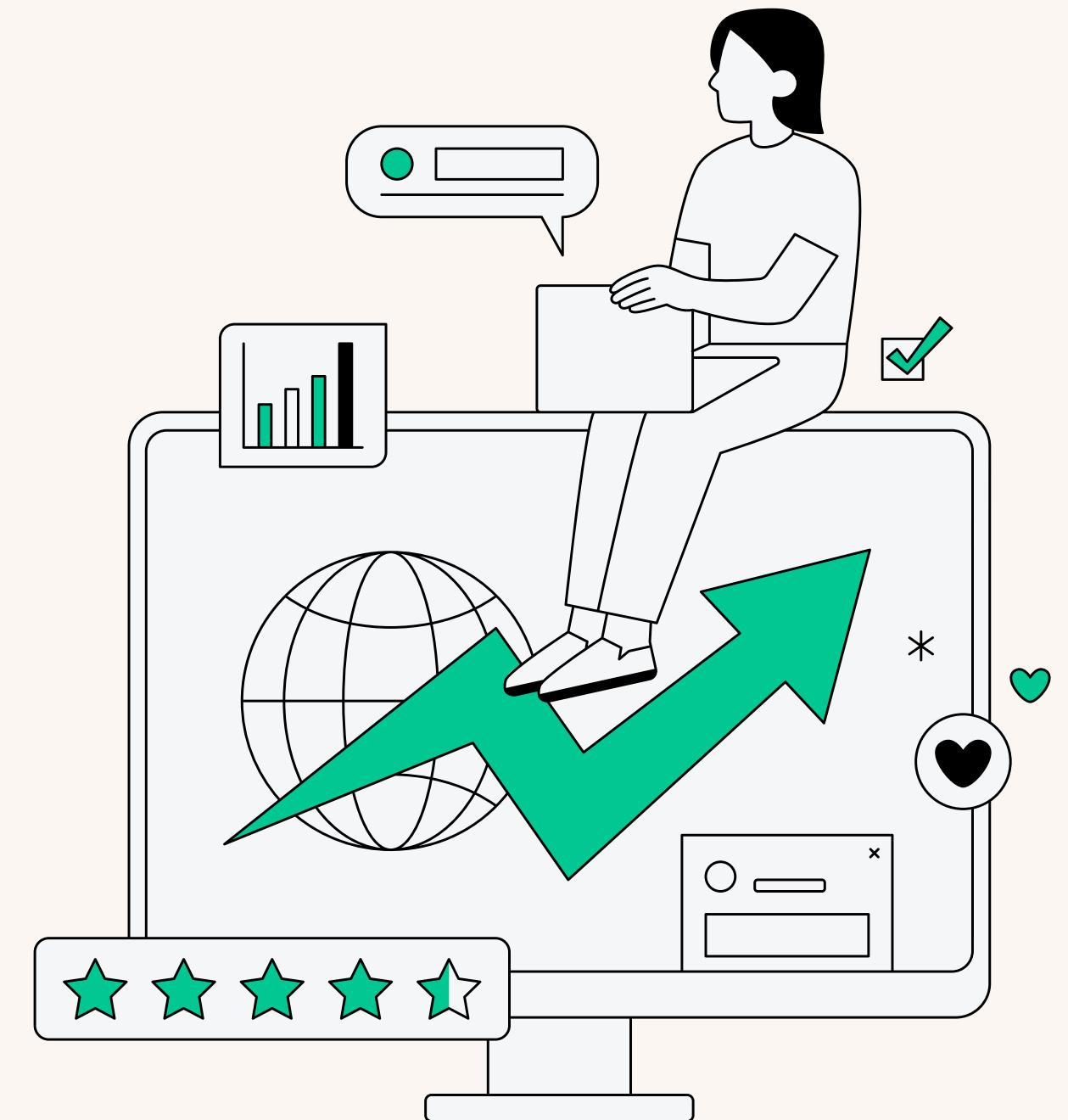
