# 14.04.2025 Week 9 exercises: Distributed hash tables and consistency models

## Exercise 1:

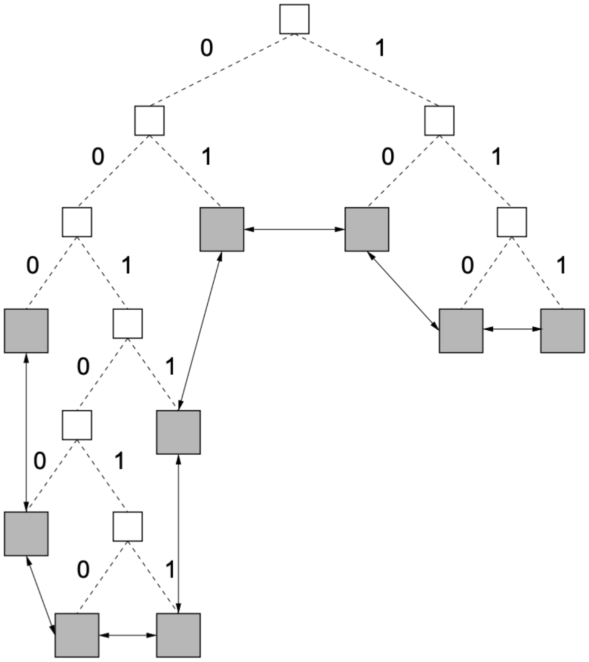
In Pastry, each node maintains 3 tables: (1) routing table, (2) leaf set, and (3) neighborhood set. The data items and nodes have unique 128-bit IDs, and are treated of as sequences of digits in base $2^4$.

Explain the routing procedure in Pastry if node $63AB$ wants to retrieve the value of key $EB3E$. Give an example of one possible routing path.

## Exercise 2:

**Background:** The figure below depicts a *Prefix Hash Tree*. Each vertex has either 0 or 2 children. The left child is reached by following the edge labelled 0, and the right child is reached by following the edge labelled 1. The white vertices are the inner vertices and the grey vertices are the leaf vertices. The leaf vertices are connected to their left and right siblings through special pointers. The path string P to a vertex is the string formed by the labels of the edges encountered on the path from the root to this vertex, in order.

The leaf vertices can store PHT key-value pairs. The keys are D-length strings consisting of 0s and 1s. **A key K can only be stored on the leaf-vertices for which P is a prefix of K.** The table on the right shows some keys that can be stored on leaf vertices with the given P. Moreover, the PHT automatically balances itself by creating, removing and merging vertices.



| Leaf nodes | Keys |
|---|---|
| 000* | 000001<br>000100<br>000100 |
| 00100* | 001001 |
| 001010* | 001010<br>001010<br>001010 |
| 001011* | 001011<br>001011 |
| 0011* | |
| 01* | 010000<br>010101 |
| 10* | 100010<br>101011<br>101111 |
| 110* | 110000<br>110010<br>110011<br>110110 |
| 111* | 111000<br>111010 |

©2004 Ramabhadran et al.

1. The vertices of the prefix hash tree need to be stored on a distributed hash table comprising of several peers, *i.e.*, PHT vertices need to be mapped into DHT nodes. Describe what will be the `DHT-key` and `value` to be stored on the DHT. How will you decide which `DHT-key` will be assigned to which peer? Note that this `DHT-key` is different from the `PHT-key` stored on the leaf-vertices of the prefix hash tree.

2. The prefix hash tree is now stored on a DHT. A key (`K`) stored on the prefix hash tree needs to be retrieved. Give the pseudocode of Prefix-Hash-Tree-Lookup(PHT-Key) for retrieving the value corresponding to `K` on the prefix hash tree over DHT. Can you find optimization opportunities in the algorithm? You may assume the implementation of DHTLookup(DHT-key) is already provided. **Hint:** Think of a simple linear algorithm.

3. Now that you are able to fetch PHT-keys from the prefix hash tree, it is time to make use of this combination of PHT and DHT to do something cool: range queries. Range queries return the values of all the available PHT-keys $K_i$, given a range $\mathtt{L} \leq \mathtt{R}$ such that $\mathtt{L} \leq K_i \leq \mathtt{R}$. For example, in the figure above, the range 000000 – 000100 should return the keys 000001, 000100, and 000100. Taking advantage of how the PHT-keys are stored on the prefix hash tree, and the special pointers mentioned before, sketch the pseudocode of PHT-Range-Query(L,R).

**Exercise 3:**

Consider a baseball game where the data (score) is read and/or written by the following participants:

1. **Official Scorekeeper**: Maintains the official score. Writes to the persistent key-value store.

2. **Umpire**: Officiates a baseball game from behind home plate. The umpire, for the most part, does not actually care about the current score of the game. The one exception comes after the top half of the 9th inning, that is, after the visiting team has batted and the home team is about to bat. Since this is the last inning (and a team cannot score negative runs), the home team has already won if they are ahead in the score; thus, the home team can and does skip its last at bat in some games.

3. **Radio reporter**: Periodically announce the scores of games that are in progress or have completed.

4. **Sportswriter**: Watches the game and later writes an article that appears in the morning paper or that is posted on some website.

5. **Statistician**: The team statistician is responsible for keeping track of the season-long statistics for the team and for individual players.

6. **Stat Watcher**: A fan inquiring about the total number of runs scored by his team this season.

Based on the following definitions, associate these consistency guarantees with each of the participants above.

(a) **Strong Consistency** See all previous writes.

(b) **Eventual Consistency** See subset of previous writes.

(c) **Consistent Prefix** See initial sequence of writes.

(d) **Bounded Staleness** Guarantee on reading all writes that are older than a certain age.

(e) **Monotonic Reads** See increasing subset of writes.

(f) **Read My Writes** See all writes performed by reader.