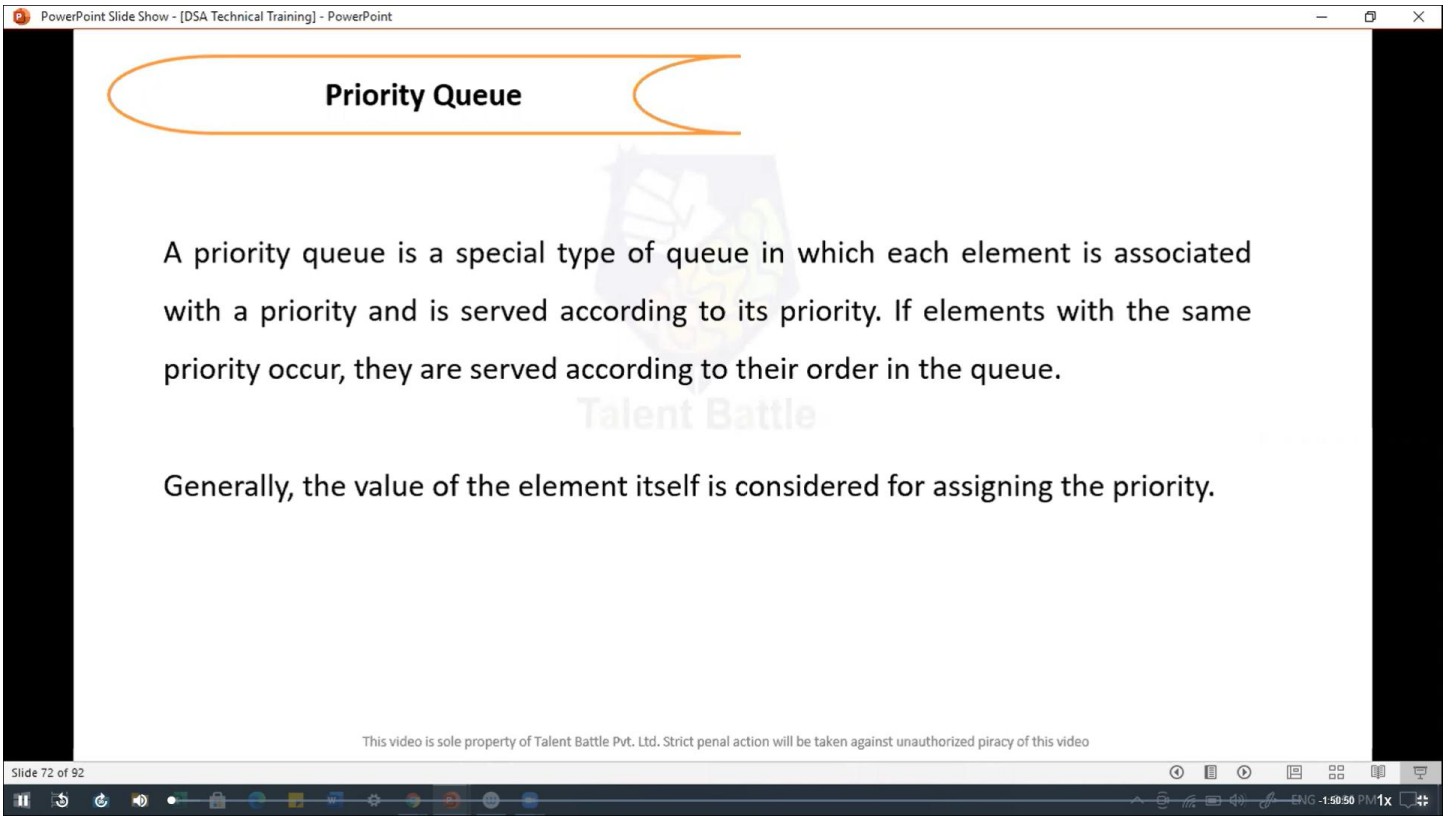
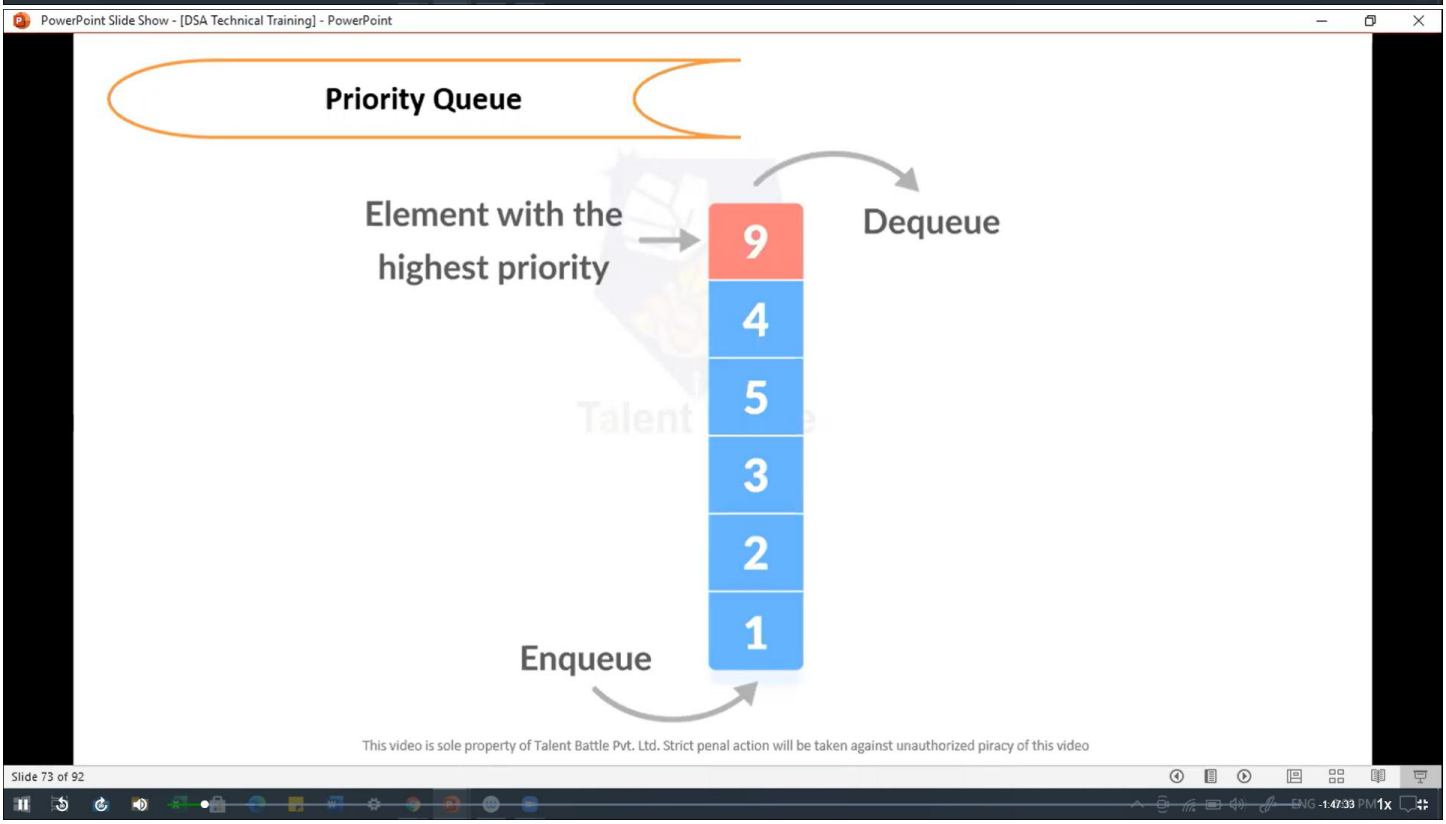
Session5