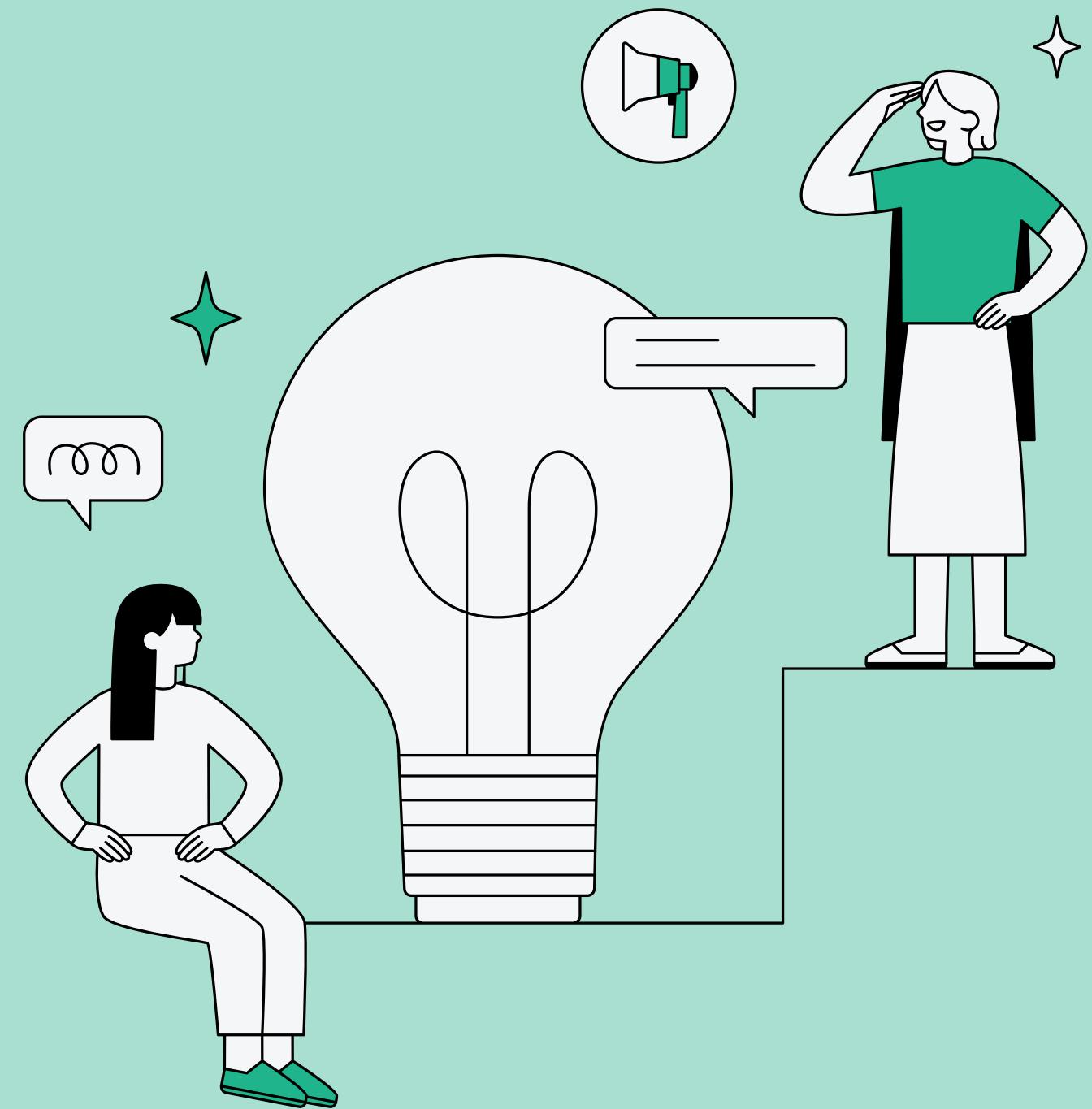


