Linked list

11	3	23	7
0	1	2	3

11

3

23

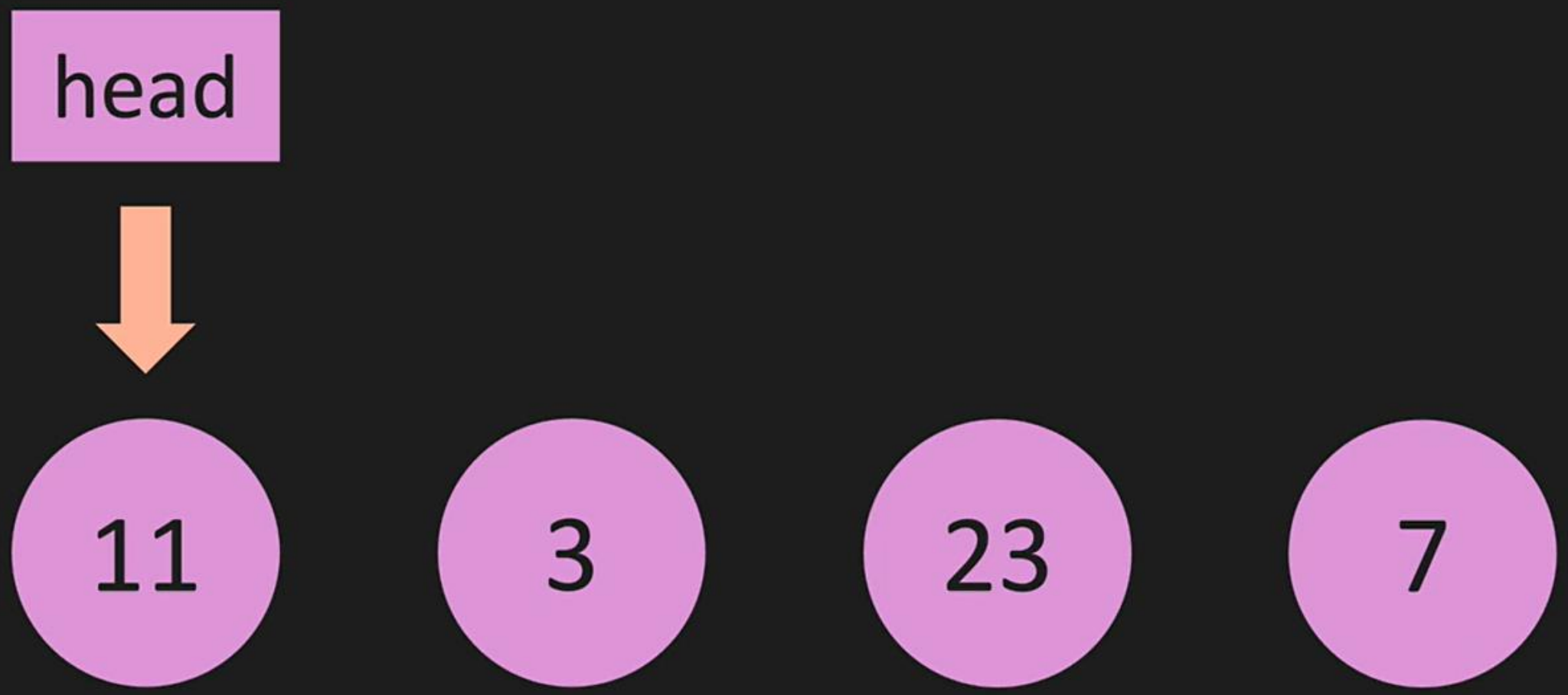
7

