

Sarah Kayembe and Esther Greene

Task 1: Read the Reports in a Binary Search Tree

- **Algorithm Complexity:** $O(N \log N)$, where N is the number of reports.
- **Justification:** The difficulty comes from creating a binary search tree for each state when reading reports from the CSV file. Inserting each report into the BST takes $O(\log N)$ time on average, and because we do this for all N reports, the total complexity is $O(N \log N)$.

Task 2: Provide a menu to calculate the number of reports/accidents on and after a given date and state

- **Algorithm Complexity:** $O(\log N)$, where N is the number of reports for the given state.
- **Justification:** The complexity of searching for reports on or after a specific date in a binary search tree is $O(\log N)$, where N is the number of reports in the tree. Because we do this procedure for each state, the overall complexity is $O(\log N)$.

Task 3: Implement a recursive method to calculate the number of records

:

- **Algorithm Complexity:** $O(N)$, where N is the number of reports for the given state.
- **Justification:** The recursive approach goes around the binary search tree recursively, visiting each node once. Because we must traverse the entire tree to count the number of reports, the complexity is $O(N)$, where N represents the number of reports in the tree.

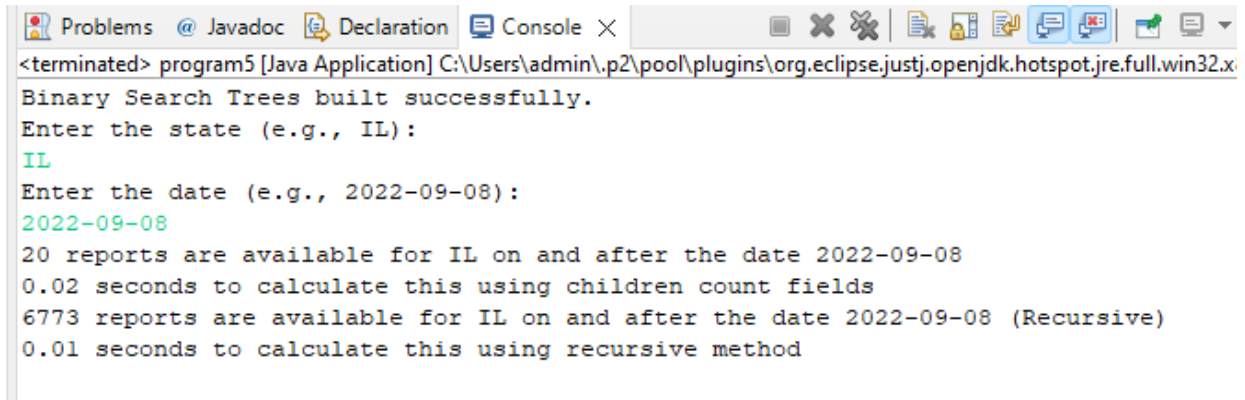
Which approach would you recommend for the final product? Justify your answer.

The recursive method showcased good performance, completing the task in just 0.01 seconds compared to calculating using children count fields (0.02 seconds), while also returning more reports (6773 reports vs. 20 reports). Given this result, it might seem that the recursive method is better.

However, it's essential to consider the trade-offs between time complexity and space complexity, as well as the potential impact on performance with larger datasets. The recursive approach operates with a time complexity of $O(N)$, where N represents the number of reports in the binary

search tree. As the dataset grows larger, the recursive method's performance might degrade significantly compared to methods with lower time complexities, such as the one utilizing children count fields.

Therefore, while the recursive method may excel in handling smaller datasets efficiently, its limitations in scalability and potential memory issues suggest that alternative methods with lower time complexities might be more suitable for larger datasets and real-world applications where performance and scalability are paramount like the children count fields.



```
<terminated> program5 [Java Application] C:\Users\admin\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x
Binary Search Trees built successfully.
Enter the state (e.g., IL):
IL
Enter the date (e.g., 2022-09-08):
2022-09-08
20 reports are available for IL on and after the date 2022-09-08
0.02 seconds to calculate this using children count fields
6773 reports are available for IL on and after the date 2022-09-08 (Recursive)
0.01 seconds to calculate this using recursive method
```