Binary Search Tree - Online Book Store



Introduction

- A Binary Search Tree (BST) is a tree structure used to efficiently store and manage items with a specific ordering property.
- It's commonly used in computer science for searching, insertion, and deletion operations.
- In an online bookstore context, it can help manage and search for books based on attributes like ISBN.
- BST nodes follow an ordering property, making it easy to search and insert items.
- Keeping the BST balanced is crucial for efficient operations, with time complexities of $O(\log n)$ when balanced and $O(n)$ when unbalanced.





Objectives



- Develop a Binary Search Tree (BST) for efficient book storage and management.
- Enable efficient searching, retrieval, and sorting of books based on various criteria.
- Implement functions for adding and removing books with $O(\log N)$ complexity.
- Optimize for scalability to handle a large number of books efficiently.
- Ensure robust error handling and validation for a seamless user experience.

Comparison

Current Results

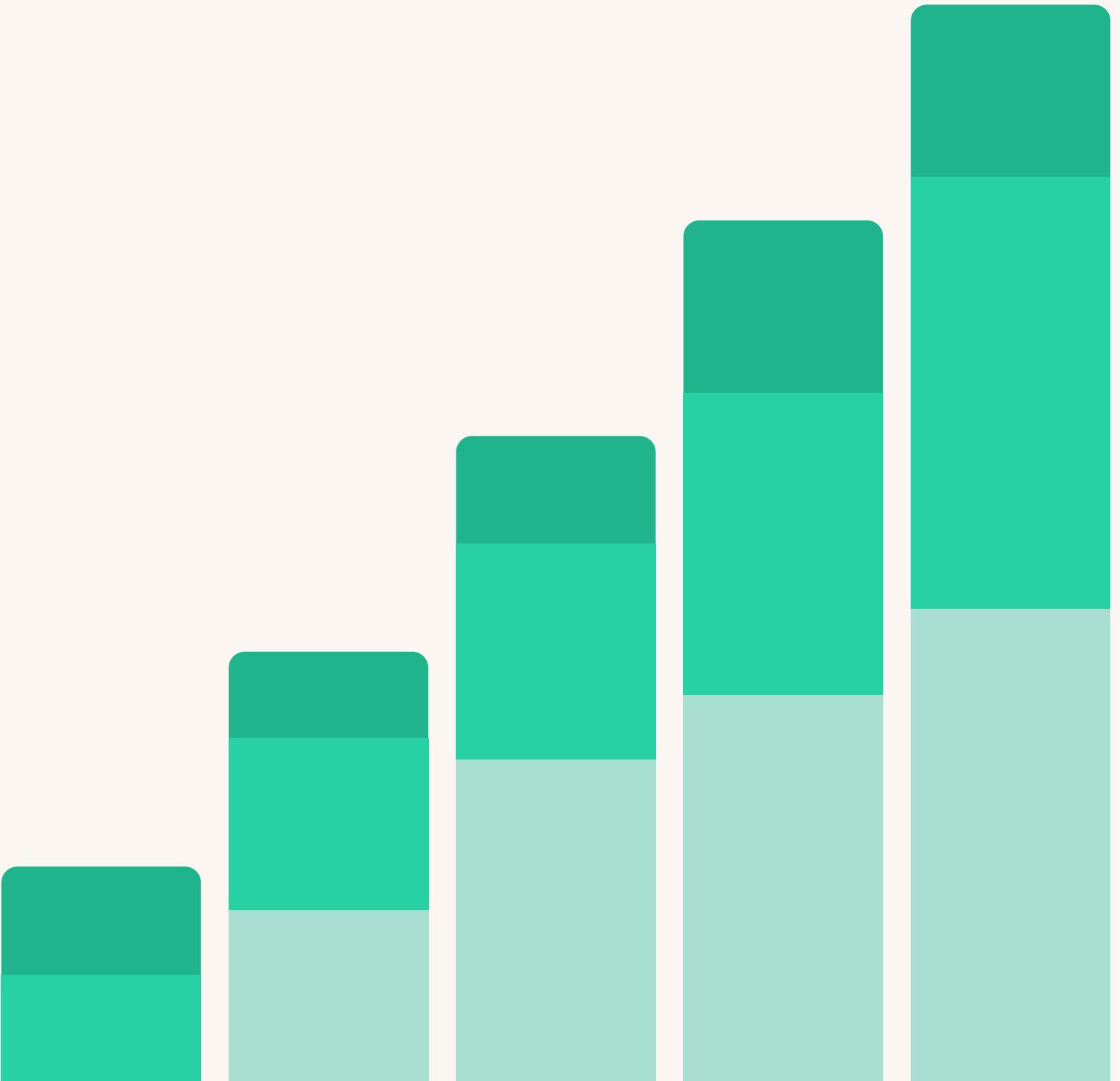
- Improved balance through self-balancing trees.
- Better performance with AVL, Red-Black, or Splay trees.
- Still not optimal for range queries.
- Average case: $O(\log n)$ time complexity.
- Efforts to reduce overhead and optimize search.