Linked list Part 1





PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

Priority Queue

Priority Queue Applications

Some of the applications of a priority queue are:

- Dijkstra's algorithm
- for implementing stack
- for load balancing and interrupt handling in an operating system
- for data compression in Huffman code

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

Double Ended Queue (Deque)

Deque or Double Ended Queue is a type of queue in which insertion and removal of elements can be performed from either from the front or rear. Thus, it does not follow FIFO rule (First In First Out).

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Double Ended Queue (Deque)

Types of Deque

•Input Restricted Deque
In this deque, input is restricted at a single end but allows deletion at both the ends.

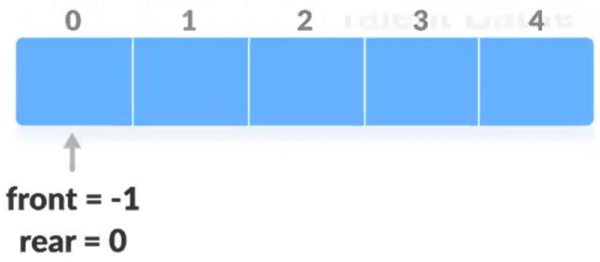
•Output Restricted Deque
In this deque, output is restricted at a single end but allows insertion at both the ends.

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Double Ended Queue (Deque)

Operations on a Deque

Before performing the following operations, these steps are followed.
1. Take an array (deque) of size n.
2. Set two pointers at the first position and set front = -1 and rear = 0.



This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

Double Ended Queue (Deque)

1. Insert at the Front

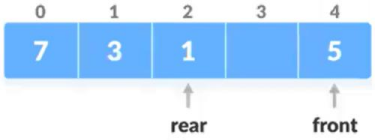
This operation adds an element at the front.

1.Check the position of front.

