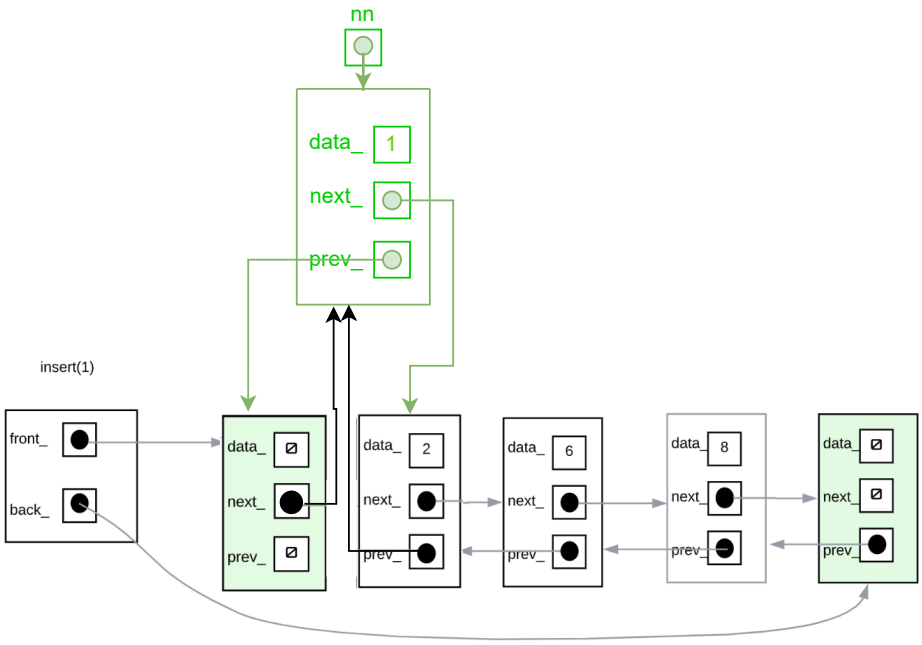
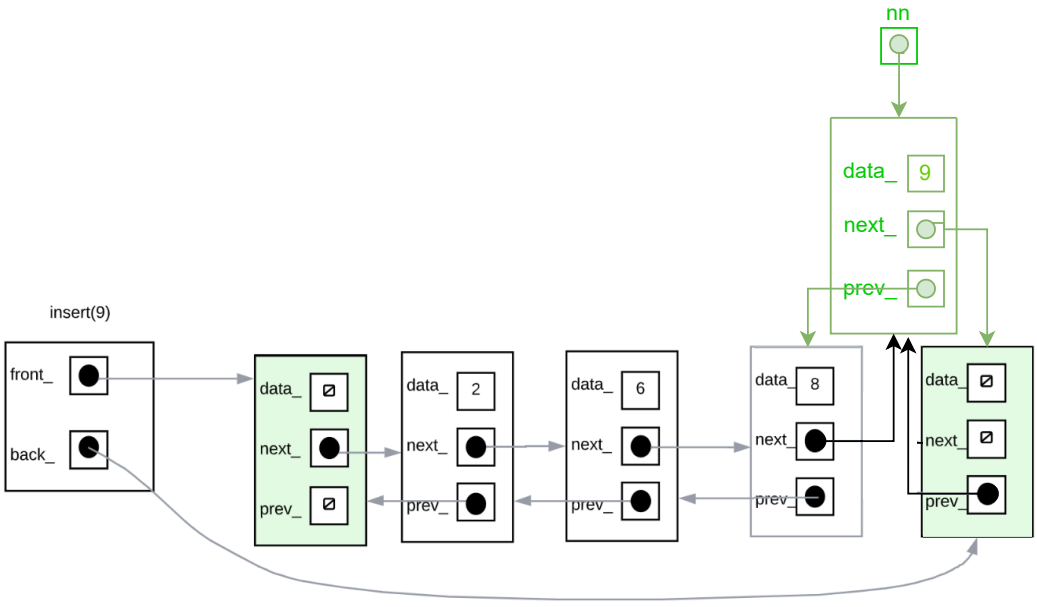
