

## Peer-to-peer networks – (due till April 22, 2009)

### Exercise 6.1: Fingers in Chord

In Chord, a node forwards a request to a neighbor which was addressed by what we called the  $i$ -th finger. Implement the function (in java, C or pseudocode) with the signature given below. It should return the best fitting neighbor (don't overshoot the mark, we can not go back easily) from the list `successor []` who is best suited for the key `hash_value`. You can assume that our hashes fit into a long Integer.

```
long returnFinger(long successor[], int no_entries, long hash_value);
```

The ID of a node can be accessed with

```
node.ID
```

## Peer-to-peer networks

### Exercise 6.2: Arrival of a new node in Chord

- a) In the lecture, we learned how the Chord protocol manages the key space and how routing is done. We also saw a brief outline of the insertion of a new node. The text

chord.pdf

on our homepage gives further details on how the insertion is actually implemented (read sections „Node arrival“ and „Stabilization Protocol“). Describe the soft-state approach adopted by the designers of Chord.

- b) Can you think of advantages and disadvantages of the soft-state solution?
- c) Find a more deterministic solution which does not build on top of a random choice.