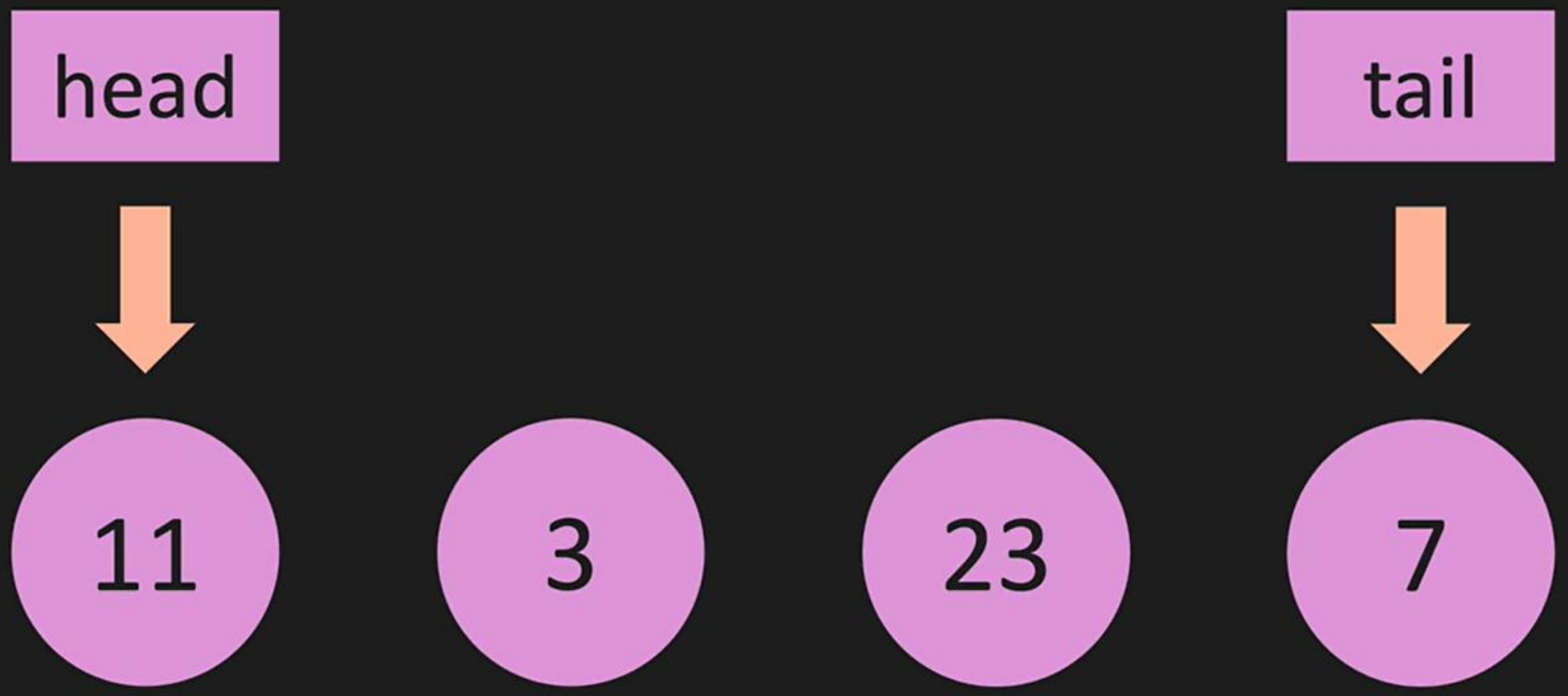
11

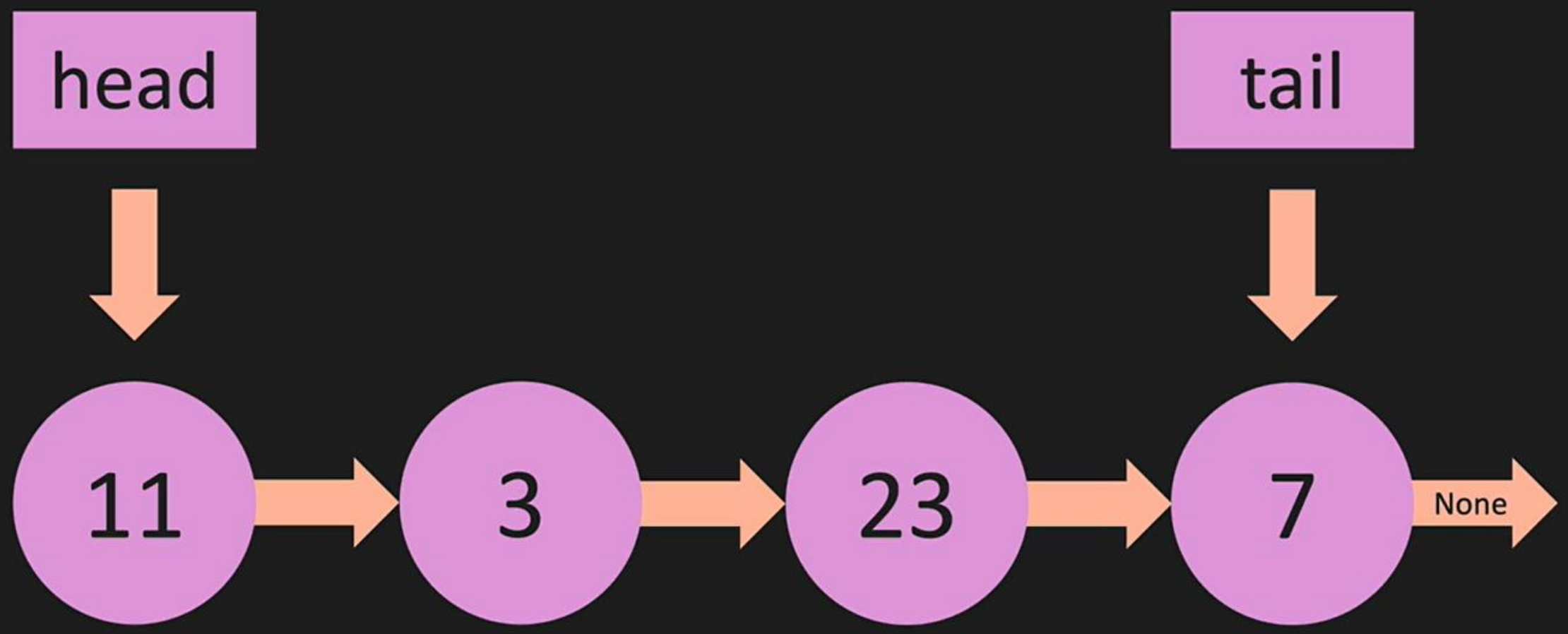
3

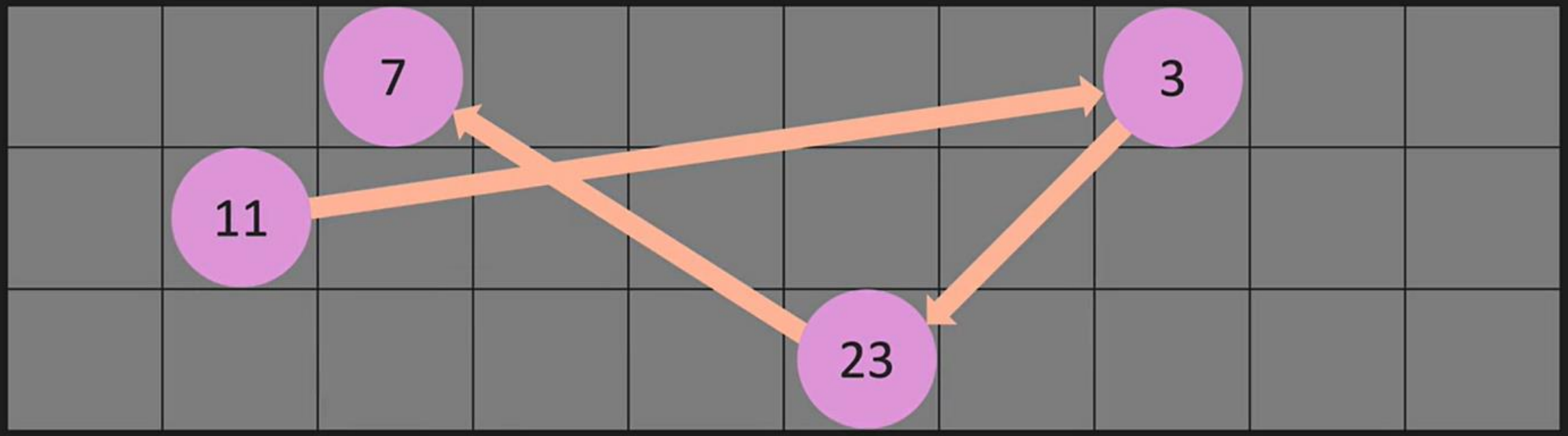