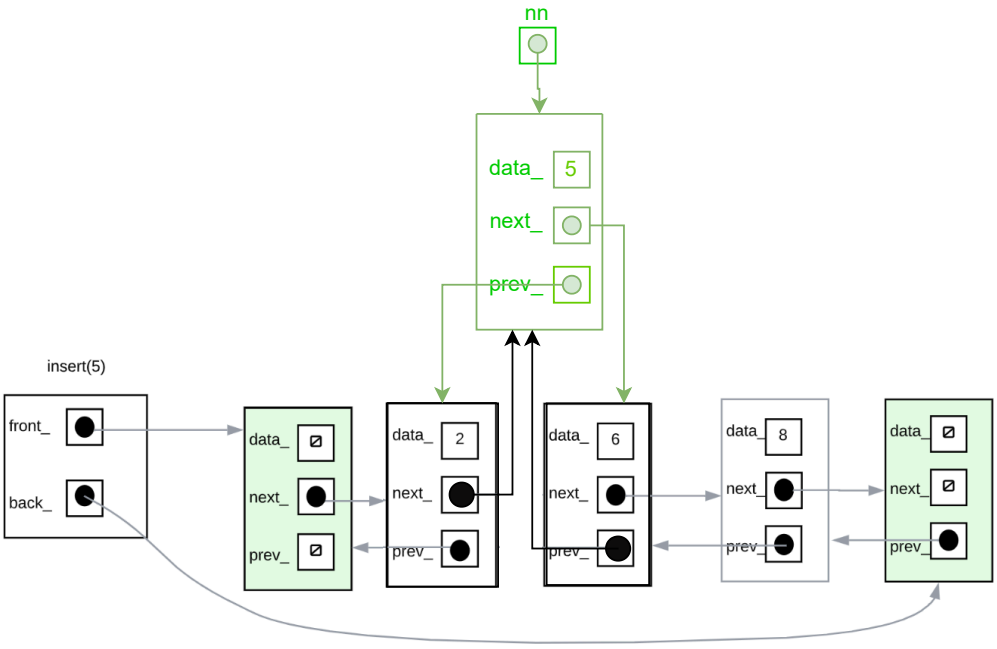
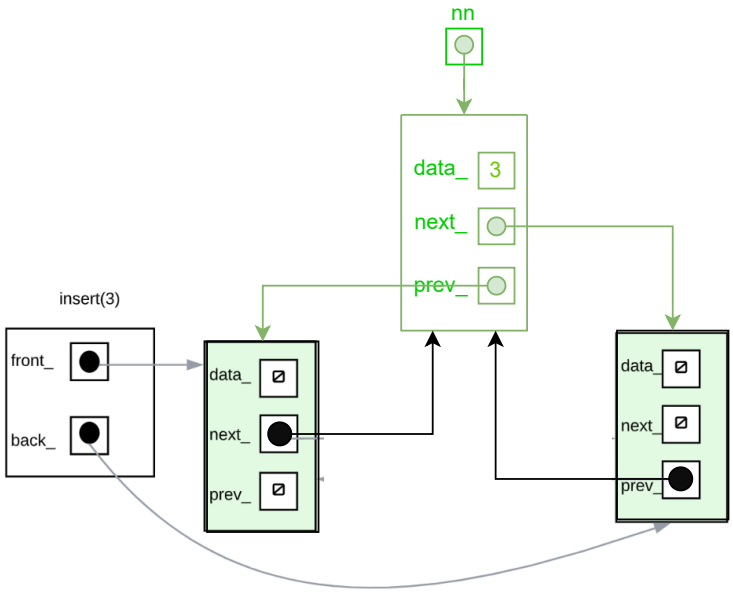

