Priority Queue

- It is a collection of elements such that each element has been assigned a priority and the order in which elements are processed and deleted follows following rules:
- An element with higher priority is processed before any element with lower priority.
- Two elements with same priority are processed on FCFS (First Come First Serve) basis.

Operations on priority queue

- initialize(): Make the empty queue.
- full(): Check if the queue is full or not.
- empty(): Check if the queue is empty or not.
- enqueue(): Insert an element as per its priority.
- dequeue(): Delete the element in front(as front element has the highest priority).
- print(): Print queue elements.

Types/Representation of priority queue

1) One way list representation of a priority:

- Each node in a list will contain 3 items of an information: INFO(), PRIORITY_NO() and a link.
- A node X precedes node Y in the list where:

  1. X has higher priority than Y.

  2. When both have same priority but X was added in the list before Y.

Algorithm

INSERT

- It adds an item with priority number ‘n’ to a priority queue which is maintained in memory as one-way list.
- Traverse one- way list until you find a node X whose priority number exceeds ‘n’.
- Insert ITEM in front of node X.
- If no such element is found, insert ITEM as the last element of the list.

Algorithm

DELETE

- It deletes and processes the first element in a priority queue which appears in memory as one-way list.
- Set ITEM=INFO[START]
- Delete first node from the list
- Process ITEM
- Exit

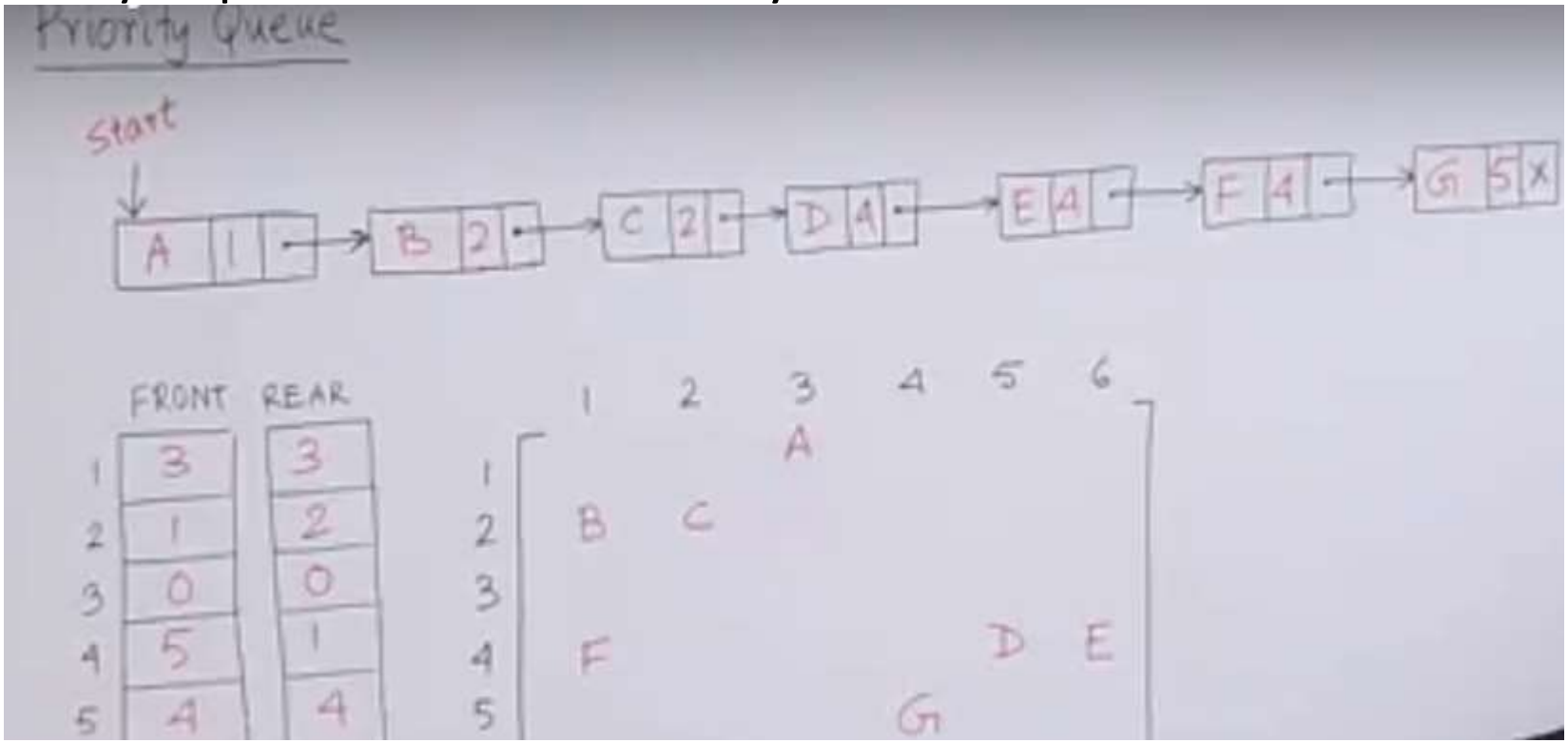
Types/Representation of priority queue

2) Array Representation of Priority Queue:

