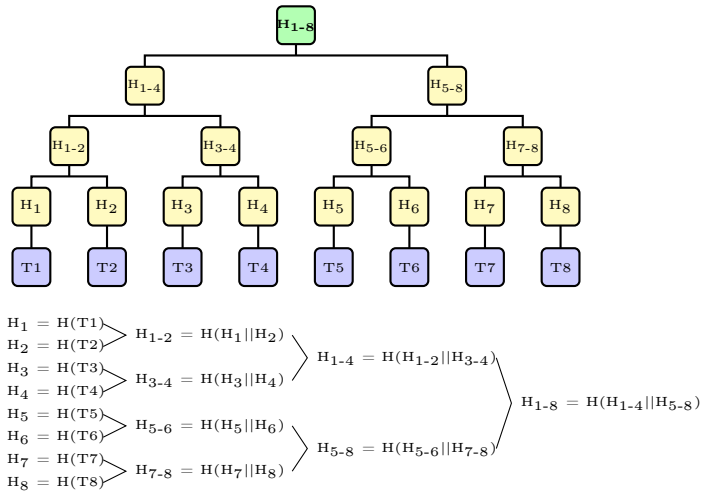
Programming III

Semester Project 2

1. The Task

A Merkle tree is a binary tree in which every leaf node is labelled with the cryptographic hash of a transaction. They provide a kind of summing up of the transactions contained in a block. For instance:



We assume a blockchain structure in which each full node has a copy of all transactions and a copy of the root of the corresponding Merkle tree (the blue and green nodes) and each *thin node* has but a copy of the root of the Merkle tree (the green node.) A thin node may want to know whether a transaction, identified by a given number, is in the block or not. Since the root of the tree is not falsifiable because of the properties of the hash functions, it does not want the full node to merely answer "yes" of "no", but it wants to verify it by recalculating the path up toward the root of the tree.

For that, it needs that the full node sends it the missing elements. Of course, the full node may send it the whole bunch of hashes. But this would be a lot of traffic. There are more efficient ways to solve this problem.

Your task is:

1. Design and implement a full node. You have to choose the data structures you consider more appropriate to solving this problem. Notice that the full node is only asked to hold the transactions and the root of the tree; you may choose whether to calculate the intermediate nodes or to have then pre-computed in memory.

2. Design an efficient algorithm so that the full node answers a hypothetical request of a thin node by sending the hashes that are necessary to compute the path to the root and only these hashes. The full node receives the number of the transaction.

3. Implement your algorithm in Java. You are expected to deliver the source files and a short report detailing the strategy of the algorithm and explaining the problem you are solving in detail. Since this is just a prototype, you may use the Java *hashCode* function as a hash function. This is not what you would do in real life, though.

2. Submission and Acceptance

The Semester Project must be submitted through the WebCampus groups. No other delivery will be accepted.

- The final implementation version together with the report must be delivered by 6th November. No delivery will be accepted after this date.
- If corrections are required, the new delivery date is 13th November. This is the definitive date.
- Defence of the Semester Project will take place on 13th November and 20th November. No defence is possible if the Semester Project has not been accepted.

Please notice that after the first delivery only one additional delivery can be made if the work was not approved, assuming the first one was submitted in due time. If the work was not delivered on the first date, there is no possibility of from another submission. The approval of the Semester Project implies the approval of both the delivery and the oral exposition.