nn
data_ 1
next_
prev_
insert(1)
front_
back_
data_
next_
prev_
data_ 2
next_
prev_
data_ 6
next_
prev_
data_ 8
next_
prev_
data_
next_
prev_

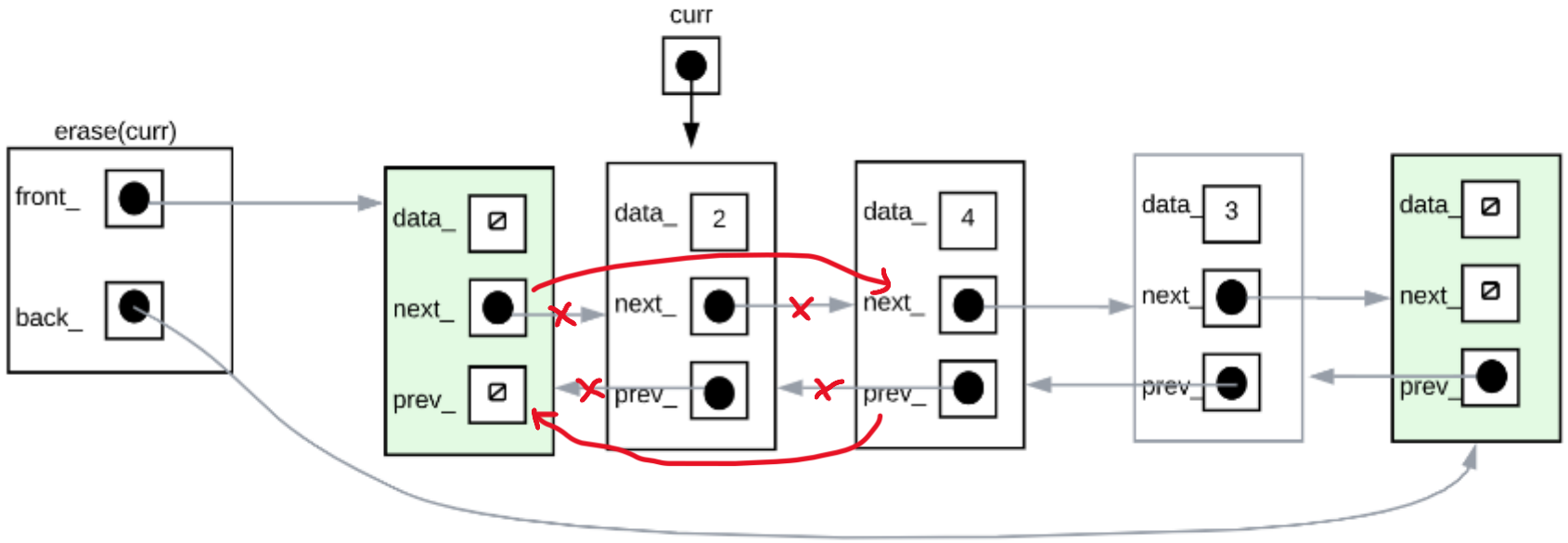

nn
data_ 9
next_
prev_
insert(9)
front_
back_
data_
next_
prev_
data_ 2
next_
prev_
data_ 6
next_
prev_
data_ 8
next_
prev_
data_
next_
prev_

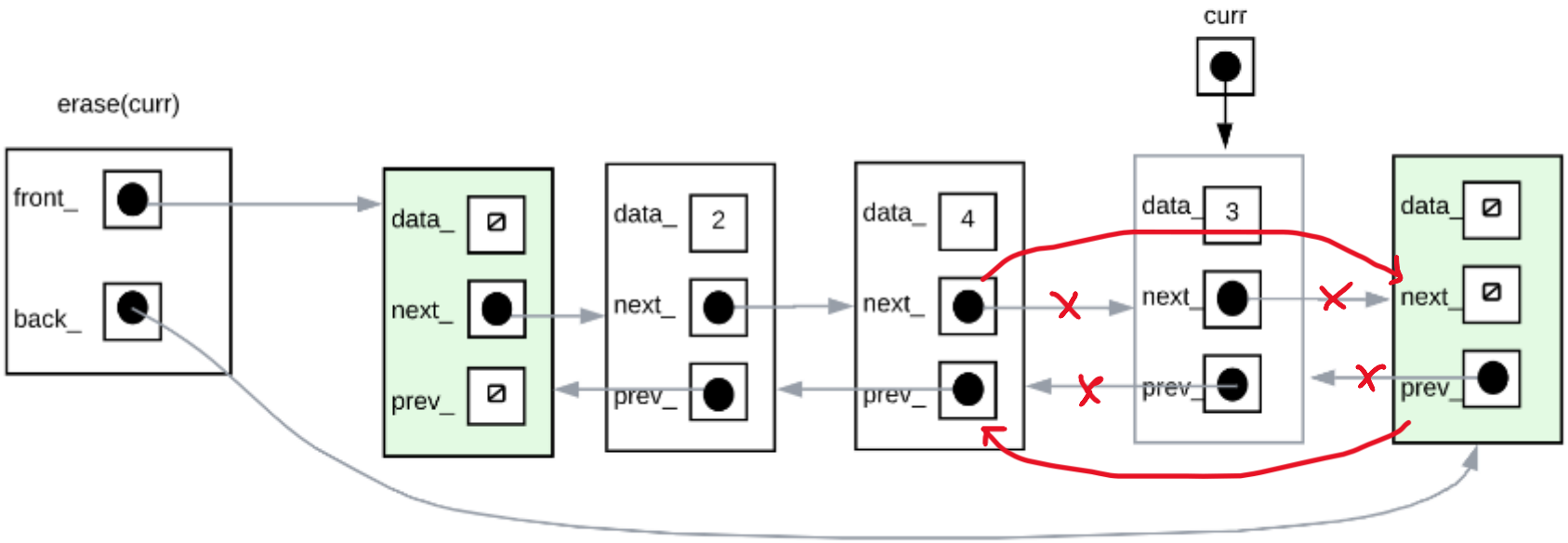

nn
data_ 5
next_
prev_
insert(5)
front_
back_
data_
next_
prev_
data_ 2
next_
prev_
data_ 6
next_
prev_
data_ 8
next_
prev_
data_
next_
prev_

