Advanced Trees

Updated: 21st July, 2023

Aims

- To manually work through the algorithms for self-balancing trees:
  - 2-3-4 Trees
  - Red-Black Trees
  - B-Trees
- To compare the resulting trees created with each of the strategies.

Before the Practical

- Read this practical sheet fully before starting.
- No coding required.

Activities

1. Red-Black Trees

Create three Red-Black Trees using the following lists of numbers with the demo app:

- [10, 5, 50, 35, 40, 15, 95, 65, 20]
- [5, 10, 20, 30, 40, 50, 60, 70]
- [100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 5]

Demo App: https://www.cs.usfca.edu/~galles/visualization/RedBlack.html

Manually create the same trees using the same algorithm from the lectures, drawing the tree after each insertion. Check it against the solution from the demo app. Do they differ?

2. 2-3-4 Trees

Create three 2-3-4 Trees using the demo app with the demo app:

- [10, 5, 50, 35, 40, 15, 95, 65, 20]
- [5, 10, 20, 30, 40, 50, 60, 70]
- [100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 5]

Demo App: https://yongdanielliang.github.io/animation/web/24Tree.html

Manually create the same trees using the same algorithm from the lectures, drawing the tree after each insertion. Check it against the solution from the demo app. How is the algorithm in the lectures different to the visualisation?

Now, manually convert the manually created 2-3-4 Trees into Red-Black Trees and check them against your solutions for Activity 1.

> **Note:** When converting 2-3-4 Trees to Red-Black Trees, there are two ways you can arrange a two-key node and still have a valid tree. This means you may not get the same tree as you did creating the Red-Black Tree directly. In the lecture notes, the middle key was set up as the parent of the first key - use this approach.

3. B-Trees

Create three B-Trees using the following lists of numbers with the demo app:

- [10, 5, 50, 35, 40, 15, 95, 65, 20]
- [5, 10, 20, 30, 40, 50, 60, 70]
- [100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 5]

Demo App: https://www.cs.usfca.edu/~galles/visualization/BTree.html

Manually create the same trees using the same algorithm from the lectures, drawing the tree after each insertion. Check it against the solution from the demo app. How is the algorithm in the lectures different to the visualisation? *(Max Degree on Demo = 4)*

> **Note:** The visualisations of the B-Trees split the node as it fills, not waiting for the 5th key to overflow it (as seen in the lecture slides). Use the lecture slide approach.

4. Reflection

Answer the following:

- Compare the heights of the resultant trees - how do they compare with a Binary Search Tree (BST) for the same input values?
- Compare the complexity of the algorithms, how much work would be required for the main operations: `insert()`, `find()`, `delete()`? Compare this to a BST.
- Compare the understandability of the algorithms, which would be easier to implement?
- Describe how an in-order traversal would work on each type of tree.

Submission Deliverable

- Your work is due 2 weeks from your current tutorial session.
  - You will demonstrate your work to your tutors during that session
  - If you have completed the practical earlier, you can demonstrate your work during the next session
- You must **submit** your work **electronically via Blackboard** under the *Assessments* section before your demonstration.

Marking Guide

Your submission will be marked as follows:

- [4] Your Red-Black Trees are correct
- [4] Your 2-3-4 Trees are correct
- [4] Your B-Trees are correct
- [8] Reflection- 2 each question.

End of Worksheet