CS2040S: Data Structures and Algorithms

Discussion Group Problems for Week 6

For: Sept. 19, 2019

Problem 1. B-Tree

Right now we are given some invariants of a B-Tree. Namely:

1. All leaf nodes are of the same depth.
2. All non-root nodes have between $b-1$ to $2b-1$ keys.
3. The root has at most $2b-1$ keys.
4. An internal node with k nodes must have $k+1$ subtrees.

Problem 1.a. Show that as long as $b \geq 2$, the **height of an n-key B-tree is at most** $O(\log(n))$. You are also allowed to use that $\log_p(n)$ and $\log_q(n)$ differ by a constant factor so long as $p, q > 1$.
Hint: *At the very least, how many keys must each node have? Relate this to the height somehow.*

Problem 1.b. Discuss about how the invariants of a B-Tree are maintained after an **insertion** operation.

Problem 1.c. Discuss about how invariants of a B-Tree are maintained after a **deletion** operation.

Problem 2. (Optional) Choosing a good b

Right now, we have shown that B-Trees are asymptotically optimal. But we also actually have a parameter b (the amount of keys that each node can hold up to). Remember, a node can have at most $2b-1$ keys.

Problem 2.a. If someone felt like implementing a B-Tree for their filesystem/database, on what factor should they base their choice of b on?

Hint: *Notice that regardless of your choice of $b \geq 2$, the operations are asymptotically the same. So just for this week, we are asking for good value of b that optimises for the constants.*

Problem 2.b. For your previous choice of b, let's say that one key corresponded to one person each. How many levels might a B-tree need to have before it has one key for each human on Earth? (*Assume that the current world population is 7.5 billion.*)