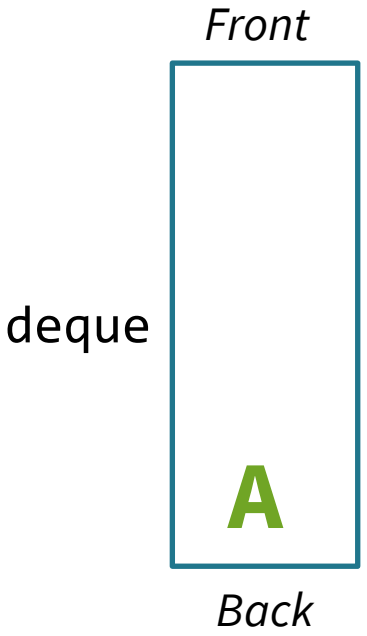
DEQUE HELP SLIDES

ADD A

Since the deque is empty, set the root to be the new node

```
root = A
```

Add the node to the back of the deque.



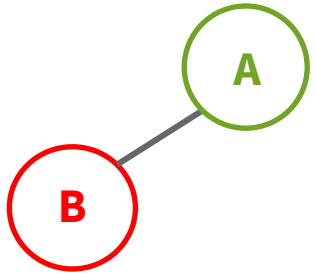
ADD B

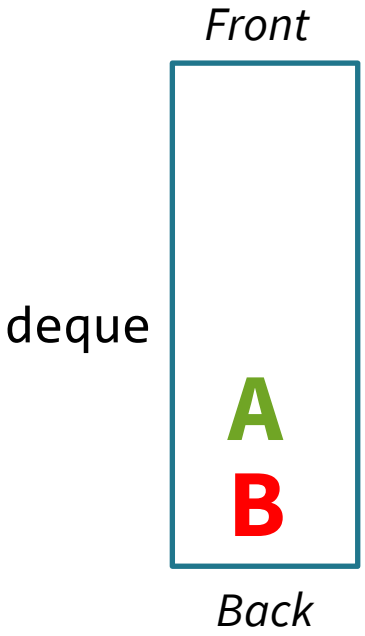
Look in the front of the deque to find the parent that needs a child added to it next

```
parent = deque.front()
```

If the parent doesn't have a left child yet, add the node as a left child.

Add the node to the back of the deque.





ADD C

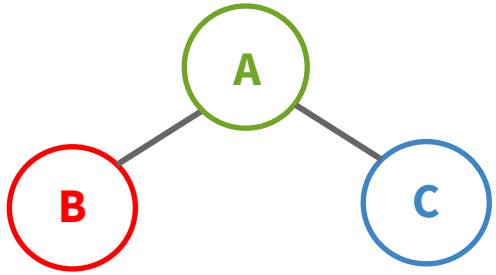
Look in the front of the deque to find the parent that needs a child added to it next

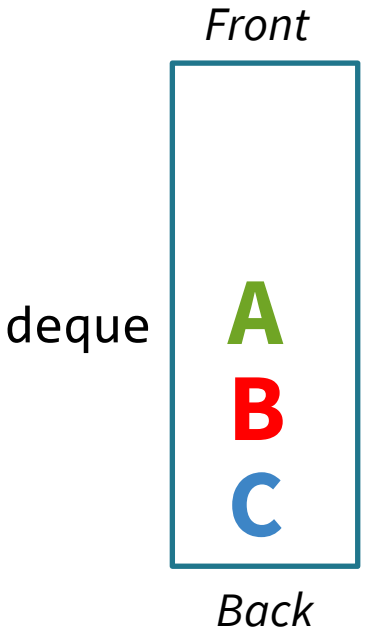
```
parent = deque.front()
```

If the parent already has a left child, add the node as a right child.

Add the node to the back of the deque.

A has no space for more children, so remove from the front of the deque.





ADD C

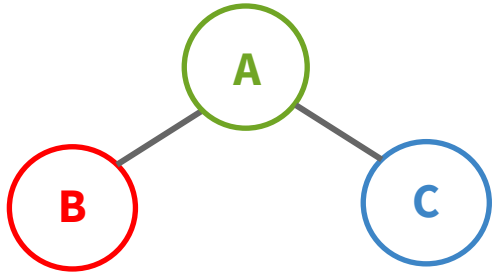
Look in the front of the deque to find the parent that needs a child added to it next

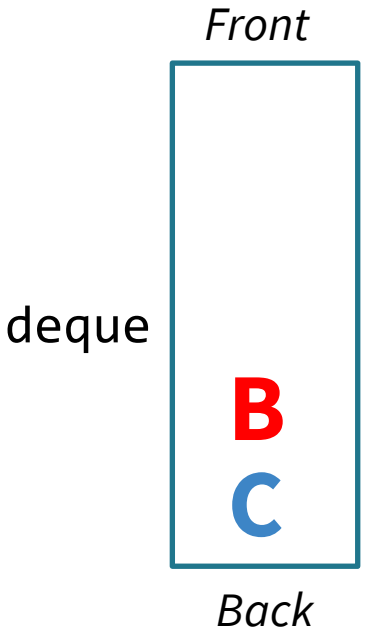
```
parent = deque.front()
```

If the parent already has a left child, add the node as a right child.

