CMPS 2200 Introduction to Algorithms – Fall 15

9/16/15

3. Homework

Due **9/24/15** at the beginning of the lab

1. **1-2-3-4-5 (4 points)**
   Describe all valid red-black trees that store the numbers 1, 2, 3, 4, 5.

2. **B-tree-search using binary search (4 points)**

   Consider changing B-TREE-SEARCH to use **binary search** instead of linear search on the key.

   (a) What is the number of disk accesses?

   (b) Show that the CPU time is $O(\log n)$, which is independent of k.

3. **Number of keys (8 points)**

   (a) (4 points) The *black-height* of a red-black tree is the black-height of its root vertex. What is the largest possible number of keys that a red-black tree of black-height b can store? Your answer should depend on b. Justify your answer.

   (b) (4 points) Suppose you have a B-tree of minimum degree k and height h. What is the largest number of keys that can be stored in such a B-tree? Your answer should depend on k and h. *(Hint: This is similar to slide 12 in the B-tree slides.)*

4. **Red-black trees and (2,3,4)-trees (8 points)**

   A $(2, 3, 4)$-tree is a B-tree with minimum degree $k = 2$. As we have seen on slides 11–16 of the red-black tree slides in class, a red-black tree can be converted into a $(2, 3, 4)$-tree by merging the red nodes into their black parent nodes.

   (a) (2 points) Convert the $(2, 3, 4)$-tree below into a corresponding (valid) red-black tree.

   (b) (2 points) Describe how an arbitrary $(2, 3, 4)$-tree can be converted into a valid red-black tree. Justify why your conversion yields a valid red-black tree.

   (c) (2 points) Insert the number 64 into the $(2, 3, 4)$-tree below and show the resulting tree.

   (d) (2 points) How do the B-tree operations for inserting 64 correspond to operations in the corresponding red-black tree?