23

7


head
11
3
23
7

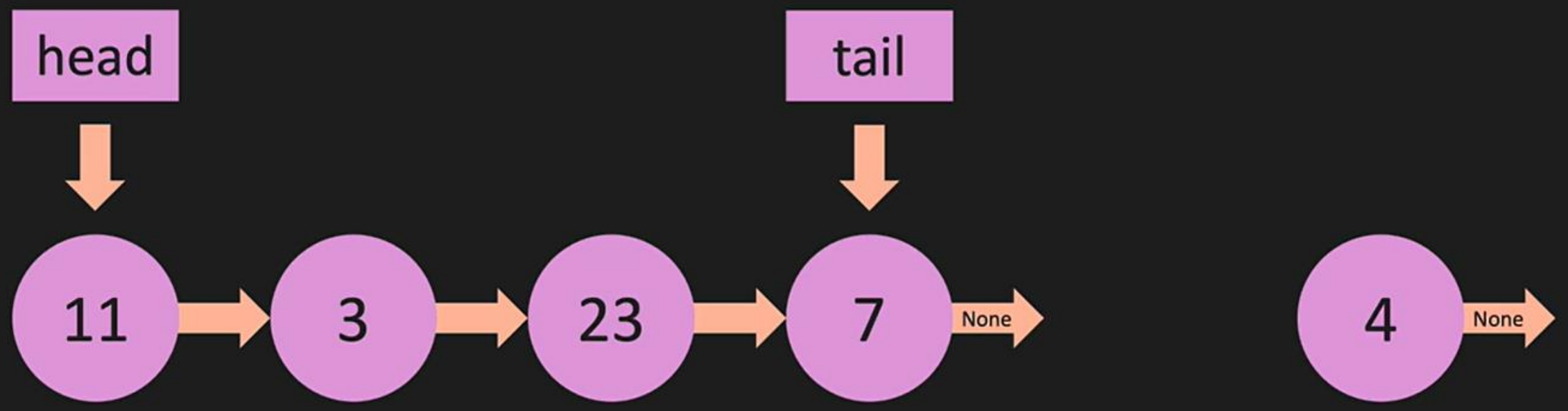

head
tail
11
3
23
7