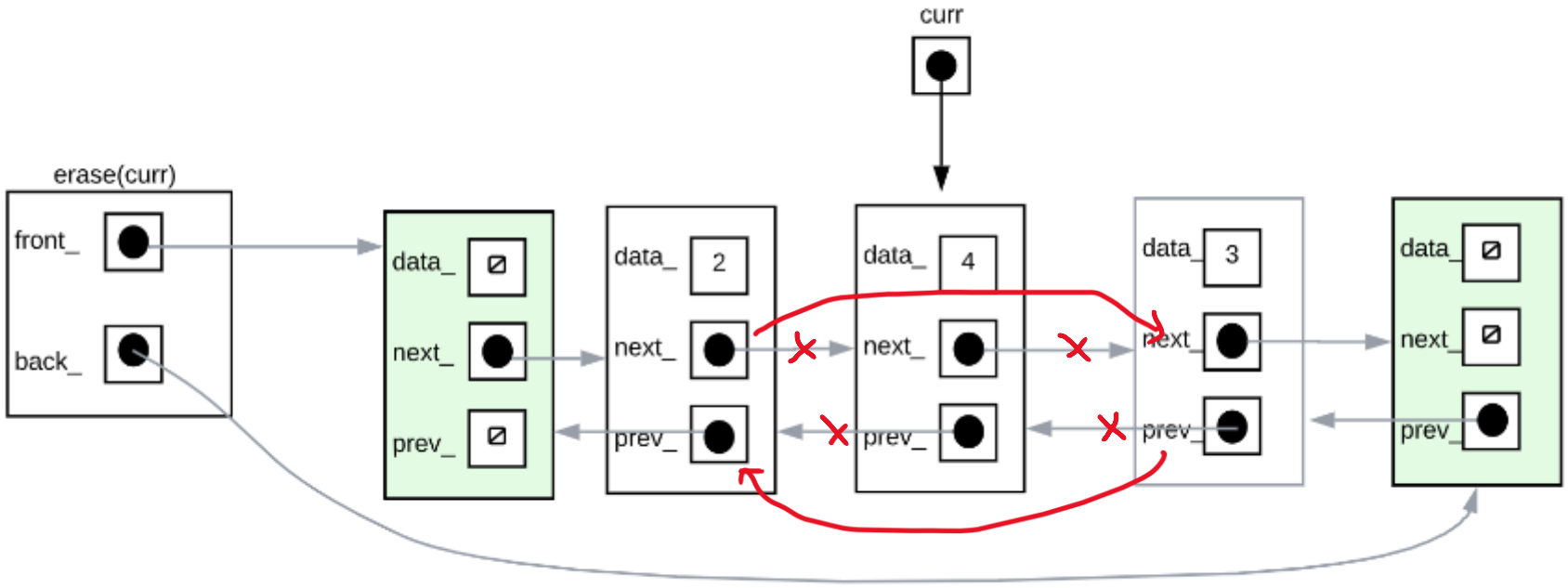

nn
data_ 3
next_
prev_
insert(3)
front_
back_
data_
next_
prev_
data_
next_
prev_

