Problem Solving With C - UE24CS151B

Queue, Priority Queue

Prof. Sindhu R Pai
PSWC Theory Anchor, Feb-May, 2025
Department of Computer Science and Engineering

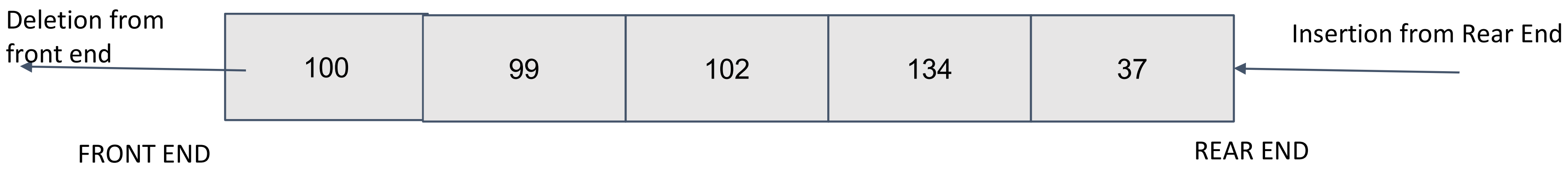
PROBLEM SOLVING WITH C

Queue, Priority Queue

1. Introduction to Queue
2. Operations
3. Types of Queues
4. Introduction to Priority Queue
5. Applications of Priority Queue
6. Implementation methods

Introduction to Queue

- A line or a sequence of people or vehicles awaiting for their turn to be attended or to proceed.
- In computer Science, a list of data items, commands, etc., stored so as to be retrievable in a definite order
- A Data structure which has 2 ends – **Rear end and a Front end**. Open ended at both ends
- Data elements are inserted into the queue from the Rear end and deleted from the front end.
- Follows the Principle of **First In First Out (FIFO)**



Deletion from front end

100	99	102	134	37

Insertion from Rear End

FRONT END

REAR END

Operations on Queue

- Enqueue – Add (store) an item to the queue from the Rear end.
- Dequeue – Remove (access) an item from the queue from the Front end.

Types of Queues

- **Ordinary Queue** - Insertion takes place at the Rear end and deletion takes place at the Front end

- **Priority Queue** - Special type of queue in which each element is associated with a priority and is served according to its priority. If elements with the same priority occur, they are served according to their order in the queue

- **Circular Queue** - Last element points to the first element of queue making circular link.

- **Double ended Queue** - Insertion and Removal of elements can be performed from both front and rear ends

Introduction

- Type of Queue where each element has a "**Priority**" associated with it.
- Priority decides about the Deque operation.
- The Enque operation stores the item and the “Priority” information
- Types of Priority Queue:

  Ascending Priority Queue: Smallest Number - Highest Priority

  Descending Priority Queue: Highest Number - Highest Priority

Applications of Priority Queue

1. Implementation of Heap Data structure.
2. Dijkstra’s Shortest Path Algorithm
3. Prim’s Algorithm
4. Data Compression
5. OS - Load Balance Algorithm.
6. …

Implementation of Priority Queue

- Using an Unordered Array
- Using an Ordered Array
- Using an Unordered Linked list
- Using an Ordered Linked List
- Using Heap

Implementation of Priority Queue Continued..

- Using an Unordered Linked list

Involves defining three new structures.

component: Type which contains **details** and **priority**

```
struct    component {    char details[20];          int priority;          };
```

node: Type that contains **component** and a **pointer to itself**

```
struct node{    struct component c;          struct node *link;        };
```

priority_queue: Type that contains **head which is a pointer to a Node**

```
struct priority_queue {      struct node *head; };
```

Implementation of Priority Queue Continued..

- Functions involved in the implementation

Enqueue: This function creates a node using the component from the client and adds this node in the beginning of the queue.

Dequeue: Deletes a node based on Priority field

Display: Display the nodes, in turn the components of the priority queue

THANK YOU

Department of Computer Science and Engineering

Dr. Shylaja S S, Director, CCBD & CDSAML, PESU
Prof. Sindhu R Pai - sindhurpai@pes.edu
Prof. Priya Badarinath, CSE, PESU