head
tail
11
3
23
7
None

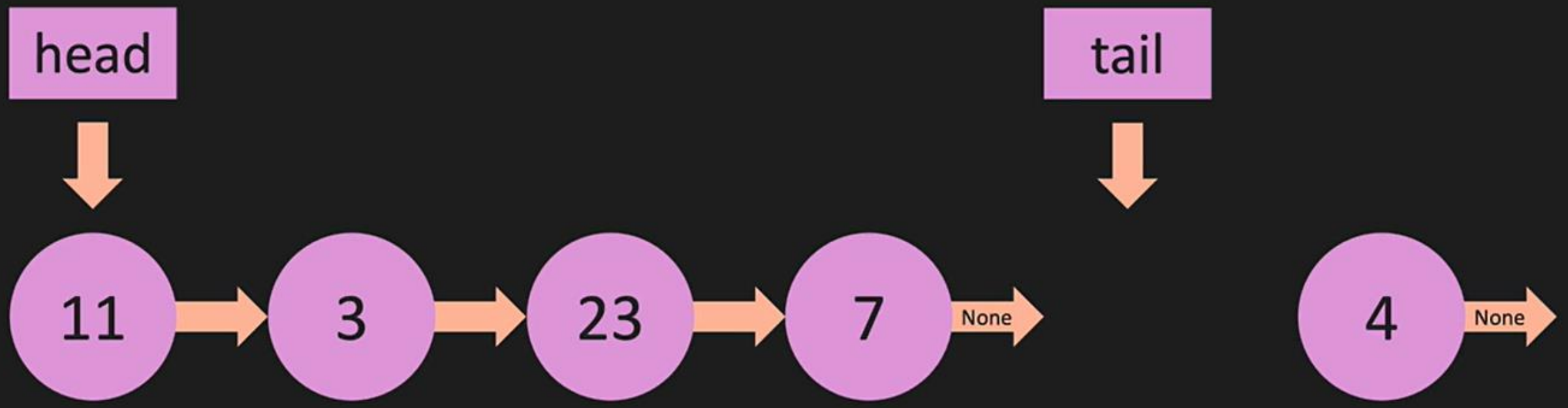

7
3
11
23

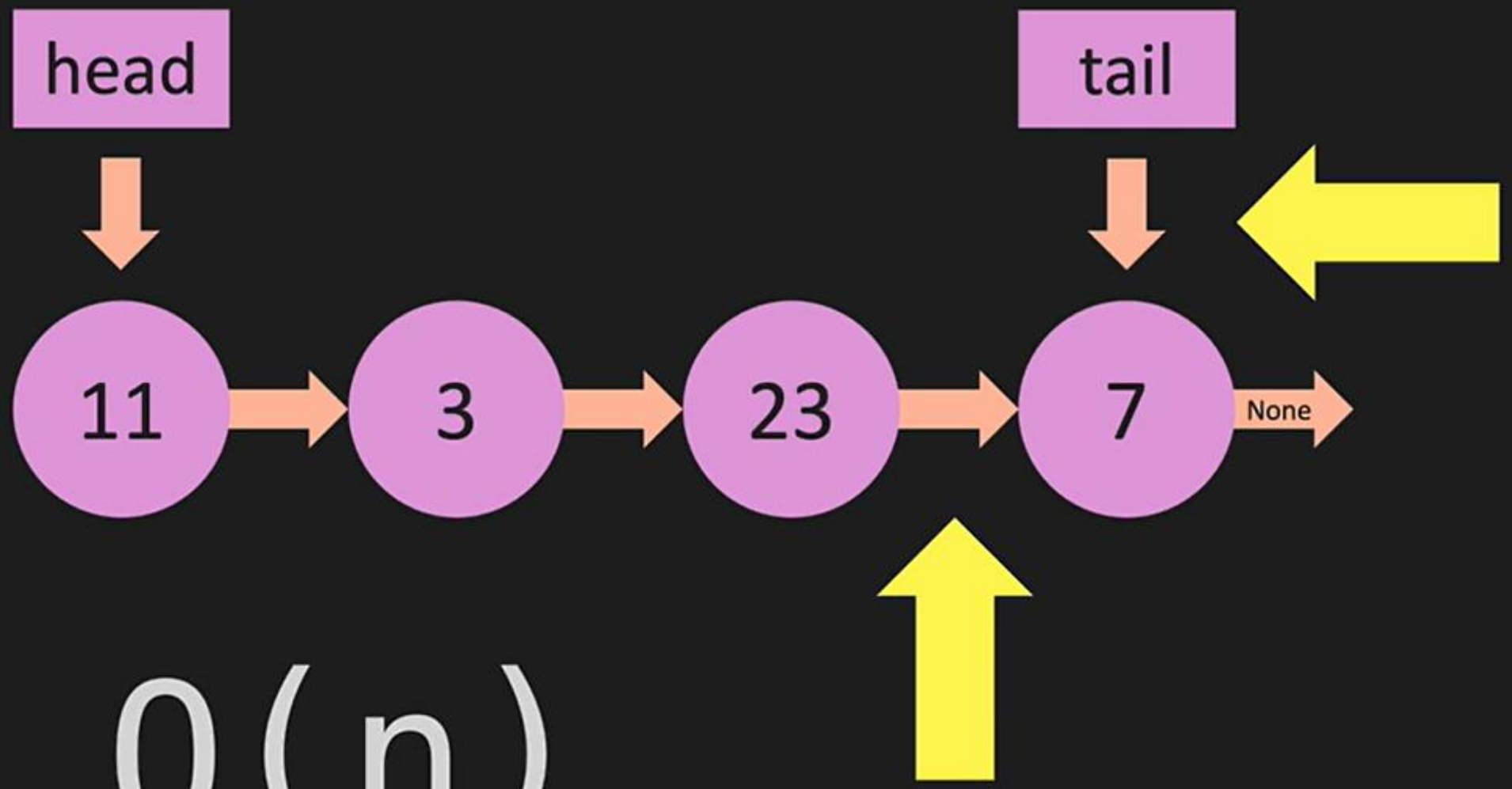