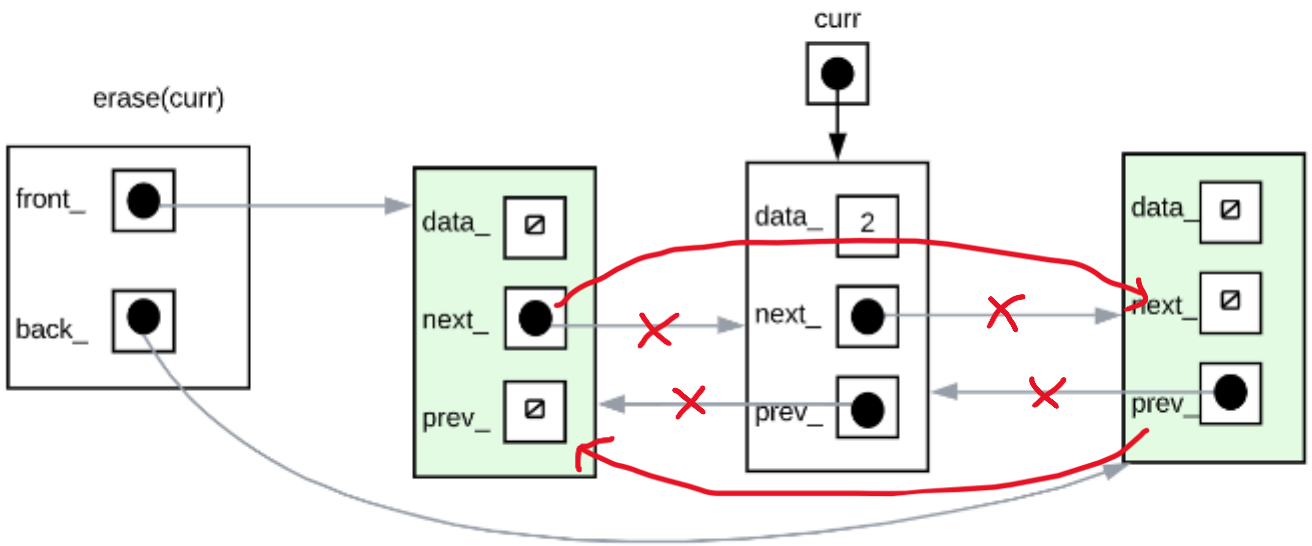
Steps to Remove a Node

1. Update the next_ pointer of the previous node to point to the next node
2. Update the prev_ pointer of the next node to point to the previous node
3. Remove the curr node



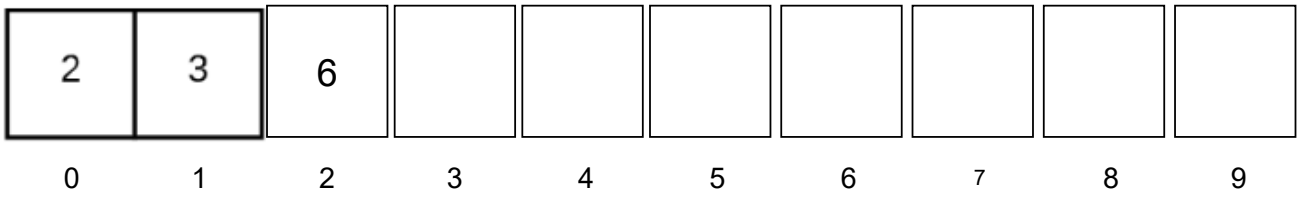

curr
erase(curr)
front_
back_
data_
next_
prev_
data_ 2
next_
prev_
data_ 4
next_
prev_
data_ 3
next_
prev_
data_
next_
prev_


curr
erase(curr)
front_
back_
data_
next_
prev_
data_ 2
next_
prev_
data_ 4
next_
prev_
data_ 3
next_
prev_
data_
next_
prev_


curr
erase(curr)
front_
back_
data_
next_
prev_
data_
2
next_
prev_
data_
next_
prev_

stack.push(6)

3 is at top of stack

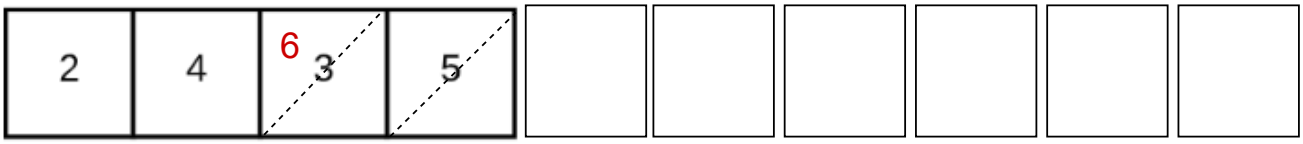


For stack.push(), the data member that needs to be tracked is capacity.

Data Members	Initial State	Operation	State after Operation
capacity	8	stack.push(6)	7

stack.pop()
stack.pop()
stack.push(6)

initially 5 is at top of stack



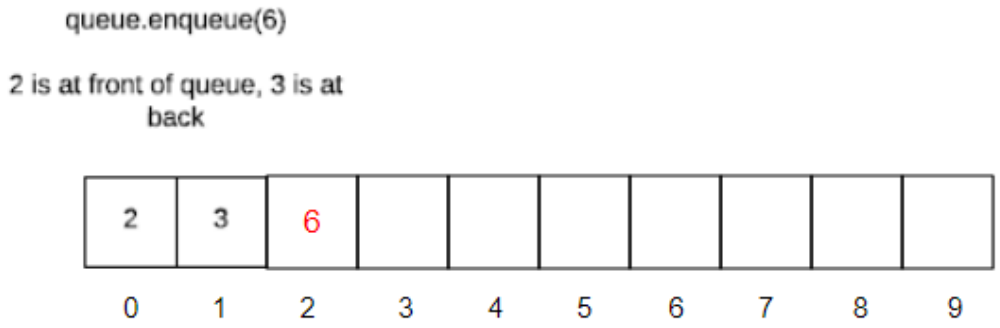
For stack.pop(), the data members that need to be tracked include : list length & top value

Data Members	Initial State	Operation	State after Operation
list length	4	stack.pop()	3
	3	stack.pop()	2
top value	5	stack.pop()	3
	3	stack.pop()	4

For stack.push(), the data member that needs to be tracked is capacity.

Data Members	Initial State	Operation	State after Operation
capacity	8	stack.push(6)	7

1.

