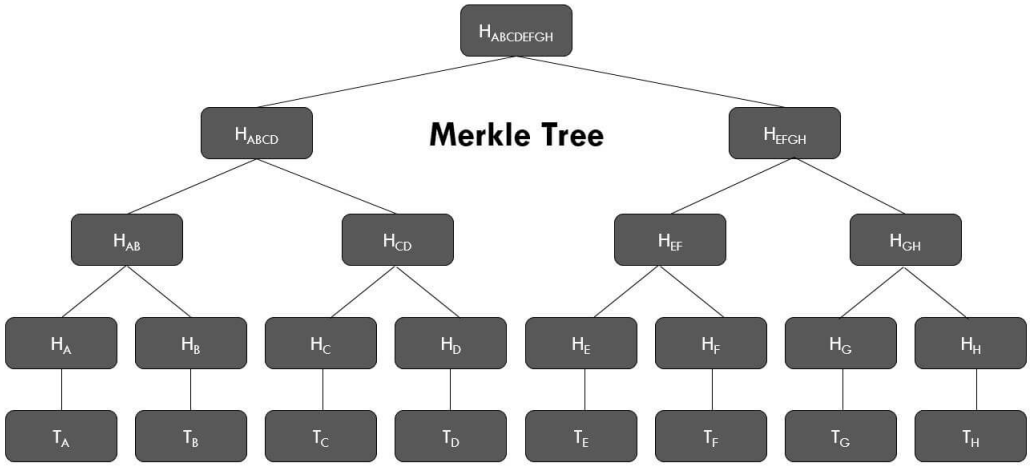
Homework 31, due November 22

(1) We said that the proof-of-work in Bitcoin is to find a nonce n such that

$$H(\text{transactions}, n) < \text{target},$$

where "transactions" are all the transactions in the proposed block. For efficiency reasons, Bitcoin does not actually hash all the transactions every time a new nonce is tried. Rather, the proposed nonce is hashed with the *Merkle root* of a *Merkle tree*.

(a) This picture shows the basic idea of a Merkle tree:



The transactions are T_A, T_B, etc at the very bottom. Transactions are hashed ($H_A = H(T_A), H_B = H(T_B)$, etc), and then pairs of hashes are hashed together ($H_{AB} = H(H_A, H_B)$ etc) until we obtain the *Merkle root* at the top of the tree (this is the $H_{ABCDEFGH}$ node at the top). Explain why the Merkle root will likely be completely different if even one transaction is changed.

(2) Suppose Alice knows the entire Merkle tree shown above, but Bob only knows the Merkle root. She reveals T_A to Bob and wants to prove T_A is in the Merkle tree, but without revealing any other transactions. How can she do this? (Hint: See what happens if there are only two transactions in the tree, T_A and T_B. Alice can reveal T_A and H_B. Bob can hash T_A to get H_A, and then Bob can hash H_A and H_B to get the Merkle root H_{AB}. Now generalize to the bigger tree. Make sure to explain why this procedure doesn't give Bob any information about the other transactions, as long as a good hash function is used.)

(3) Some cryptocurrencies, like Ethereum, do not use proof-of-work to achieve blockchain consensus. Instead, they use a different mechanism called proof-of-stake. Read about proof-of-stake, and briefly describe potential strengths or weaknesses of proof-of-stake compared to proof-of-work.