			11	3	23	7			

11	3	23	7
0	1	2	3

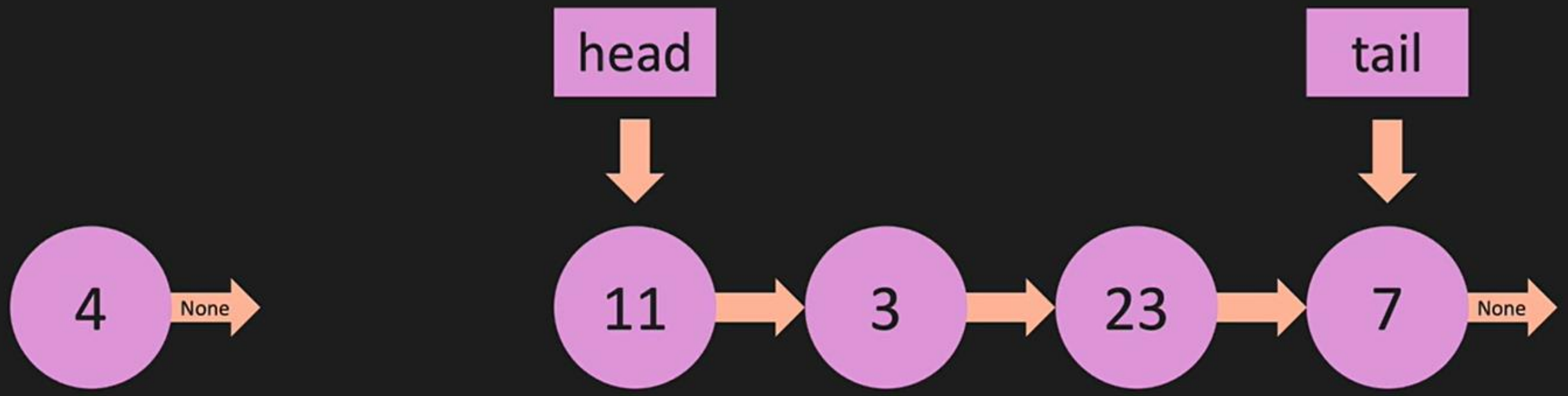


O(1)

