

Practice Problems for Homework 4

1. You are to select data structures to implement a caller-id system (for all US phone numbers). You can find a good solution just using the data structures that we have covered in class.

Your implementation is to be designed for the following conditions. There are 10^8 phone numbers that will be inserted with 10^5 different area code/exchange (first 6 digits of the number) combinations occurring. Further, suppose you only have space in main memory (a.k.a. RAM) to hold 8,000,000 bytes.

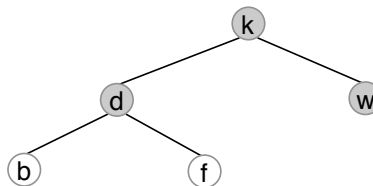
I've added these constraints on the memory to make this a more interesting practice problem. The key facts you need to know about this are:

- All information stored on the disk (i.e., not in main memory) is grouped into pages which are brought into RAM as a whole. To keep it simple, assume you can hold 1,000,000 bytes on a page and in constant time a page can be read from disk (or written to disk) when given its location (a long integer).
- An integer takes 4 bytes (32 bits) – the phone number can be stored as an integer
- A long takes 8 bytes – when you need to store something on disk you can represent the location of a page using a long integer.

You need to support the following methods. All must run in expected constant time.

- Insert a new area code/exchange combination into the system. (This will be used infrequently).
- Insert into the system a new item that consists of a phone number and name. (You can assume that the area code/exchange from the number corresponds to one of the area code/exchanges already in the system.)
- Given a phone number, return the name.
- Given a phone number, remove the corresponding item from the system.

2. Below is a red-black tree in which the filled nodes are black and the unfilled nodes are red.



- What is the black height of this tree?
- Show the red-black tree that results that results when a is added.
- Show the red-black tree that results that results when c is added.
- Show the red-black tree that results that results when a second a is added. In the binary search tree insertion, the solutions will take the right branch when an equivalent element is encountered.
- Show the red-black tree that results that results when a second b is added. In the binary search tree insertion, the solutions will take the right branch when an equivalent element is encountered.
- Show the red-black tree that results that results c is removed.