CL2001 – Data Structure Lab

Assignment # 2

Note:

- Submit a pdf file containing all your C++ code with all possible screenshots of every task output on Google Classroom.
- **Also submit .cpp extension file with pdf.**
- Copied task will be awarded **zero** marks.
- Submit your file in this format (roll-no-name) i.e (23P-1234-Ali.pdf).

Problem: 1 - Implementation of Auto-Completion System using AVL Trees

Auto-completion systems are those intelligent features you often encounter while typing text in search engines, word processors, or messaging applications. As you start typing a word or phrase, the system suggests possible completions based on what you've typed so far. **AVL Trees** can be employed to implement such systems efficiently.

Here's how it works:

1. **Dictionary of Words and Frequencies**: The AVL Tree is used to store a dictionary of words where each word is associated with its frequency or popularity. This frequency can be based on various factors such as how often the word appears in a corpus of text or how frequently it's been selected by users in the past. You can hardcode some words along with their frequencies into the AVL Tree during initialization.
2. **Partial Inputs**: As the user starts typing, the auto-completion system receives partial input from the user. For example, if the user has typed "ca," the system needs to suggest possible completions like "car," "cat," "calendar," etc.
3. **Prefix Search**: The AVL Tree is structured in such a way that words with similar prefixes are grouped together. This allows for efficient prefix search operations. When the user types a partial input, the system traverses the AVL Tree starting from the root and descends into the subtree corresponding to the prefix entered by the user.
4. **Frequency-Based Suggestions:** Prioritize suggestions based on word frequencies. When retrieving words from the subtree corresponding to the entered prefix, prioritize words with higher frequencies. This ensures that more popular words are suggested first, improving the relevance of auto-completion suggestions.
5. **Efficiency**: AVL Trees ensure that the auto-completion system remains efficient even for large dictionaries of words. Their self-balancing property guarantees that the tree remains reasonably balanced, resulting in fast search and retrieval operations.

6. **Dynamic Updates**: As users interact with the auto-completion system, selecting suggestions or entering new words, the AVL Tree can be dynamically updated to reflect changes in word frequencies or to incorporate new words into the dictionary.

By utilizing AVL Trees in this manner, auto-completion systems can provide users with accurate and relevant suggestions in real-time as they type, enhancing the user experience and productivity in various applications.

Sample Output:

Auto-completion system

Enter word: ca

car

calendar
