Experiment - 15: Queue

Queue

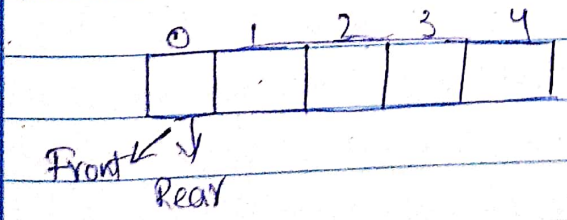
Queue is a linear data structre in which it is open at both the ends. One end is always used to insert data and the other end is used to remove the data. Queue follows first-in-first-out methodology i.e the data Queue follows for item stored first will be accessed first. We always use dequeue (or access) data pointed by front pointer and to enqueue (or storing) data in the queue, we take the help of the rear pointer. There are 4 types of queues - linear queue, circular queue, double ended queue and priority queue.

Example:-

Some examples of the queue in real life are-

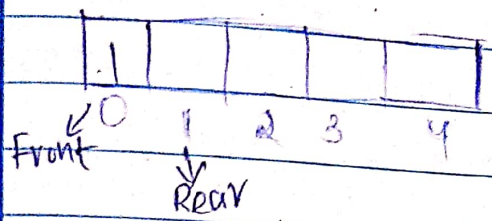
* people on a escalator
* cashier line in a store
* A car wash line
* One way exits

Let us take an empty queue:-

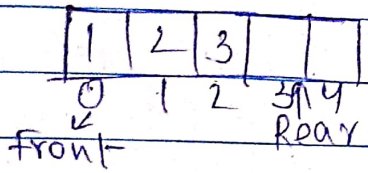


Initially, the head (front) and tail (rear) of the queue points to at the first index of the array, i.e. (row '0'. As we keep on adding elements, the rear keeps on moving ahead.

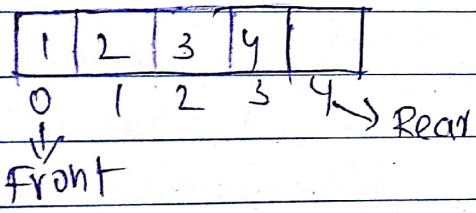
Now, let us add an element in the queue. The front will remain at index 0. Whereas the rear will move one index to the forward i.e to index 2.



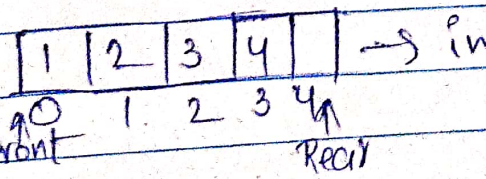
Now let us add another element. Now the rear will move to index 3.



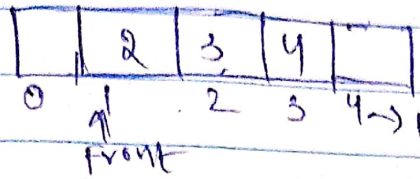
Now let us add another element, Now the rear will remain at index 4.



To remove an element from the queue, we remove the element from front position and then move front to the next position.
Let us remove the first element which is at index position 0. The element is 1. After removing the index 0 will be empty (as the element is removed) and the front will point to the next index, i.e. 1.

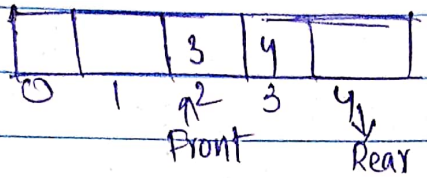


→ initial Queue



→ After removing first element

Now, let us remove the second element i.e 2. The index 1 will be empty and the front will point to index 2.



Code:

```
class Queue:
  def __init__(self,max):
      self.max=max
      self.q=[]
      self.rear=-1
      self.front=0
  elif choice==2:
      ele=q.deque():
      print("The deleted element is:",ele)
  elif choice==3:
      q.display()
  elif choice==4:
      exit()
```