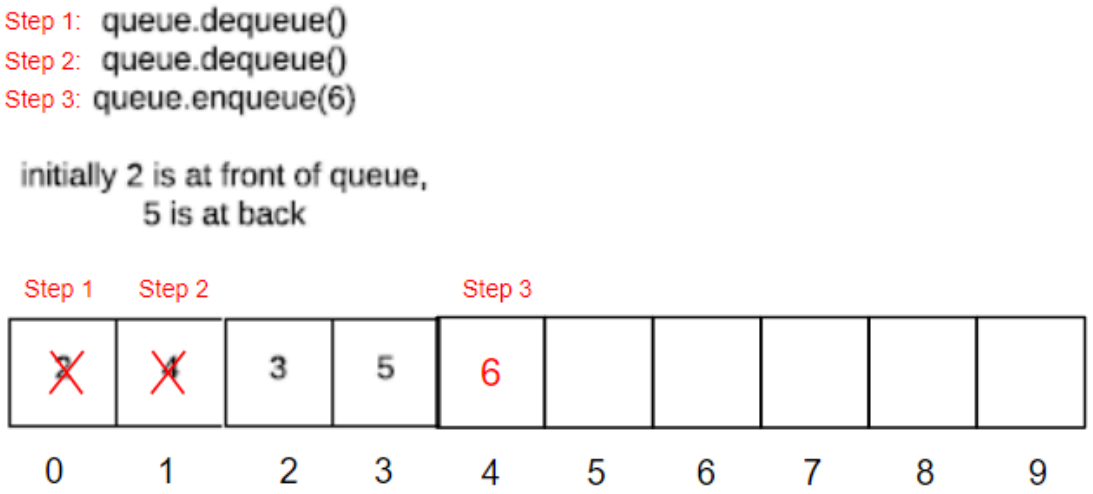


For queue.enqueue(6), we need to track the **rear** and **size**.

The front will not change.

Data Members	Initial State	Operation	Final State	Comments
Front	2	queue.enqueue(6)	2	No changed
Rear	3	queue.enqueue(6)	6	Updated to 6
Size	2	queue.enqueue(6)	3	Increased by 1

2.



Step 1 : queue.dequeue()

For queue.dequeue(), we need to track the **front** and **size.**
The rear will not change.

Data Members	Initial State	Operation	Final State	Comments
Front	2	queue.dequeue()	4	Updated to 4
Rear	5	queue.dequeue()	5	No changed
Size	4	queue.dequeue()	3	Decreased by 1

Step 2 : queue.dequeue()

Data Members	Initial State	Operation	Final State	Comments
Front	4	queue.dequeue()	3	Updated to 3
Rear	5	queue.dequeue()	5	No changed
Size	3	queue.dequeue()	2	Decreased by 1

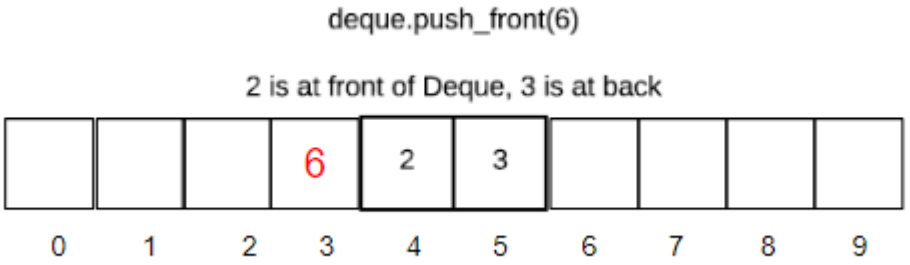
Step 3 : queue.enqueue(6)

For queue.enqueue(6), we need to track the **rear** and **size**.

The front will not change.

Data Members	Initial State	Operation	Final State	Comments
Front	3	queue.enqueue(6)	3	No changed
Rear	5	queue.enqueue(6)	6	Updated to 6
Size	2	queue.enqueue(6)	3	Increased by 1

3.

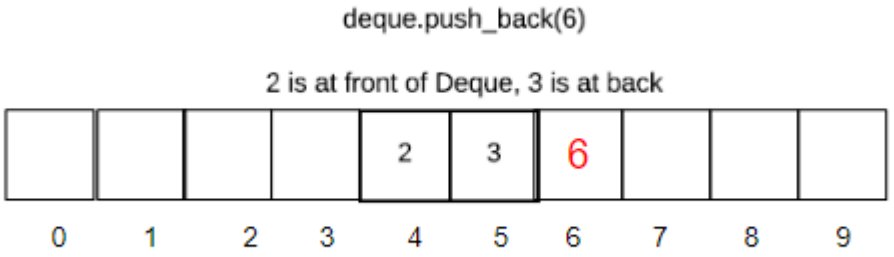


For queue.enqueue(6), we need to track the **front** and **size**.