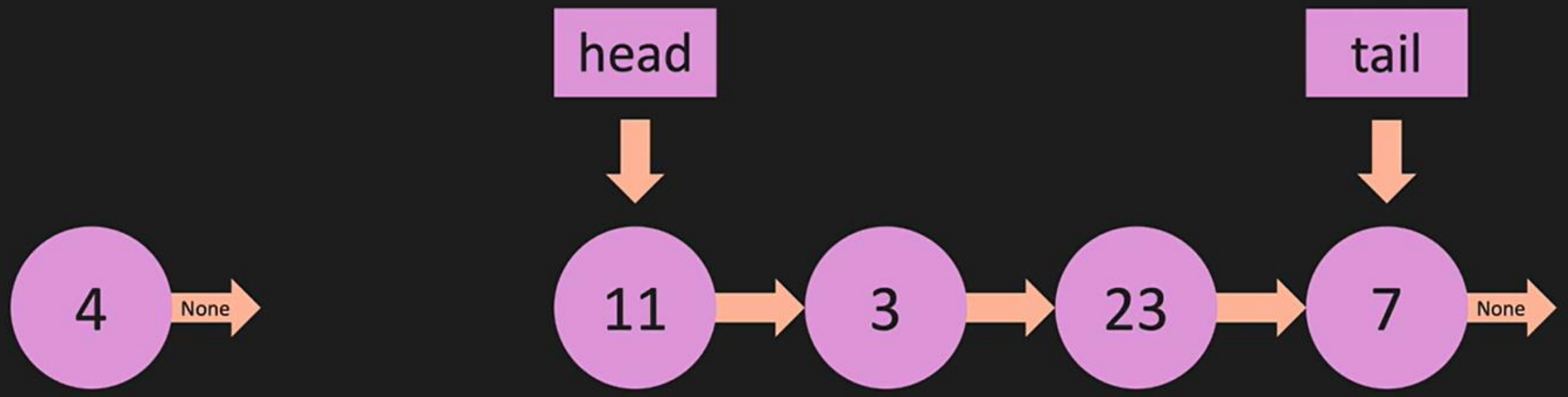

head
tail
11
3
23
7
None
4
None



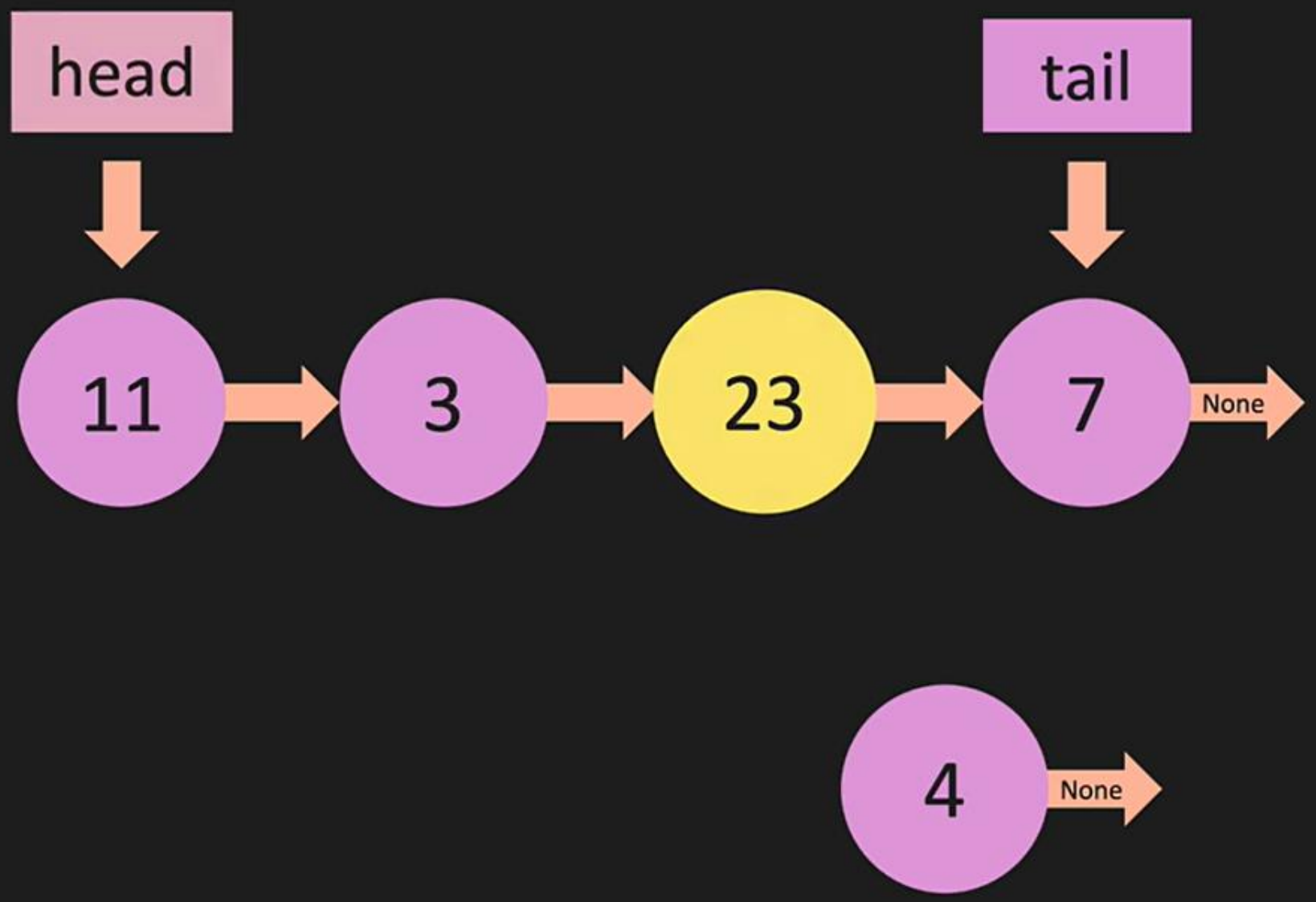
O(n)

