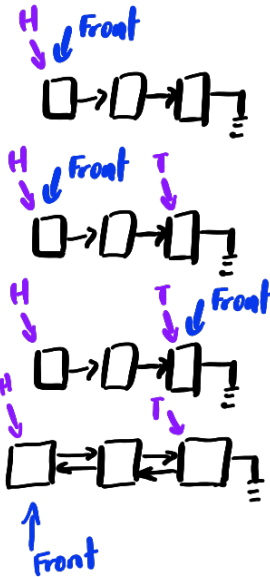


Recitation 10

1. What is the worst case complexity of the following:

Just Move
induces for
circular queue.

	Enqueue	Dequeue
Array with pointer to next available slot	$O(1)$	Shift array after delete $O(n)$
Array without pointer to next available slot	Find slot $O(n)$	Shift array $O(n)$
Circle Array	$O(1)$	$O(1)$
Singly linked list with head reference and head as the front	Add New Tail $O(n)$	Remove Head $O(1)$
Singly linked list with head and tail references with the head as the front	Add New Tail $O(1)$	Remove Head $O(1)$
Singly linked list with head and tail references with the tail as the front	Add New Head $O(1)$	Remove Tail $O(n)$
Doubly linked list with head and tail references with the head or tail as the front	Add New Head OR Tail $O(1)$	Remove Head OR Tail $O(1)$



2. Write a method called enqueue that accepts a *CircleArray* and an element to be added into the array. You can access the array, the front and the rear of the *CircleArray*.

```

public void enqueue(CircleArray arr, Object element)
{
    if (arr.getRear() + 1 % CAPACITY == arr.getFront())
        throw new FullQueueException("Queue is Full");
    else if (arr.getFront() == -1)
    {
        arr.getArray[0] = element;
        arr.setFront(0);
        arr.setRear(0);
    }
    else
    {
        arr.setRear((arr.getRear() + 1) % CAPACITY);
        arr.getArray[arr.getRear()] = element;
    }
}

```