- Another way to maintain priority queue is to use a **separate queue for each level of priority.**
- Each such queue will appear in its own circular array and must have its own pair of pointers (rear and front).
- In each queue is allocated same amount of space, a 2D array of queue can be used instead of linear arrays.

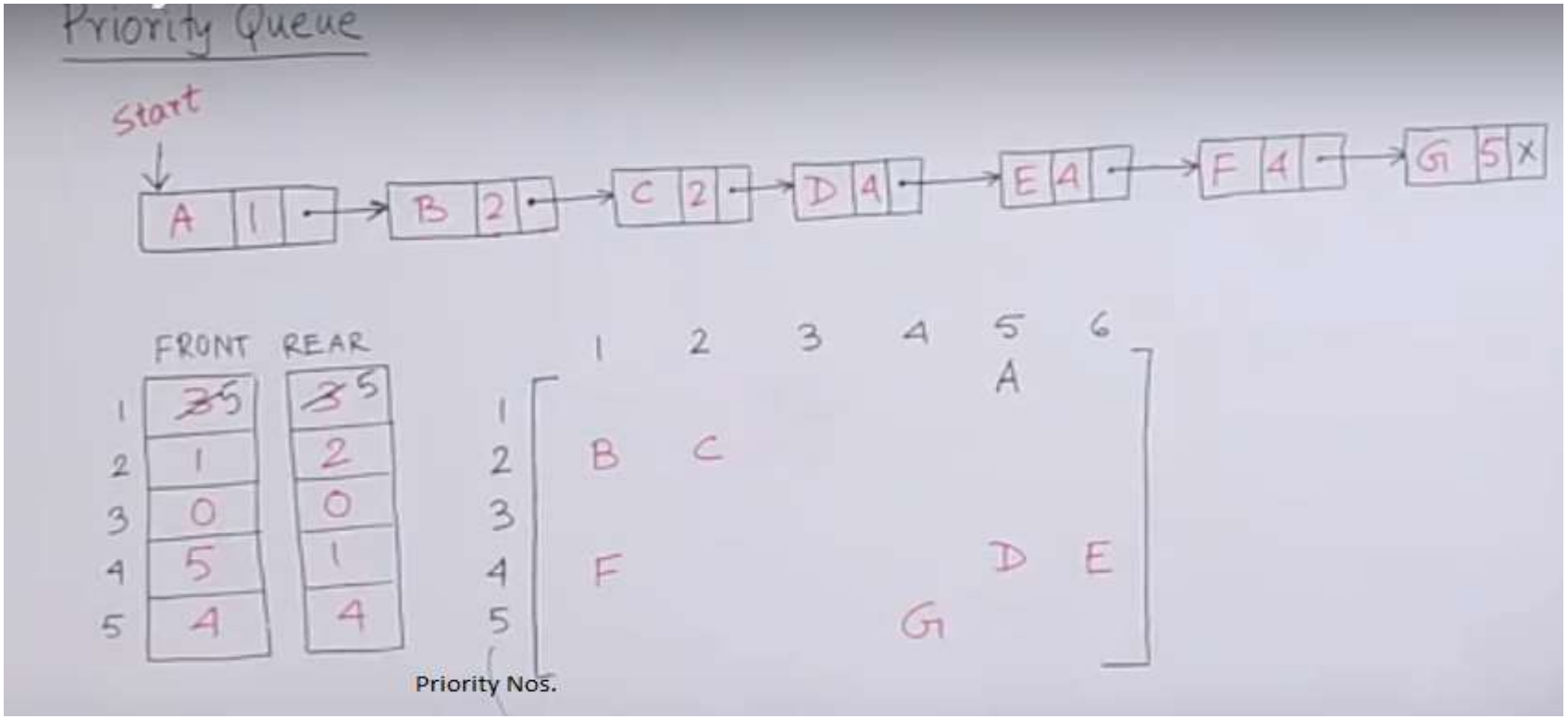
Types/Representation of priority queue

2) Array Representation of Priority Queue:



