Depamaylo, Ma. Samantha Elizabeth M.
BSITWMA – AW12

Introduction

In the realm of data structures, a queue operates on the fundamental principle of First In, First Out (FIFO). Analogous to waiting in line at a ticket counter or a cashier, a queue arranges elements in a sequential manner, where the first element added is the first to be removed. This structure ensures that tasks or processes are handled in the order they arrive, making it an essential tool for efficiently managing waiting lines.

Within a medical centre, the concept of a queue finds practical application in various scenarios to streamline patient services and optimize workflow. One primary application is managing patient queues for appointments, consultations, and treatments. Patients are added to the queue as they arrive or schedule appointments, and healthcare professionals serve them based on their position in the queue. This ensures that patients are attended to in a fair and organized manner, respecting their order of arrival. Another significant use of queues in a medical setting is for managing lab tests. Patients requiring diagnostic tests are added to the queue, allowing the laboratory staff to conduct tests systematically based on patient arrival time. This approach helps priorities tests efficiently and ensures that patients receive their results promptly while maintaining a structured and orderly process flow.

At the Medical Centre, a ticket counter simulation scenario is implemented to replicate a real-life queue management system. The system operates through a queue data structure, where patients receive ticket numbers and are served in the order of their arrival. The simulation scenario unfolds as patients are issued ticket numbers and added to the queue, reflecting their entry into the medical center's service system. Upon starting the system, patients are served in the order of their ticket numbers, mimicking a real-life scenario where patients are attended to base on their arrival order. Each patient is displayed with their ticket number, and a visual representation of the serving process is simulated, ensuring patients are served methodically and efficiently. This ticket counter simulation at the Medical Centre exemplifies the practical application of a queue data structure in a healthcare environment, emphasizing the importance of orderly patient management and service delivery based on the FIFO principle.

Q and A is in the next page…

Depamaylo, Ma. Samantha Elizabeth M.
BSITWMA – AW12

Q n' A

A queue is a fundamental data structure that follows the First In First Out (FIFO) principle, meaning that the element added first is the one removed first. It is analogous to a queue of people waiting for a service, where the person who arrived earliest is served first. In this program, a queue is implemented using a linked list. Each element in the queue is represented by a Node containing the data (ticket number) and a pointer to the next Node. The enqueue function is used to add elements to the queue. When a new ticket is issued, a new Node is created and added to the rear of the queue. The dequeue function is used to remove elements from the queue. When a patient is served, the front of the queue is removed, simulating the process of serving the next customer in line. The isEmpty function checks if the queue is empty by verifying if the front pointer is NULL. This is important to prevent errors when trying to dequeue from an empty queue. The display function is used to show the current state of the queue, displaying each ticket number in the order they are in the queue.

Queues play a vital role in managing processes and scenarios like ticket counters due to their systematic and organized nature. In a ticket counter setting, queues help streamline customer flow, ensuring that customers are served in the order they arrive. This approach minimizes wait times, enhances customer satisfaction, and promotes an efficient and structured service delivery process. In ticket counter scenarios, queues are instrumental in managing customer flow and optimizing service delivery. By using a queue data structure, ticket counters can handle customer arrivals systematically, ensuring fair and organized customer service. Queues at ticket counters help prioritize tasks, maintain order, and optimize resource utilization, contributing to effective process management and enhanced customer experiences.

In the ticket counter simulation, if a request to start the system is received while there are still customers in the queue, the current implementation does not clear the queue. This could lead to unexpected behavior if new customers are added while the previous ones are still waiting. To address this, the program could be modified to clear the queue when the system is started, ensuring that all customers are served in the correct order and no customers are left waiting from a previous session. To include a maximum queue size and handle scenarios where the queue is full, the program would need to implement a check in the enqueue function to ensure that the queue does not exceed its capacity. If the queue is full, new customers would need to be turned away or placed in a separate overflow queue. This would help manage the flow of customers and prevent the queue from becoming too long, ensuring that the ticket counter can serve customers efficiently. One improvement for the ticket counter simulation could be to add a timer to simulate the time it takes to serve each customer. This would provide a more realistic simulation and allow for a better understanding of the time it takes to serve each customer. Additionally, implementing different types of tickets (e.g., regular, priority) could add complexity to the simulation and allow for more detailed analysis of queue management strategies. This would make the simulation more realistic and provide insights into how different types of customers are served in a real-world ticket counter scenario.