Previous Results

- Search time depends on tree height.
- May become unbalanced if not managed.
- Ordered data insertion can lead to skewed trees.
- Inefficient for range queries.
- Average case: $O(\log n)$ time complexity.

Methodology

- Define the problem: Improve online bookstore book search efficiency using Binary Search Trees (BST).
- Literature review: Research existing BST applications in information retrieval and online bookstores.
- Data collection: Gather book dataset for constructing the BST.
- Design and implementation: Create efficient BST algorithms for book insertion, search, and deletion.
- Performance metrics: Measure search time, insertion time, and memory usage to evaluate BST effectiveness.



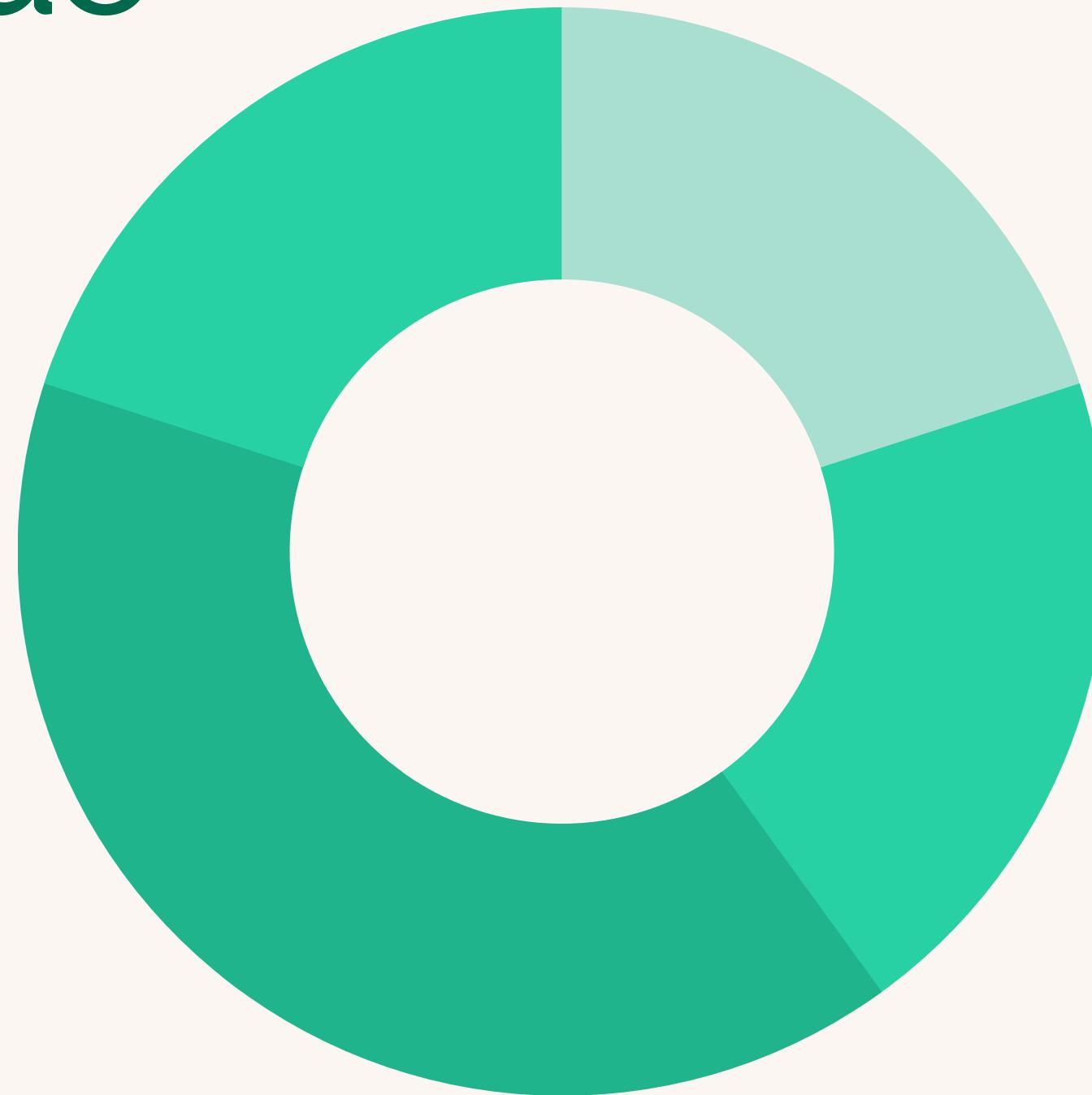
Key Topics Covered

- BST for efficient online book management.
- Insertion, deletion, and tree balance importance.
- Efficient searching, time complexity.
- In-order traversal for sorted lists.
- Real-world benefits: search speed, inventory.
- Efficient data retrieval with BST.
- Balancing for optimal performance.
- Support for range queries.
- Scalability and dynamic data management.
- Storing metadata and user preferences.



System Design of Code

- Data structures: Define structures for books and tree nodes.
- Create node: Create a new tree node with a given book and initialize its left and right pointers.
- Insert: Insert a book into a binary search tree while maintaining its order.
- Search: Search for a book in the binary search tree by its title.
- Main function: Create a binary search tree, insert sample books, search for a book by title, and display book information in an inorder traversal.



Code Structure

1. Includes necessary header files.
2. Define a 'struct Book' to represent book information, and a 'struct Node' for the binary search tree structure.
3. Implement a function 'createNode' to create a new node for the BST and return it.
4. Implement an 'insert' function to add a new book to the BST.
5. Implement a 'search' function to search for a book in the BST by its title.
