Design Queue

First In First Out

Adds to Rear : dummyTail.next

enQueue()

Removes from front : dummyHead.prev

deQueue()

5

4

3

2

1

dummyTail

Next

Rear

Front

Prev

dummyHead

Queue

enQueue(element)

deQueue()

getFirst()

getLast()

isEmpty()

ArrayList : ✗

LinkedList

SingleLinkedList

Can be possible put extra effort for prev reference

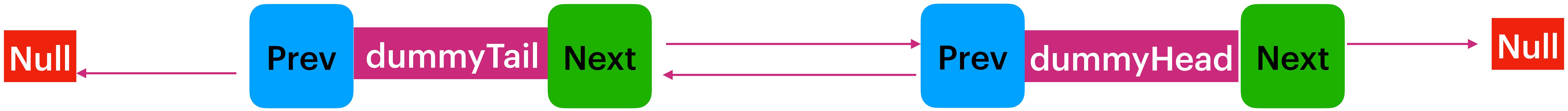
DoubleLinkedList ✓

Queue

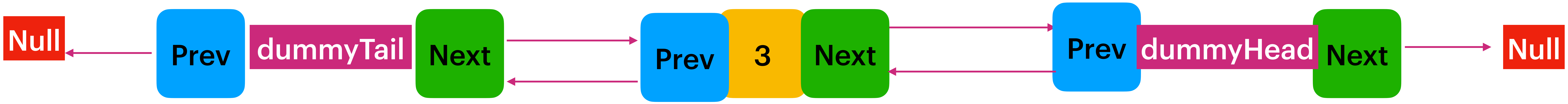
enQueue(element)

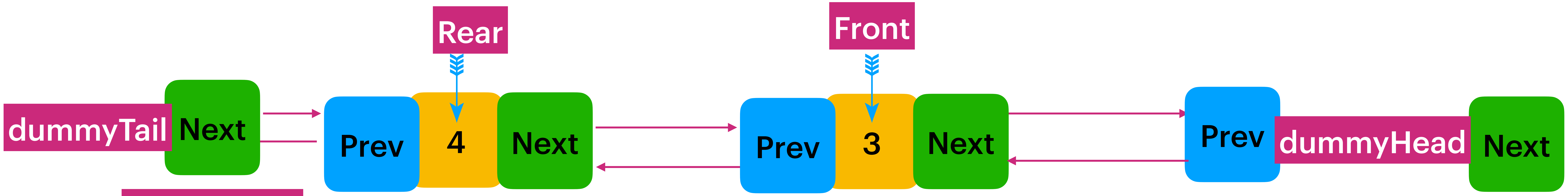
deQueue()

Represents Empty Queue



enQueue(3)

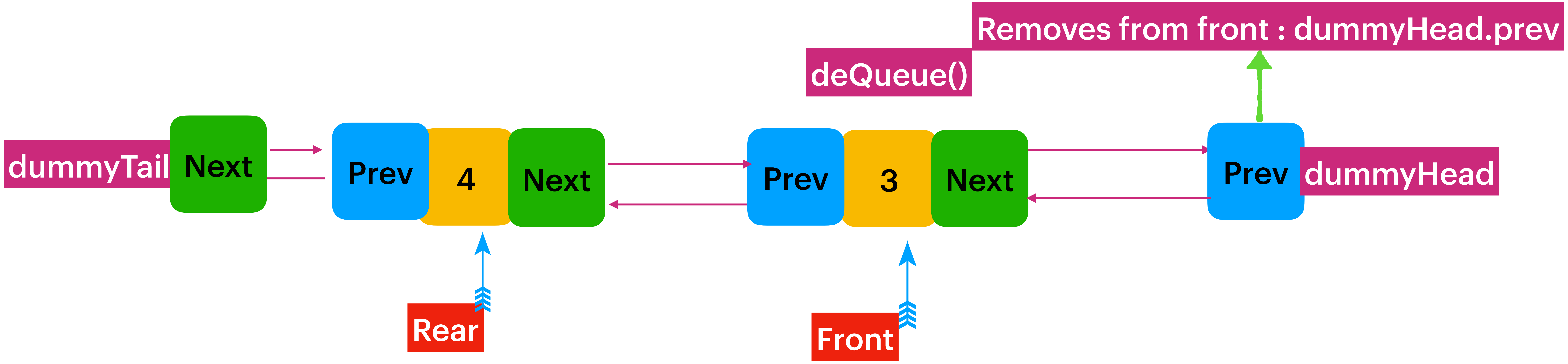




enQueue(4)

```
DLLNode tailNext = dummyTail.next;
DLLNode current = new DLLNode(value);
current.next = tailNext;
tailNext.prev = current;
dummyTail.next = current;
current.prev = dummyTail;
```

enQueue(element) ——>
Adds to dummyTail i.e dummyTail.next



deQueue() —> Removes from Front : i.e dummyHead.prev

```
DLLNode headPrev = dummyHead.prev;
dummyHead.prev = headPrev.prev
headPrev.prev.next = dummyHead

headPrev.next = null;
headPrev.prev = null; //  Helps GC
```