Add the node to the back of the deque.

A has no space for more children, so remove from the front of the deque.

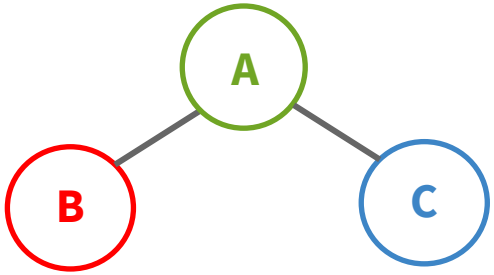


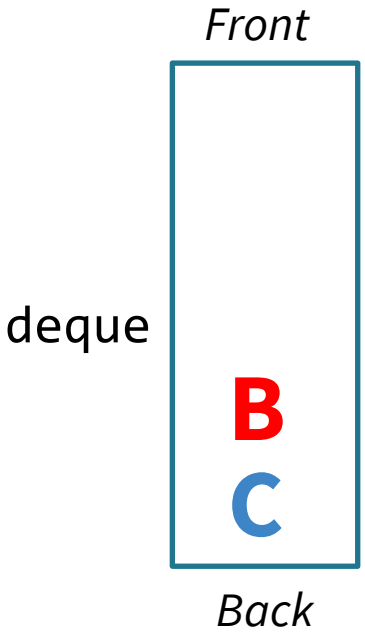


DELETE

Look in the back of the deque to find the node that needs to be removed next

```
node = deque.back()
```





Get the node's parent

```
parent = node.parent()
```

If the parent has a right child, remove the parent's right child

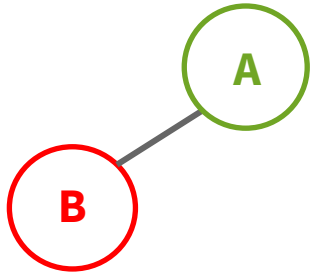
Remove the node from the back of the deque.

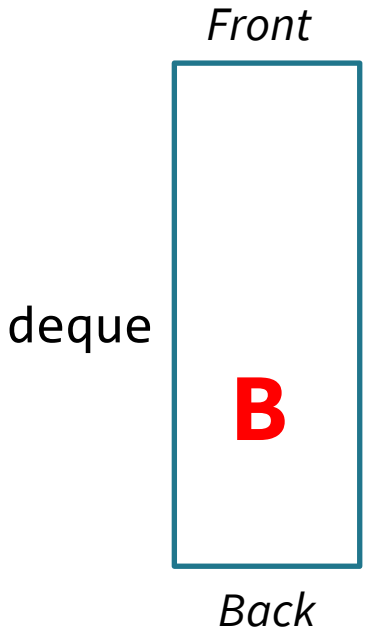
Now **A** has space for more children, so add **A** to the front of the deque.

DELETE

Look in the back of the deque to find the node that needs to be removed next

```
node = deque.back()
```





Get the node's parent

```
parent = node.parent()
```

If the parent has a right child, remove the parent's right child

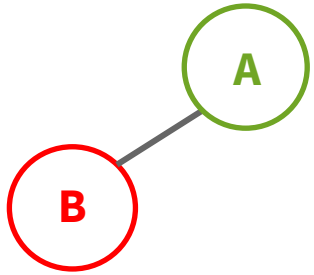
Remove the node from the back of the deque.

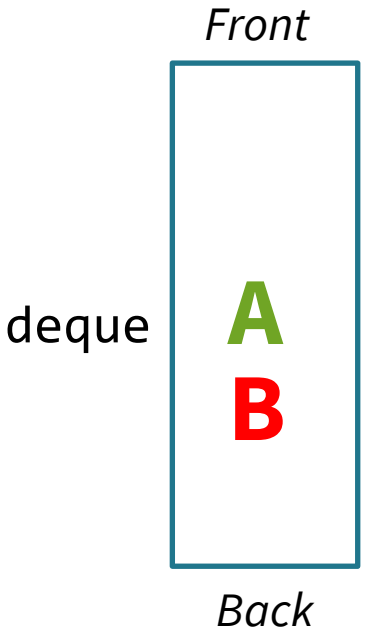
Now **A** has space for more children, so add **A** to the front of the deque.

