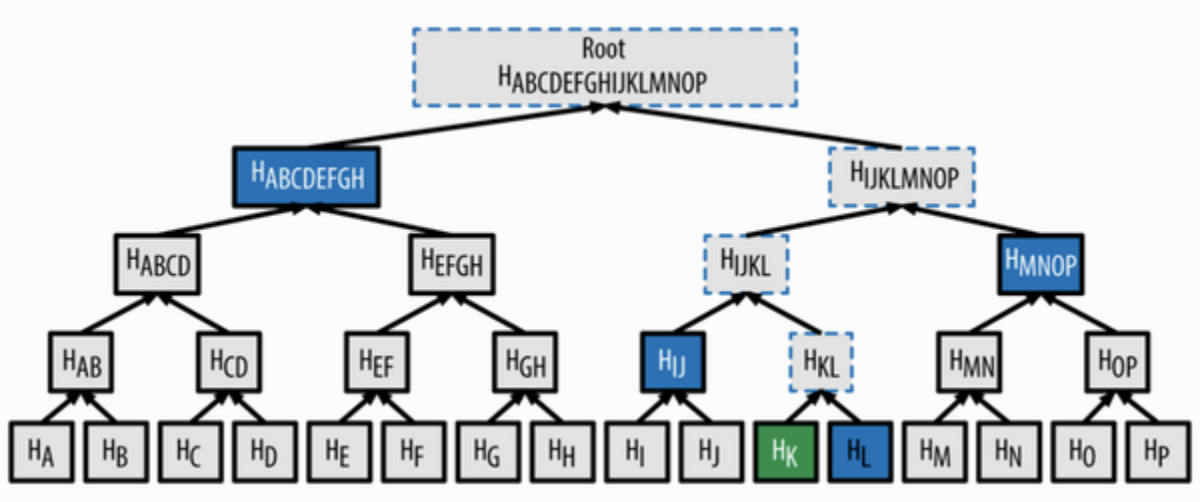
7. 블록체인

- 머클 트리


  -
  - 이진 해시 트리(binary hash tree)
    - leaf → root: 일반적인 트리 자료구조와 달리 거꾸로 연결되어 있다.
    - balanced tree: 비어있는 리프 노드를 가장 가까운 리프노드로부터 복사하여 채운다.

```
        ABCDEEEE .......Merkle root
       /        \
    ABCD         EEEE
   /    \       /
  AB     CD    EE .......E is paired with itself
 /  \   /  \   /
A   B  C   D  E .........Transactions
```

    -
    - 비트코인 개발자 가이드 - https://bitcoin.org/en/developer-guide#transaction-data
    - 리프노드를 2^n 개 만큼 미리 복사하는게 아니라, 해시 계산 시점에 짝이 없으면 자기 자신을 복사 하여 만듦
  - 머클 루트
    - 각 리프 노드로 부터 한 쌍씩 더블 SHA256 를 통해서 나온 결과값을 부모노드로 하여 채워 올라갔을 때 최상위 노드
  - 머클 경로
    - 존재여부를 확인하기 위한 해시(리프노드에 있음)로부터 머클 루트까지 계산하기 위하여 필요한 해시 값의 리스트
    - 위 그림에서는 Hk : Hl, Hij, Hmnop, Habcdefgh
  - 시간복잡도와 공간복잡도

Number of transactions	Approx. size of block	Path size (hashes)	Path size (bytes)
16 transactions	4 kilobytes	4 hashes	128 bytes
512 transactions	128 kilobytes	9 hashes	288 bytes
2048 transactions	512 kilobytes	11 hashes	352 bytes
65,535 transactions	16 megabytes	16 hashes	512 bytes

    -
    - 시간복잡도 O(logN): 특정 해시가 트리에 포함되어 있는지 확인하기 위해 필요한 해시 연산 수
      - 확인하고자 하는 거래가 해당 블록에 존재하는지 확인할 때, 머클트리를 이용하면 O(N) → O(logN) 으로 줄일 수 있다.
      - 실제로는 더블 SHA256을 하기 때문에 O(2logN)
    - 공간복잡도 O(logN): 특정 해시가 트리에 포함되어 있는지 확인하기 위해 필요한 해시 개수
      - 비트코인 네트워크 구조상 블록을 모두 가지고 있지 않는 노드(SPV노드)는 전체 데이터가 아닌 머클루트에 해당하는 데이터 만으로 검증할 수 있다.
- 블록
  - 블록 구조
    - 블록의 크기, 블록헤더, 거래

- 블록 헤더
  - 이전블록 해시 - 이전 블록의 해시값을 명시함으로써 directed linked list 처럼 블록의 자료구조를 형성한다.
    - 이전블록의 해시는 이전블록의 헤더를 해시한 값으로 만약, 이전블록이 변경되었다면 이 값도 바뀌게 되어 체인이 끊어진다.
  - 머클루트 - 블록에 포함된 거래들의 해시 값을 머클트리 자료구조를 이용하여 계산한 값. 거래의 검증에 쓰인다.
    - 거래변경 → 머클 루트 변경 → 블록헤더 변경 → 블록헤더 해시 변경 → 다음블록에서의 이전블록 해시 값과 불일치
  - 타임스탬프 - 블록의 생성시간 timestamp 또는 블록의 높이
  - 난이도목표 - 작업증명 알고리즘에 사용할 난이도, 특정 난스값의 변경에 따라 해시된 결과에서 앞의 0의 갯수
  - 난스 - 작업증명에 사용할 값. 난스값이 바뀌면 헤더의 해시값도 변경됨. 특정 난스를 찾는게 작업증명 알고리즘.
- 최초 블록

```
getblockhash 0

000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f

getblock 000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f

{
  "hash": "000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f",
  "confirmations": 494126,
  "strippedsize": 285,
  "size": 285,
  "weight": 1140,
  "height": 0,
  "version": 1,
  "versionHex": "00000001",
  "merkleroot": "4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b",
  "tx": [
    "4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b"
  ],
  "time": 1231006505,
  "mediantime": 1231006505,
  "nonce": 2083236893,
  "bits": "1d00ffff",
  "difficulty": 1,
  "chainwork": "0000000000000000000000000000000000000000000000000000000100010001",
  "nextblockhash": "00000000839a8e6886ab5951d76f411475428afc90947ee320161bbf18eb6048"
}
```

-

```
getblockhash 1

00000000839a8e6886ab5951d76f411475428afc90947ee320161bbf18eb6048

getblock 00000000839a8e6886ab5951d76f411475428afc90947ee320161bbf18eb6048

{
  "hash": "00000000839a8e6886ab5951d76f411475428afc90947ee320161bbf18eb6048",
  "confirmations": 494125,
  "strippedsize": 215,
  "size": 215,
  "weight": 860,
  "height": 1,
  "version": 1,
  "versionHex": "00000001",
  "merkleroot": "0e3e2357e806b6cdb1f70b54c3a3a17b6714ee1f0e68bebb44a74b1efd512098",
  "tx": [
    "0e3e2357e806b6cdb1f70b54c3a3a17b6714ee1f0e68bebb44a74b1efd512098"
  ],
  "time": 1231469665,
  "mediantime": 1231469665,
  "nonce": 2573394689,
  "bits": "1d00ffff",
  "difficulty": 1,
  "chainwork": "0000000000000000000000000000000000000000000000000000000200020002",
  "previousblockhash": "000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f",
  "nextblockhash": "000000006a625f06636b8bb6ac7b960a8d03705d1ace08b1a19da3fdcc99ddbd"
}
```