Project D
Authors - Sruthi Anand and Diego Otero-Caldwell

Part B Answers

---

1. **Which version of the tree had the fastest processing:**
    - **the tree from Comparison A (built from sorted list, ascending)**
    - **the tree from Comparison B (built from shuffled list)**
    - **or the tree from Comparison C (build from sorted list, descending)?**
    - **Why do you think that is?**

Comparison B is faster as the time required to build the tree took less compared to the other two comparisons. We think this is because shuffled list creates a balance when building a tree which increases efficiency and decreases processing time.

2. **How can you explain the difference in building and processing time between Comparison A tree (built from sorted list, ascending) and the Comparison C tree (built from sorted list, descending)?**
    - **The tree built from the sorted list ascending took less time to build and process. Why?**
    - **Hint: Try drawing a small tree to see what is going on. Perhaps a tree built from 1, 1, 2, 2, 3, 4, 4, 5 and from 5, 4, 4, 3, 2, 2, 1.**

When the search tree was built from a sorted list in ascending order, equal values (reports with the same date) were placed into the left sub-trees of their identical nodes, and were taken off the long chain of right child nodes. This meant that the height of the tree when built in ascending order was equal to the number of unique dates in the range, while the height of the tree built in descending order was equal to the number of total reports. If equal nodes were placed into the right sub-trees of their parent nodes, this relationship would be reversed.

3. **For Comparison B, which processing was faster- the tree or the shuffled list? How would you describe the big-o of what was going on in the processing with these two structures?**

The binary search tree is faster. In terms of Big-O, the BST has a processing time of O(log n) which is faster than the shuffled list's processing time which is O(n).

4. **What characteristic of a binary search tree affects its efficiency?**

Balance is an important characteristic of a binary search tree as it allows for more efficient searches.