Merkle Tree

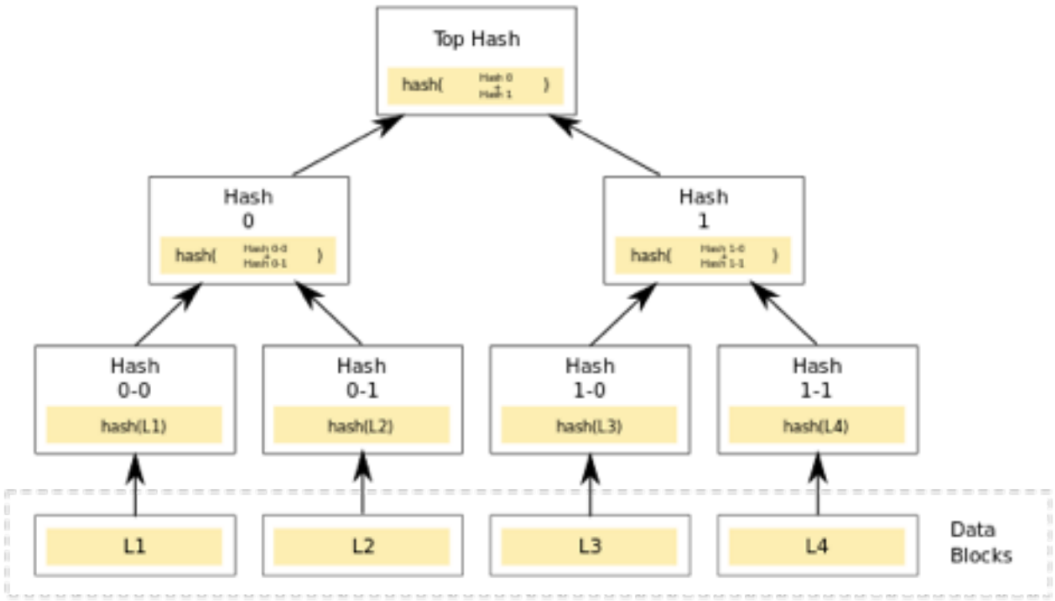
Un Merkle Tree est une structure relativement simple permettant de vérifier l'intégrité de données.

C’est un arbre binaire ou chaque noeud contient un hash cryptographique de ses deux enfants:

$H_{parent} = hash(H_{gauche} + H_{droite})$

Si un noeud n’a qu’un seul enfant, son hash est le même que celui de son enfant.

La hash de la racine de l’arbre s’appelle le *Merkle root*. Les feuilles (*leaves*) de l’arbre contiennent les hashes de blocs de données.



Partie 1 - Implémentation d'un Merkle tree

En utilisant Node.js ou Go, écrire un module permettant de créer un Merkle tree à partir d'une série de données (dans l'illustration ce serait les données L1, L2, L3, et L4).

Par exemple:

```
createMerkleTree([string1, string2, string3, ...]) => MerkleTree
```

Le Merkle tree créé doit avoir les fonctions suivantes:

```
// Retourne le hash à la racine de l'arbre MerkleTree#root()

// Retourne le nombre de niveau de l'arbre
MerkleTree#height()

// Retourne un Array contenant les hashs du niveau spécifié
MerkleTree#level(index)
```

Les hashs sont calculés en utilisant la fonction cryptographique SHA256 (voir le package crypto de node.js).

Bonus Points: écrire une suite de test

Partie 2 - Questions

1) Dans l'illustration, imaginons que je possède le Merkle tree. Quelqu'un me donne le bloc de données L2 mais je ne lui fais pas confiance. Comment puis-je vérifier si les données de L2 sont valides?

2) Je possède le bloc L3 et un Merkle root. Par contre, je ne possède pas les autres blocs ni le Merkle tree. Quelles informations dois-je obtenir au minimum pour m'assurer que le bloc L3 fait bien partie du Merkle tree qui a pour racine le Merkle root que je possède?

3) Quelles sont des exemples d'application pour un Merkle tree?