The rear will not change.

Data Members	Initial State	Operation	Final State	Comments
Front	2	deque.push_front(6)	6	Updated to 6
Rear	3	deque.push_front(6)	3	No changed
Size	2	deque.push_front(6)	3	Increased by 1

4.

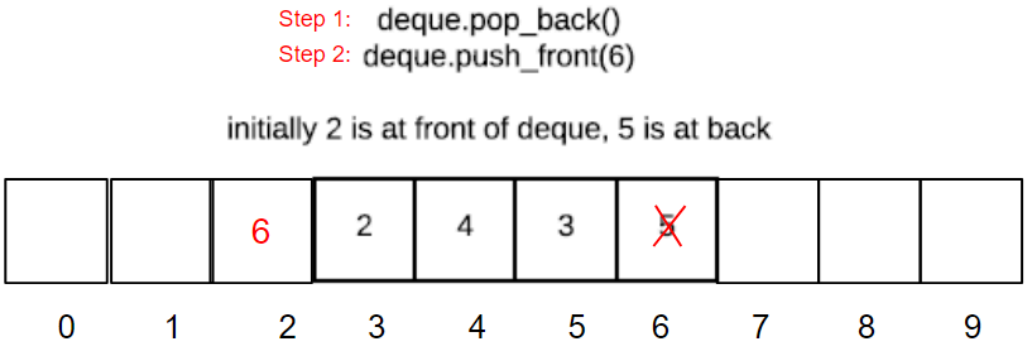


For queue.enqueue(6), we need to track the **rear** and **size**.

The front will not change.

Data Members	Initial State	Operation	Final State	Comments
Front	2	deque.push_back(6)	2	No changed
Rear	3	deque.push_back(6)	6	Updated to 6
Size	2	deque.push_back(6)	3	Increased by 1

5.



Step 1 : deque.pop_back()

For deque.pop_back(), we need to track the **rear** and **size.**
The front will not change.

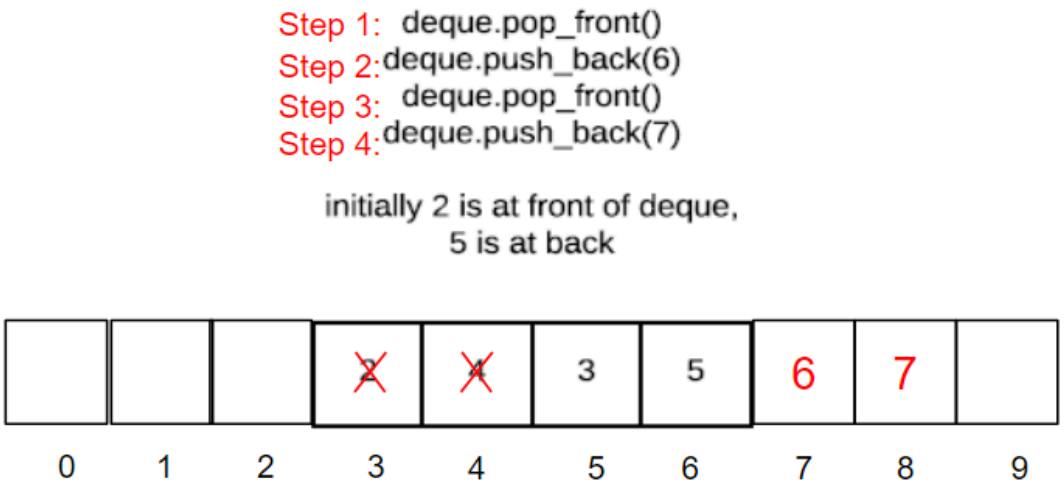
Data Members	Initial State	Operation	Final State	Comments
Front	2	deque.pop_back()	2	No changed
Rear	5	deque.pop_back()	3	Updated to 3
Size	4	deque.pop_back()	3	Decreased by 1

Step 2 : deque.push_front(6)

For deque.push_front(6), we need to track the **front** and **size.**
The rear will not change.

Data Members	Initial State	Operation	Final State	Comments
Front	2	deque.push_front(6)	6	Updated to 6
Rear	3	deque.push_front(6)	3	No changed
Size	3	deque.push_front(6)	4	Increased by 1

6.



Step 1 : deque.pop_front()

For deque.pop_back(), we need to track the **front** and **size.**
The rear will not change.

Data Members	Initial State	Operation	Final State	Comments
Front	2	deque.pop_front()	4	Updated to 4
Rear	5	deque.pop_front()	5	No changed
Size	4	deque.pop_front()	3	Decreased by 1

Step 2 : deque.push_back(6)

For deque.push_front(6), we need to track the **rear** and **size.**
The front will not change.