2. If front < 1, reinitialize front = n-1 (last index).

3. Else, decrease front by 1.

4. Add the new key 5 into array[front]



This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

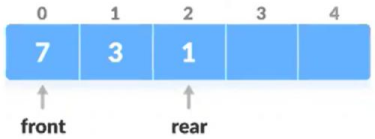
PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

Double Ended Queue (Deque)

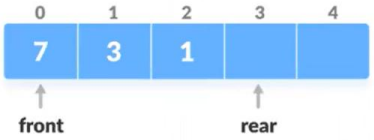
2. Insert at the Rear

This operation adds an element at the rear.

1.Check if the array is full.

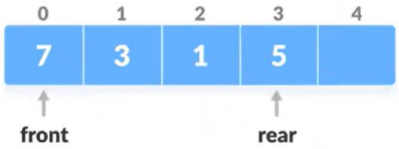


2. If the deque is full, reinitialize rear = 0 .



3. Else, increase rear by 1.

4. Add the new key 5 into array[rear]



This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

Double Ended Queue (Deque)

3. Delete from the Front

This operation deletes an element from the front.

1.Check if the deque is empty.

2. If the deque is empty (i.e. front = -1), deletion cannot be performed (**underflow condition**).
3. If the deque has only one element (i.e. front = rear), set front = -1 and rear = -1.
4. Else if front is at the end (i.e. front = n - 1), set go to the front front = 0.
5. Else, front = front + 1.

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

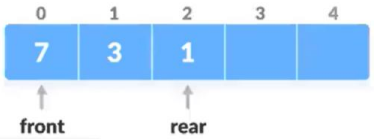
PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

Double Ended Queue (Deque)

4. Delete from the Rear

This operation deletes an element from the rear.

1.Check if the deque is empty.



2. If the deque is empty (i.e. front = -1), deletion cannot be performed (**underflow condition**).
3. If the deque has only one element (i.e. front = rear), set front = -1 and rear = -1.
4. Else if rear is at the front (i.e. rear = 0), set go to the front rear=n-1.
5. Else, rear = rear - 1.

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

Double Ended Queue (Deque)

5. Check Empty

This operation checks if the deque is empty .

Close

If front = -1, deque is empty.

6. Check Full

This operation checks if the deque is full.

If front = 0 and rear = n-1 OR front = rear+1, the deque is full.

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

Deque Implementation.

Time Complexity

The time complexity of all the above operations is constant i.e. O(1).

Applications of Deque Data Structure

1.In undo operations on software.

2.To store history in browsers.

3.For implementing both stacks and queues.

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

Linked List

A linked list data structure includes a series of connected nodes. Here, each node store the data and the address of the next node. For example,

You might have played the game Treasure Hunt, where each clue includes the information about the next clue. That is how the linked list operates.

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

Linked List

Representation of Linked List

Let's see how each node of the linked list is represented. Each node consists:
- A data item
- An address of another node

We wrap both the data item and the next node reference in a struct as:

```
struct node
{
int data;
struct node *next;
};
```

Understanding the structure of a linked list node is the key to having a grasp on it. Each struct node has a data item and a pointer to another struct node.

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

Linked List

```
/* Initialize nodes */
struct node *head;
struct node *one = NULL;
struct node *two = NULL;
struct node *three = NULL;

/* Allocate memory */
one = malloc(sizeof(struct node));
two = malloc(sizeof(struct node));
three = malloc(sizeof(struct node));
```

```
/* Assign data values */
one->data = 1;
two->data = 2;
three->data=3;

/* Connect nodes */
one->next = two;
two->next = three;
three->next = NULL;

/* Save address of first node in head */
head = one;
```

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

PowerPoint Slide Show - [DSA Technical Training] - PowerPoint

Linked List

The power of a linked list comes from the ability to break the chain and rejoin it.
E.g. if you wanted to put an element 4 between 1 and 2, the steps would be:

- Create a new struct node and allocate memory to it.
- Add its data value as 4
- Point its next pointer to the struct node containing 2 as the data value
- Change the next pointer of "1" to the node we just created.

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Linked List

Linked List Complexity

Time Complexity

	Worst case	Average Case
Search	O(n)	O(n)
Insert	O(1)	O(1)
Deletion	O(1)	O(1)

Space Complexity: O(n)

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Linked List

Linked List Applications

- Dynamic memory allocation
- Implemented in stack and queue
- In **undo** functionality of software's
- Hash tables, Graphs

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video

Linked List

Linked List Operations: Traverse, Insert and Delete

```
struct node
{
int data;
struct node *next;
};
```

Two important points to remember:

- head points to the first node of the linked list
- next pointer of the last node is NULL, so if the next current node is NULL, we have reached the end of the linked list.

Talent Battle

This video is sole property of Talent Battle Pvt. Ltd. Strict penal action will be taken against unauthorized piracy of this video