Data Structures and Algorithms

EasyShopHub

Course Project Report

School of Computer Science and Engineering
2023-24

Contents

1. Course and Team Details

1.1 Course details

Course Name	Data Structures and Algorithms
Course Code	23ECAC203
Semester	III
Division	F
Year	2023-24
Instructor	KMMR

1.2 Team Details

Roll No	Name	USN
111	Pragati Itigowni	01FE22BCI013
114	Disha Kalyanshettar	01FE22BCI016
126	Drakshayani	01FE22BCI028
129	Aishwarya Gopal	01FE22BCI033
131	Bhoomika Marigoudar	01FE22BCI035

1.3 Report Owner

Roll No.	**Name**
131	Bhoomika Marigoudar

2. Introduction

The selected project revolves around the development of algorithms and tools aimed at assisting businesses in designing, managing, and optimizing their warehouses. This initiative seeks to enhance overall efficiency, reduce operational costs, and minimize the environmental impact associated with warehouse operations. The domain encompasses a multidisciplinary approach, incorporating elements of logistics, supply chain management, and environmental sustainability. Efficient warehouse management is pivotal for businesses to meet customer demands, streamline operations, and remain competitive in the dynamic market landscape.

3. Problem Statement

3.1 Domain

can we develop algorithms and tools to help businesses design, manage, and optimize their warehouses in order to improve efficiency, reduce costs, and minimize environmental impact?

The chosen need statement asks if we can create tools and algorithms to help businesses run their warehouses better—making them more efficient, cost-effective, and environmentally friendly. The motivation behind this choice is the noticeable challenges companies face in managing warehouses amid a constantly changing market and the need to reduce environmental impact. Traditional methods are falling short, and there's a clear demand for smart solutions that not only improve operations and cut costs but also align with efforts to be more environmentally conscious. Warehouses are crucial in supply chains, and enhancing their design and management can bring significant advantages. By addressing this need, the project aims to offer practical solutions that cater to the current challenges in the logistics and supply chain industry while promoting sustainability. I am an Amazon Future Engineer Scholar , I had got an opportunity to visit the Amazon Office in Bengaluru , when I had been there we had the session wherein we were introduced to Amazon warehouses , the Warehouse Manager gave us a glimpse of how the things are managed in there and what all hurdles they used to faced and how tough was it for them during the COVID time . Since , I had seen all these things it motivated me to work towards this.

3.2 Module Description

In this project, my individual contributions focused on several key modules. I designed and implemented the user signup and login functionalities, ensuring a secure authentication process. Additionally, I developed the modules related to managing items, including storing, finding, displaying, and updating their status. For efficient order processing, I implemented a Depth-First Search (DFS) traversal algorithm, optimizing the handling of orders. Furthermore, I integrated a merge sort algorithm to enhance the sorting functionality to sort orders based on price within the system. These modules aimed to provide user-friendly experience in navigating and managing the features of the application.

4. Functionality Selection

Si. No.	Functionality Name	Known	Unknown	Principles applicable	Algorithms	Data Structures
	Name the functionality within the module	What information do you already know about the module? What kind of data you already have? How much of process information is known?	What are the pain points? What information needs to be explored and understood? What are challenegs?	What are the supporitng principles and design techniques?	List all the algorithms you will use	What are the supporting data structures?

1	User Signup	Creating user accounts with unique usernames and passwords.	- Unique username and Password - Implementing secure password	brute force/space and time trade off.Security:Hashing for password storage	- Brute force for string check during username and password validation.	- Hash table because array cannot ensure uniqueness also linkedlist Efficient memory usage and uniqueness are the advantages of hashtables over linkedlist ,array.
2	User Login	Checking for valid usernames and passwords.	- Ensuring secure authentication without exposing sensitive information. - Managing failed login attempts to prevent brute force attacks . - Proper redirection after success	brute force/space and time trade off.Security:Hashing for password comparrison	Brute force for string check during login credential validation.	Hash table because array cannot ensure uniqueness also linkedlist

			ful login.			
3	Store_item	Storing items (using arrays or linked lists).	- Choosing between arrays and linked lists based on operations needed (addition, deletion, retrieval). - Efficient Storage.	Linear Operation	-basic add/delete operations for arrays or linked lists.	- Arrays or linked lists
4	Find Item	Finding items using linear search,Binary Search….etc	Linear search might become inefficient for larger datasets. - Finding . alternative search algorithms for larger dataset	divide and conquer	Binary Search Algorithm	Merge sort

