**Student Portfolio**

|  |  |
| --- | --- |
| **YOGESH S** | |
| **A person with a beard  Description automatically generated** | **Name: S. Yogesh**  **Register Number: RA2311056010018**  **Mail ID: ys2041@srmist.edu.in**  **Department: Data Science and Business System**  **Year / Sem/ Section: II / 3 /AL1** |
| **Subject Title: 21CSC201J Data Structures and Algorithms**  **Handled By: Dr. K. Rajkumar** | |
| **Email:** [**yogeshsasikumar2005@gmail.com**](mailto:yogeshsasikumar2005@gmail.com)  **LinkedIn: https://www.linkedin.com/in/yogesh-s-94a454322/**  **GitHub: https://github.com/YogeshSasikumar**  **Portfolio Website (if any):** **https://www.notion.so/Hi-I-m-YOGESH-133a88287f85809aa0f1cfeeab722cf0?pvs=4** | |
| **ELab Completion Status**  **A computer screen with a circular object  Description automatically generated** | |
| **Lab Experiment Completion status** | |
| **SOLVED REAL WORLD PROBLEM / CONCEPTUAL TASK**  **Overview:**  **A call center system efficiently manages customer inquiries by routing calls to available agents based on skills and workload. It uses real-time data and predictive analytics to optimize agent allocation, ensuring timely and relevant support for customers.**  **Queue Data Structures in Call Center System:**   1. **Queue Representation**    * **Elements:**      + **Each element in the queue represents a customer call, typically containing details such as Caller ID, Name, Priority Level (for VIP customers), and Call Reason.**    * **Attributes:**      + **Front: Points to the first call in the queue to be served next.**      + **Rear: Points to the last call added to the queue.**      + **Max Capacity: Maximum number of calls the queue can hold.**      + **Current Size: Tracks the current number of calls in the queue.** 2. **Operations**    * **Enqueue:**      + **Adds a new customer call to the end of the queue. If the queue is full, it either rejects new calls or places them on hold until space is available.**    * **Dequeue:**      + **Removes the call at the front of the queue to be processed by the next available agent. If the queue is empty, it notifies that no calls are waiting.**    * **Peek:**      + **Allows viewing of the call at the front of the queue without removing it, providing agents or managers with caller details to anticipate the next task.** 3. **Algorithm Implementation**    * **First Come, First Served:**      + **Processes customer calls in the exact order they arrived in the queue, ensuring fairness and maintaining a FIFO (First-In-First-Out) structure.**    * **Priority Queue:**      + **Calls are ordered by priority levels, where higher-priority calls (such as VIP or emergency calls) are served first. Lower-priority calls may be deferred if more urgent requests are added.**    * **Dynamic Queue Management:**      + **Adapts to real-time changes, such as call cancellations or escalations, ensuring that the queue always reflects the current situation accurately. For example, if a caller cancels while waiting, they are removed from the queue, and the next call moves up in priority**      + **.**   **Key Components :**   1. **Array/Linked List-Based Queues**    * **Array:**      + **A fixed-size array-based queue offers fast indexing and efficient memory usage for small call centers with a limited number of calls. However, it**      + **may reach its maximum capacity quickly, requiring alternative solutions or a predefined maximum size to avoid overflow.**    * **Linked List:**      + **A linked list-based queue is ideal for handling a dynamic number of calls without a fixed size. It supports dynamic memory allocation, making it suitable for high-traffic call centers where call volume fluctuates. Linked lists are flexible but require more memory due to pointer storage and are slightly slower in access time than arrays.** 2. **Circular Queue**    * **A circular queue efficiently utilizes space by treating the queue as a circular structure, where the rear wraps around to the front once the end of the queue is reached. This setup reduces space wastage and avoids frequent shifting of elements, which is particularly useful in call centers with predictable high and low traffic periods, as it optimizes memory usage.** 3. **Priority Queue**    * **Priority queues are essential for handling urgent or high-value calls, such as VIP customers or emergency issues. Calls are prioritized based on criteria like customer status, waiting time, or issue severity, ensuring that the most important calls are addressed first. This enhances customer satisfaction by giving high-priority cases immediate attention and reduces overall wait times for critical calls.** 4. **Dynamic Updates**    * **Dynamic updates allow real-time adjustments in the queue to reflect cancellations, priority changes, and new call entries. This flexibility ensures that the call center can respond to shifting demands, like call surges or high-priority cases, without disrupting the flow of service. With dynamic updates, the system remains accurate and efficient, even during unpredictable traffic, improving the overall responsiveness and resource allocation.**   **Skills Demonstrated**  **1. Data Structure Proficiency: Skillfully implemented and utilized data structures like queues (array-based, linked list, and priority queues) to manage and prioritize calls efficiently.**  **2.Algorithm Design:** **Designed and applied algorithms for call enqueueing, dequeueing, and priority management**  **3.** **Programming and Debugging Skills**  **Utilized languages like C for low-level control and efficient memory management. Demonstrated strong programming skills in creating modular, readable, and debugged code that efficiently handles call management tasks, supporting high reliability and stability in the call center system.** | |
| **CERTIFICATIONS (Coursera/edX/etc.)**  **Include any online courses, workshops, or certifications related to Data Structures and algorithms, problem-solving**  **A white card with black text  Description automatically generated** | |
| **CODING COMPETITIONS (Hacker Rank)**  **A screenshot of a computer  Description automatically generated** | |

**A close-up of a blue pen

Description automatically generated**

**Signature of the Student**