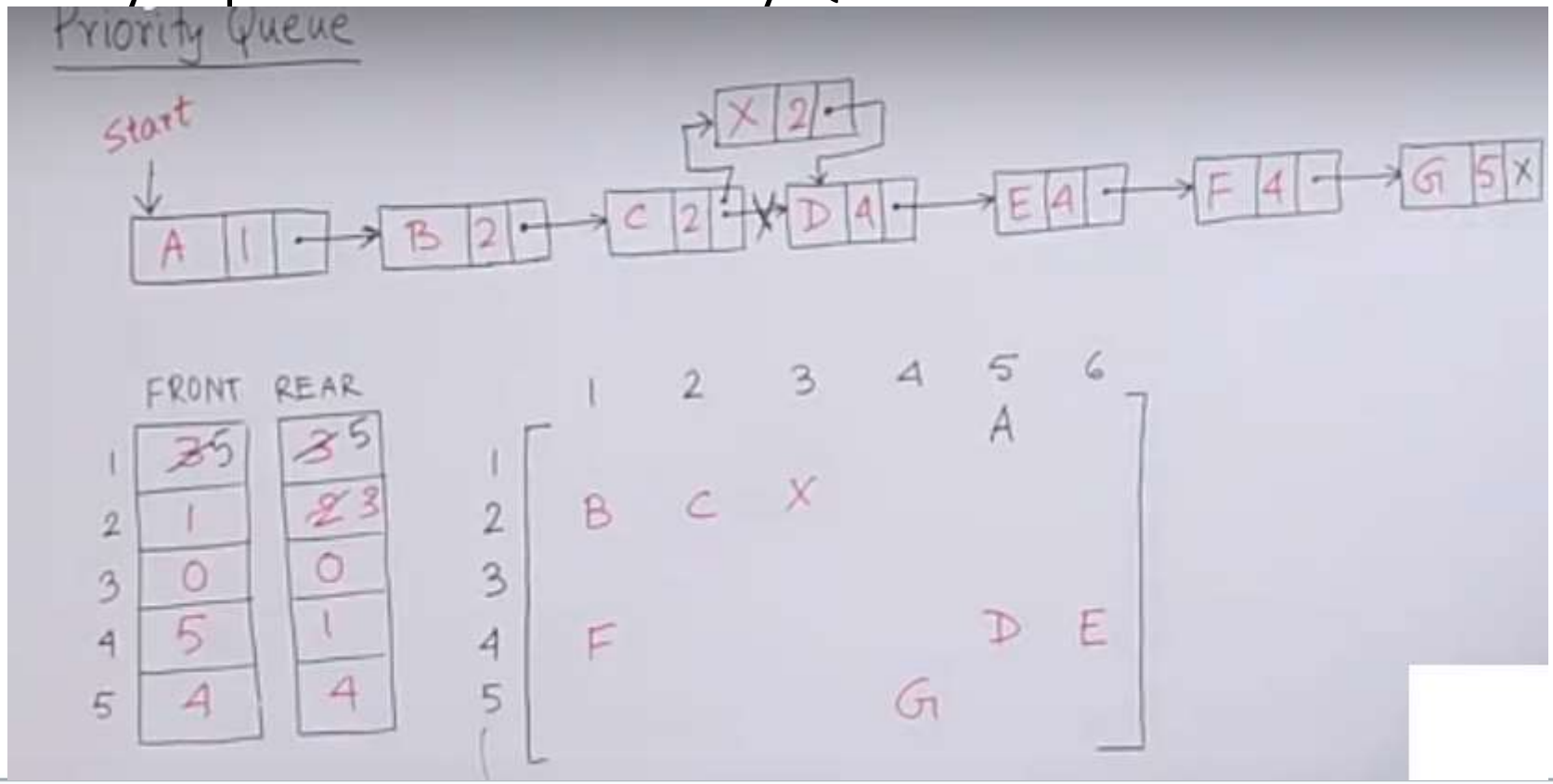
Types/Representation of priority queue

2) Array Representation of Priority Queue:



Types/Representation of priority queue

2) Array Representation of Priority Queue:



Implementation of priority queue

- Implementation with an unsorted list

- Performance:
  - insert takes ***O***(1) time since we can insert the item at the beginning or end of the sequence
  - remove take ***O***(***n***) time since we have to traverse the entire sequence to find the smallest key

- Implementation with a sorted list

- Performance:
  - insert takes ***O***(***n***) time since we have to find the place where to insert the item
  - remove and take ***O***(1) time, since the smallest key is at the beginning