6. Create a function 'displayBookInfo' to display book information.
7. Define an 'inorderTraversal' function to traverse and display the books in the BST in sorted order.
8. In the 'main' function:
 - Initialize the root of the BST as NULL.
 - Define sample book data in an array.
 - Insert the sample books into the BST using a loop.
 - Prompt the user to search for a book by title and display the result.
 - Display the entire book list using an inorder traversal.

Output

Output

```
/tmp/D1FKOAgSaX.o
Welcome to the Online Book Store
Enter the title of the book you want to search: Book1
Book found!
Title: Book1
Author: Author1
Year: 2020

Book List:
Title: Book1
Author: Author1
Year: 2020
Title: Book2
Author: Author2
Year: 2019
Title: Book3
Author: Author3
Year: 2021
Title: Book4
Author: Author4
Year: 2018
```

Output

```
/tmp/D1FKOAgSaX.o
Welcome to the Online Book Store
Enter the title of the book you want to search: Book 5
Book not found.

Book List:
Title: Book1
Author: Author1
Year: 2020
Title: Book2
Author: Author2
Year: 2019
Title: Book3
Author: Author3
Year: 2021
Title: Book4
Author: Author4
Year: 2018
```

Learning & Future Scope

01.

Balanced BSTs: Employing AVL and Red-Black trees for efficient book data management, ensuring $O(\log n)$ search time.

02.

Custom BST Variants: Tailoring BSTs for specific book attributes, enhancing user experience in searching.

03.

Concurrency & Multithreading: Fine-grained locking and thread-safe BSTs for data integrity in multi-user systems.

Conclusion

01.

Binary Search Trees (BSTs) significantly enhance data organization and retrieval in online bookstores, offering efficient search and management capabilities.

02.

Implementing BSTs in online bookstores optimizes search, accommodates diverse criteria, and streamlines inventory, resulting in an improved user experience and business success. Improves search efficiency by ($O(\log N)$)



Thank you !

Presented by:
Akshata Gedam
Prachi Kale
Himanshi Kawade
Ayushi Latkar
Akshata Udapurkar

