COMS21103: Problem set 5

2015/2016

Remark: Most problem sets for the next few weeks will contain at least one starred problem, which is more challenging. If any of the problems seem unclear, please post a question on the Blackboard discussion board.

1. Using the analysis from the Bloom filters lecture, if we want to use a Bloom filter for a Web cache which stores up to 1,000 URLs picked from a universe of 10,000,000 URLs, with a 0.1% probability of false positives, how many bits of storage should we use for the Bloom filter?

2. Imagine the sequence of integers $5, 4, 11, 15, 9, 10, 3, 12, 6, 13$ are inserted into a skip list (in that order). By tossing a (real) fair coin, simulate the operations involved. Draw a diagram of the resulting skip list.

3. Imagine that you have a skip list data structure, as described in the lecture, except that your coin-flipper is broken! What is the worst case time complexity of the operations INSERT, DELETE and FIND if the coin-flips are fixed to be,

   $$\text{HTHTHT}\ldots \text{(the first coin flip is always a heads)}$$

4. Instead imagine that your coin-flipper is broken so that the coin-flips are always,

   $$\text{HHTTHHTTHHTT}\ldots \text{(the first coin flip is always a heads)}$$

   How many levels will there be in the tree? What is the worst case time complexity of the operations INSERT, DELETE and FIND now?

5. Prove the claim made in the lecture on self-balancing trees that the height of a 2-3-4 tree is always $O(\log n)$. You may use the fact that a 2-3-4 tree is always perfectly balanced when proving this.

6. ($\star$) Explain how to augment the 2-3-4 tree data structure discussed in the lecture on self-balancing trees to support the PREDECESSOR and RANGEFIND operations. Descriptions of these operations are given at the start of said lecture. The time complexity of the PREDECESSOR operation should be $O(\log n)$. The time complexity of the RANGEFIND operation should be $O(\log n + m)$ where m is the number of elements returned. That is it should be *output sensitive*.

7. ($\star$) Explain how to augment the skip list data structure discussed in the lecture to support the PREDECESSOR and RANGEFIND operations. Descriptions of these operations are given at the start of said lecture. The time complexity of the PREDECESSOR operation should be $O(\log n)$. The time complexity of the RANGEFIND operation should be $O(\log n + m)$ where m is the number of elements returned. That is it should be *output sensitive*.