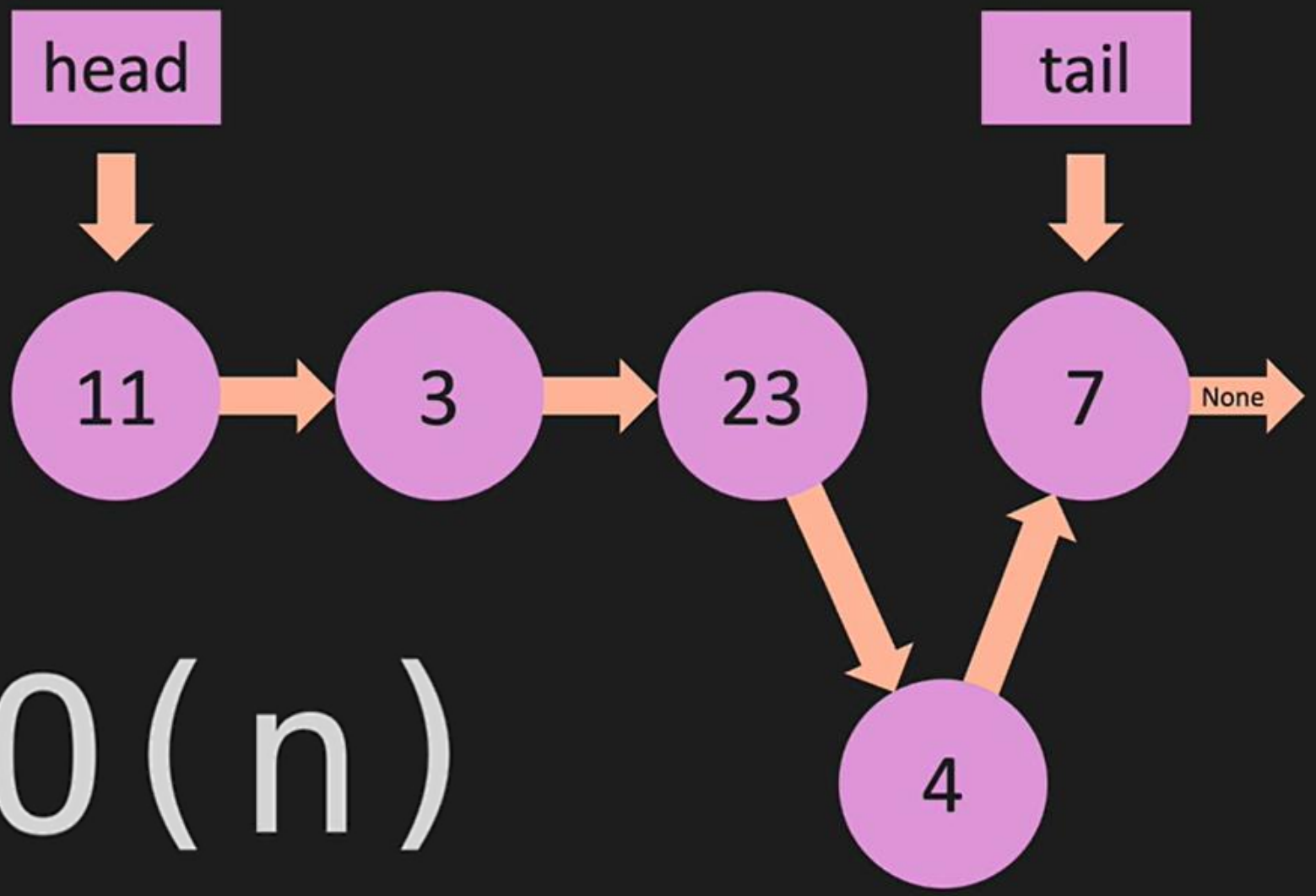


O(1)

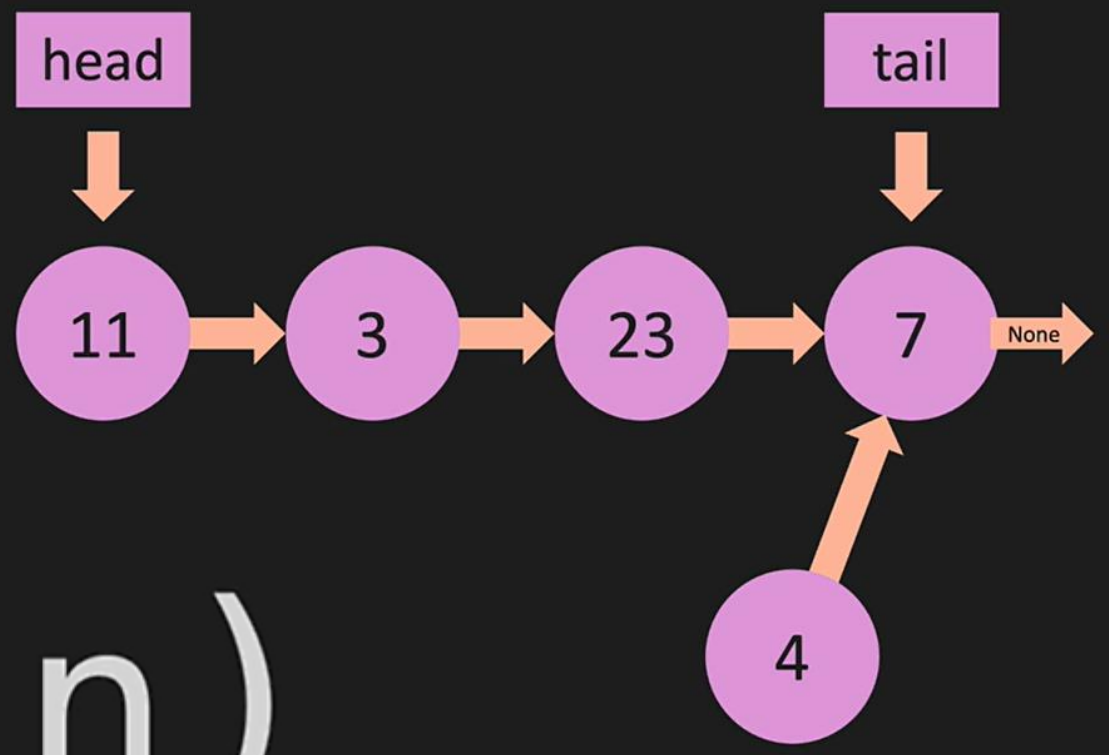


O(1)