DELETE

Look in the back of the deque to find the node that needs to be removed next

```
node = deque.back()
```





Get the node's parent

```
parent = node.parent()
```

If the parent has a right child, remove the parent's right child

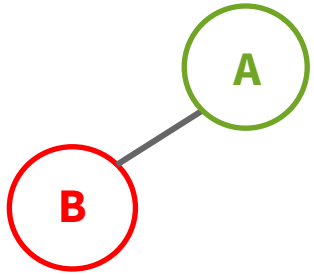
Remove the node from the back of the deque.

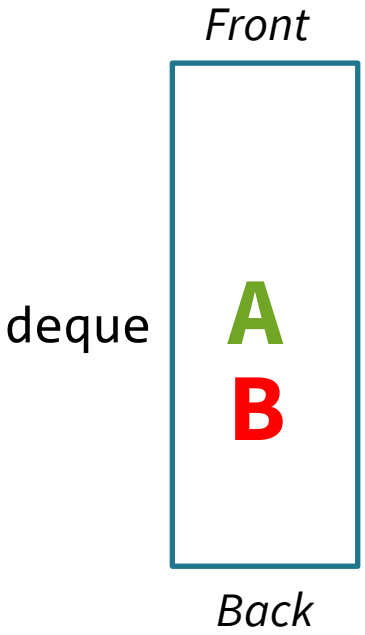
Now **A** has space for more children, so add **A** to the front of the deque.

DELETE

Look in the back of the deque to find the node that needs to be removed next

```
node = deque.back()
```





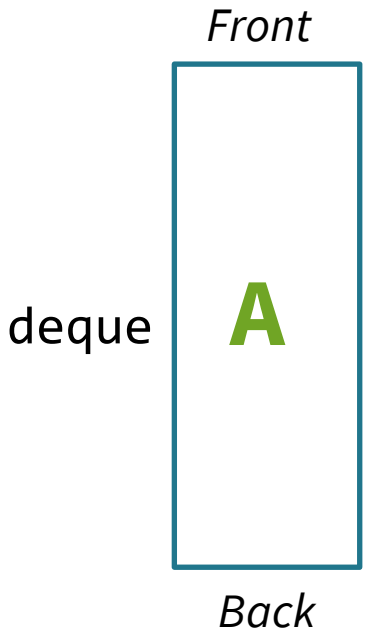
Get the node's parent

```
parent = node.parent()
```

If the parent has no right child, remove the parent's left child.

Remove the node from the back of the deque.

DELETE



Look in the back of the deque to find the node that needs to be removed next

```
node = deque.back()
```

Get the node's parent

```
parent = node.parent()
```

If the parent has no right child, remove the parent's left child.

Remove the node from the back of the deque.

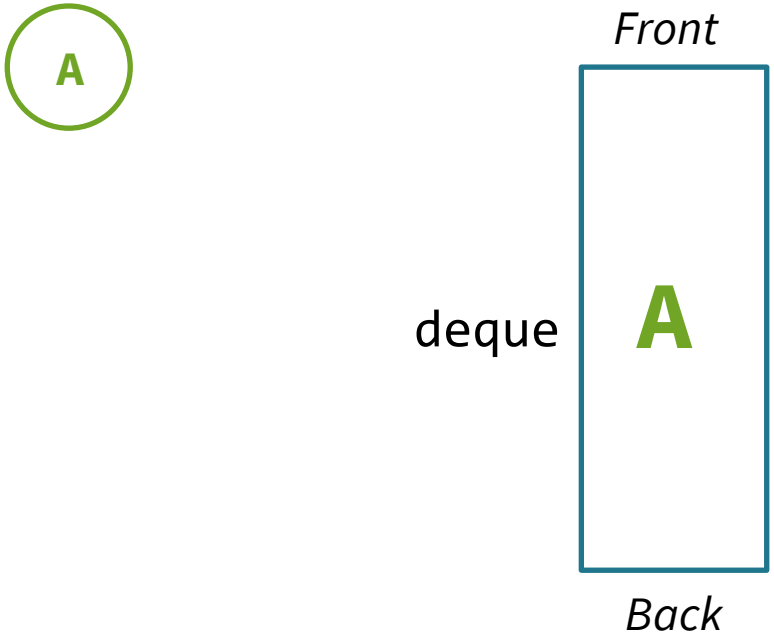
DELETE

Look in the back of the deque to find the node that needs to be removed next and pop it from the deque

```
node = deque.popBack()
```

If the deque is empty, set the root to null.

```
root = null
```



DELETE

Look in the back of the deque to find the node that needs to be removed next and pop it from the deque

```
node = deque.popBack()
```

If the deque is empty, set the root to null.

```
root = null
```