Your Ultimate Guide To Landing
Top AI roles

DECODE
AiML

Introduction to Deque

→ Data structure — Linear

Front ⟶ | | 5 | 10 | 15 | 20 | | ← Rear

Linear

→ data structure is a Concept. It describes a way of storing and managing data in a Computer So that it can be accessed and modified efficiently.

→ Deque allows us to add and remove elements from both ends — the front and rear efficiently.

→ In Python, we implement Queue data structures using Collections. deque

* **Goal of Deque Data Structures**

→ Fast Insertion and deletion from both the ends.

Collections.deque Internals

→ In Python, deque (from Collections) is implemented as a doubly linked list of fixed size blocks.

NULL ← [5] ⇄ [10] ⇄ [15] ⇄ [20] → NULL

Head ↗

→ Doubly-Linked list is implemented in C Style. Each block is a struct.

[| |] ← Each block is implemented as Structure

→ Fast O(1) Push and Pop from both ends

Operations in Deque

Operation	Meaning	T(n)
Append right	insert at rear end	$O(1)$
Append Left	insert at front end	$O(1)$
Pop Right	Remove from rear	$O(1)$
Pop Left	Remove from front	$O(1)$
Peek front	View front item	$O(1)$
Peek Rear	view Rear item	$O(1)$
is Empty	Check if deque empty?	$O(1)$
Size	Check len of deque	$O(1)$

Example: Let's check an example

① appendL(5) ② appendL(2) ③ appendR(-1) ④ peekR() ⑤ PopR()

| 5 |

| 2 | 5 |

| 2 | 5 | -1 |

return $\boxed{-1}$

| 2 | 5 |

⑥ isEmpty()   ⑦ PopR()   ⑧ size()   ⑨ appendR(4)

return $\boxed{False}$

| 2 |

return ①

| 2 | 4 |

Deque using Collection.deque

→ <u>Initialization</u> ← O(1)

```
from Collections import deque
dq = deque()
```

→ <u>Append Right</u> ← O(1)

```
dq.append(x)
```

→ <u>Append Left</u> ← O(1)

```
dq.appendleft(x)
```

→ Pop right ← O(1)

```
if len( dq )>0:
    dq.pop()
```

→ Pop Left ← O(1)

```
if len( dq )>0:
    dq.popleft()
```

→ Peek right ← O(1)

```
if len( dq )>0:
    return dq[-1]
```

→ Peek Left ← O(1)

```
if len( dq )>0:
    return dq[0]
```

→ $\underline{\text{IsEmpty}}$ ← O(1)

```
return len(dq )==0
```

→ $\underline{\text{Size}}$ ← O(1)

```
return len( dq )
```