			s such as binary search using ORDER_ID			
5	Display Items	Iterates through the order history list and prints item details.	Number of items	Linear Operation	Linked List (OrderHistoryNode).	Arrays,Linkedlist
6	UpdateOrderStatus Function	Root of the order status history (Segment Tree), customer name, order ID, and order status.	- Specific logic for updating the Segment Tree is unknown. - *Information to Explore:* - Detailed logic for updating the Segment Tree. - *Challenges:* - Implementing efficient Segment Tree	Lazy Propogation	Segment Tree update logic	Segment Tree

			update logic.			
7	DFS Order Traversal	DFS Order traversal for traversing the orders placed and displaying the same	Order duplicate display ,Empty Order History, DFS might not be the most efficient traversal method for large order histories .	DFS	Depth First Search	DFS Using linkedlist
8	Merge sort	Order history list. Implements merge sort to sort the order history list based on item price.	Specific sorting logic is unknown-Implementing efficient merge sort logic.	Divide and Conquer	Merge Sort.	Linkedlist
9	Delete by OrderID	Delete's the customer information based on OrderD	OrderID which has to be deleted	Linear Operation	Deletion Algorithm	Linkedlist

5. Functionality Analysis

User Signup (userSignup)

Workflow:

1. Allocates memory for a new user.
2. Takes user input for username and password.
3. Inserts the new user at the beginning of the user list.

Efficiency Analysis:

- **Time Complexity:** O(1)
    - Basic input operations and memory allocation are constant time.
- **Space Complexity:** O(1)
    - Allocates memory for a single user.

User Login (userLogin)

Workflow:

1. Takes user input for username and password.
2. Iterates through the user list to find a match.
3. Prints login status.

Efficiency Analysis:

- **Time Complexity:** O(n)
    - In the worst case, it iterates through the linked list of users.
- **Space Complexity:** O(1)
    - Uses a constant amount of memory.

Store Item (storeItem)

Workflow:

1. Allocates memory for a new order.
2. Takes user input for customer name, order date, time, item name, price, and quantity.
3. Inserts the new order at the beginning of the order history list.

Efficiency Analysis:

- **Time Complexity:** O(1)
    - Basic input operations and memory allocation are constant time.
- **Space Complexity:** O(1)
    - Allocates memory for a single order.

Find Item (findItem)

Workflow:

1. Takes user input for the item name to find.
2. Iterates through the order history to find a matching item.
3. Prints the result.

Efficiency Analysis:

- **Time Complexity:** O(n)
    - In the worst case, it iterates through the linked list of orders.
- **Space Complexity:** O(1)
    - Uses a constant amount of memory.

Display Items (displayItems)

Workflow:

1. Iterates through the order history to print all items.

Efficiency Analysis:

- **Time Complexity:** O(n)
  - Iterates through the linked list to display all items.
- **Space Complexity:** O(1)
  - Uses a constant amount of memory.

Update Order Status (updateOrderStatus)

Workflow:

1. Allocates memory for a new order status node.
2. Updates the segment tree with placeholder logic.
3. Returns the new root of the segment tree.

Efficiency Analysis:

- **Time Complexity:** O(log n)
  - Depends on the implementation of the segment tree.
- **Space Complexity:** O(log n)
  - Recursive memory allocation.

Sort Customer Order History (mergeSort)

Workflow:

1. Splits the order history into two halves.
2. Recursively sorts the halves.
3. Merges the sorted halves.

Efficiency Analysis:

- **Time Complexity:** O(n log n)
  - Divide-and-conquer sorting algorithm.
- **Space Complexity:** O(log n)
  - Recursive memory allocation.

DFS Order Traversal (dfsOrderTraversal)

Workflow:

1. Allocates memory for a visited array.
2. Performs DFS traversal and prints order information.
3. Frees the visited array.

Efficiency Analysis:

- **Time Complexity:** O(n)
  - Visits each order once.
- **Space Complexity:** O(n)
  - Allocates memory for the visited array.

6. Conclusion

Working on this project has provided me with valuable insights into fundamental concepts such as data structures and algorithms. Through practical application, I learned how to leverage data structures like linked lists for managing user accounts and order histories, and implemented a segment tree for tracking order status history efficiently. The project also deepened my understanding of memory management, involving dynamic memory allocation (malloc) in C, and I gained insights into potential issues and error handling in memory allocation processes. Additionally, the implementation of a user authentication system, encompassing user signup and login functionalities, allowed me to explore principles of user security and credentials protection. Overall, this hands-on experience has enhanced my skills in programming, data management, and system design as an engineering student, providing practical insights that go beyond theoretical knowledge.

7. References

- https://www.geeksforgeeks.org/data-structures/
- Data Structures and Algorithms Made EasyBy Writer: Narsimha Karumanchi
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to data structures (3rd ed.). The MIT Press.

~*~*~*~*~*~*~*~