
3. Yield
4. Pay out Ratio
5. Free Cash Flow
6. Risk