Data Members	Initial State	Operation	Final State	Comments
Front	4	deque.push_back(6)	4	No changed
Rear	5	deque.push_back(6)	6	Increased by 1
Size	3	deque.push_back(6)	4	Increased by 1

Step 3 : deque.pop_front()

For deque.pop_back(), we need to track the **front** and **size.**
The rear will not change.

Data Members	Initial State	Operation	Final State	Comments
Front	4	deque.pop_front()	3	Updated to 3
Rear	6	deque.pop_front()	6	No changed
Size	4	deque.pop_front()	3	Decreased by 1

Step 4 : deque.push_back(7)

For deque.push_back(7), we need to track the **rear** and **size.**
The front will not change.

Data Members	Initial State	Operation	Final State	Comments
Front	3	deque.push_back(7)	3	No changed
Rear	6	deque.push_back(7)	7	Updated to 6
Size	3	deque.push_back(7)	4	Increased by 1

Step 1

-2	1	-3	-3	0
2	0	3	2	0
0	0	-3	0	0
0	0	1	0	0

Step 2

0	-3	1	1	-1
-3	0	-4	-3	0
0	0	-3	0	0
0	0	1	0	0

Step 1 to 2 :

Overflowing cells:

- **(0,0)**: The value is -2, with 2 neighbors, meeting the overflow condition becoming 0.
- **(0,2)**: The value is -3, with 3 neighbors, meeting the overflow condition and was affected by (0,3), becoming 1.
- **(0,3)**: The value is -3, with 3 neighbors, meeting the overflow condition and was affected by (0,2), becoming 1.

Neighbor updates:

- **(0,1)**: it increases by 2, and its sign changes to match theirs, becoming -3.
- **(1,0)**: it increases by 1, and its sign changes to match (0,0), becoming -3.
- **(1,2)**: it increases by 1,and its sign changes to match theirs becoming -4.
- **(1,3)**: it increases by 1, and its sign changes to match theirs becoming -3.
- **(0,4)**: it increases by 1, and its sign changes to match (0,3), becoming -1.

Step 2

0	-3	1	1	-1
-3	0	-4	-3	0
0	0	-3	0	0
0	0	1	0	0

Step 2 to 3 :

Overflowing cells:

- **(1,0)**: The value is -3, with 3 neighbors, meeting the overflow condition, becoming 0.
- **(0,1)**: The value is -3, with 3 neighbors, meeting the overflow condition, becoming 0.
- **(1,2)**: The value is -4, with 4 neighbors, meeting the overflow condition,, becoming 0.

Step 3

-2	0	-3	1	-1
0	-3	0	-4	0
-1	0	-4	0	0
0	0	1	0	0

Neighbor updates:

- **(0,0)**:it increases by 2, and its sign changes to match theirs, becoming -2.
- **(2,0)**: it increases by 1, and its sign changes becoming -1.
- **(1,1)**: it increases by 3, and its sign changes to match theirs, becoming -3.
- **(0,2)**: it increases by 2, and its sign changes to match theirs, becoming -3.
- **(2,2)**: it increases by 1,becoming -4.
- **(1,3)**: it increases by 1,becoming -4.

Step 3

-2	0	-3	1	-1
0	-3	0	-4	0
-1	0	-4	0	0
0	0	1	0	0

Step 4

0	-2	0	-3	-1
-1	-3	-3	0	-1
-1	-1	0	-2	0
0	0	-2	0	0

Step 3 to 4 :

Overflowing cells:

- **(0,0)**: The value is -2, with 2 neighbors, meeting the overflow condition, becoming 0.
- **(0,2)**: The value is -3, with 3 neighbors, meeting the overflow condition, becoming 0.
- **(1,3)**: The value is -4, with 4 neighbors, meeting the overflow condition,, becoming 0.
- **(2,2)**: The value is -4, with 4 neighbors, meeting the overflow condition,, becoming 0.

Neighbor updates:

- **(1,0)**: it increases by 1, and its sign changes becoming -1.
- **(0,1)**: it increases by 2, becoming -2.
- **(2,1)**:it increases by 1, and its sign changes becoming -1.
- **(1,2)**: it increases by 3, and its sign changes to match theirs, becoming -3.
- **(3,2)**: it increases by 1, and its sign changes becoming -2.
- **(0,3)**: it increases by 2, becoming -3.
- **(2,3)**: it increases by 2,and its sign changes becoming becoming -2.
- **(1,4)**: it increases by 2, and its sign changes becoming becoming -1.

Step 4 (Last Step)

0	-2	0	-3	-1
-1	-3	-3	0	-1
-1	-1	0	-2	0
0	0	-2	0	0

The grid contains cells with all the same signs, and the process has already stopped

The return value is 3.