Deque: Its a double ended queue. We can insert and delete from front & rear.

It Represents Empty Deque:

dummyTail.next = dummyHead;
dummyHead.prev = dummyTail;

Delete

Delete

dummyTail | **Next** → | **Prev** | dummyHead

Add

Add

Delete

Delete

dummyTail | **Next** → | 5 | 4 | 3 | 2 | 1 | ← **Prev** | dummyHead

Add

Rear

Front

Add
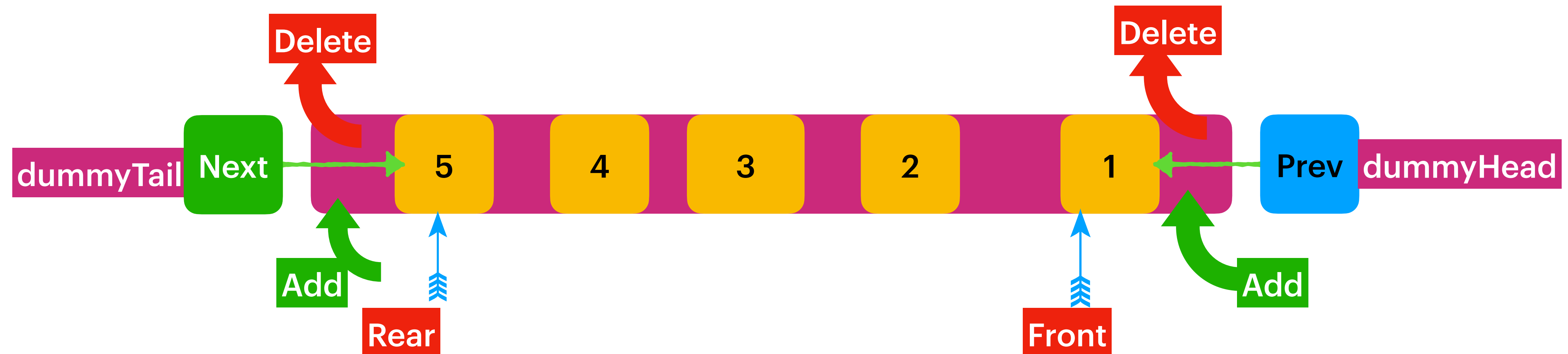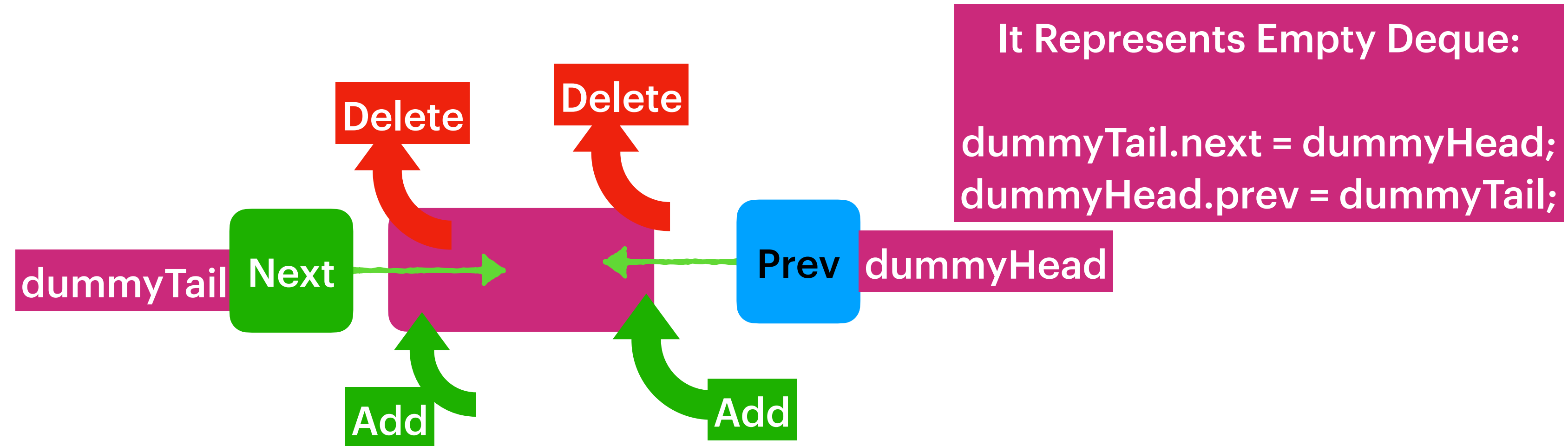
Design Deque :
It's a double ended Queue. We can add, remove from front & rear.

- addFirst(element)
- addLast(element)
- removeFirst(element)
- removeLast(element)
- front()
- rear()
- isEmpty()

Moving Average from Data Stream

Given a stream of integers and a window size, calculate the moving average of all integers in the sliding window.

Implement the MovingAverage class:

MovingAverage(int size) Initializes the object with the size of the window size.
double next(int val) Returns the moving average of the last size values of the stream.

Input
["MovingAverage", "next", "next", "next", "next"]
[[3], [1], [10], [3], [5]]
Output
[null, 1.0, 5.5, 4.66667, 6.0]

Explanation
MovingAverage movingAverage = new MovingAverage(3);
movingAverage.next(1); // return 1.0 = 1 / 1
movingAverage.next(10); // return 5.5 = (1 + 10) / 2
movingAverage.next(3); // return 4.66667 = (1 + 10 + 3) / 3
movingAverage.next(5); // return 6.0 = (10 + 3 + 5) / 3