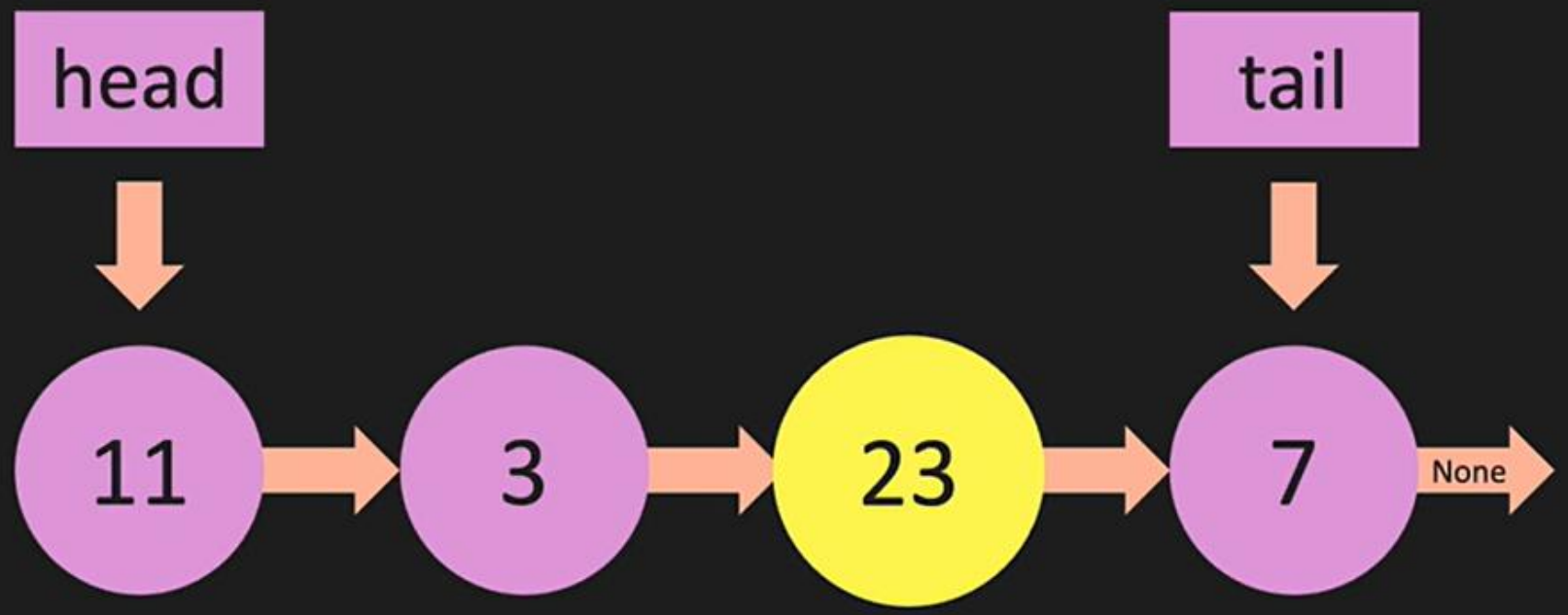

head
tail
11
3
23
7
None
4
None



O(n)

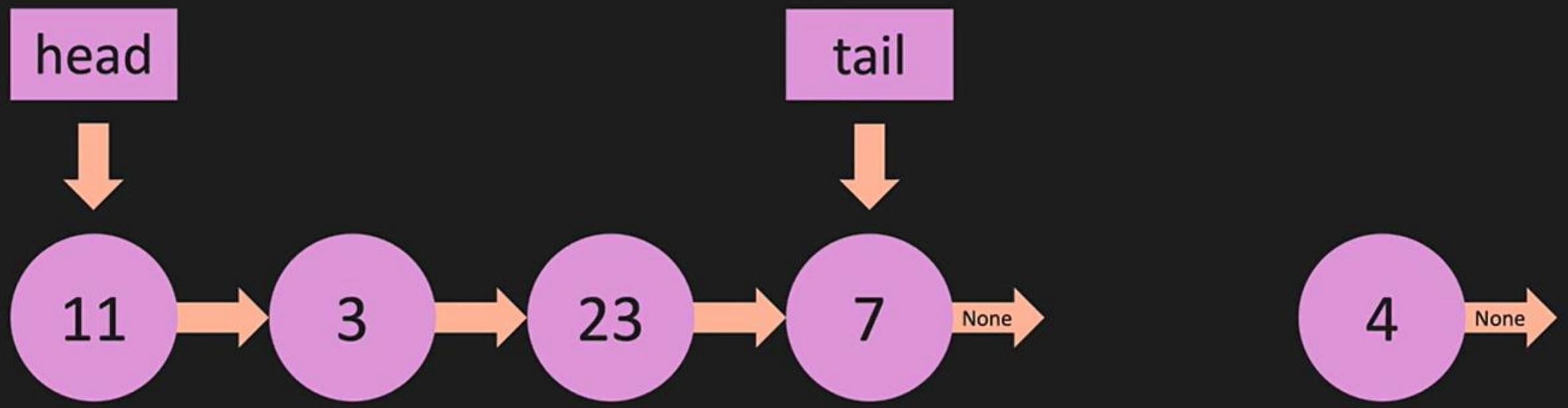


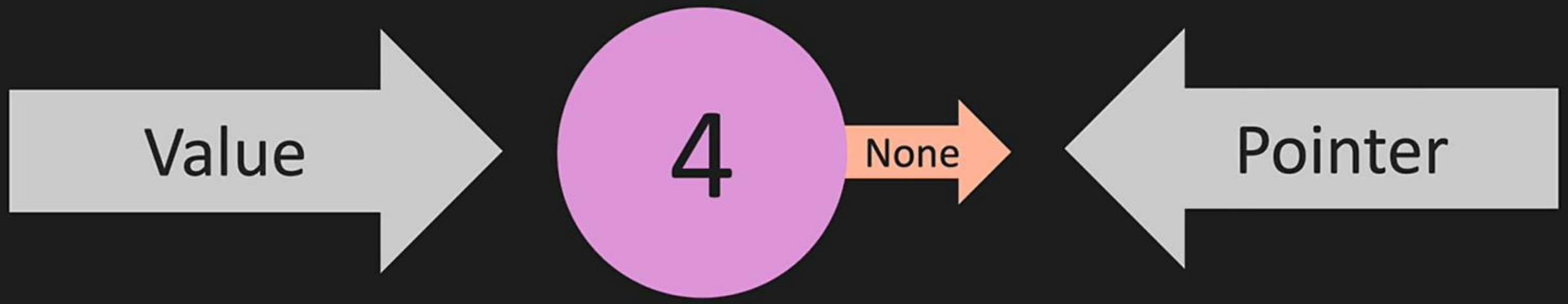
O(n)



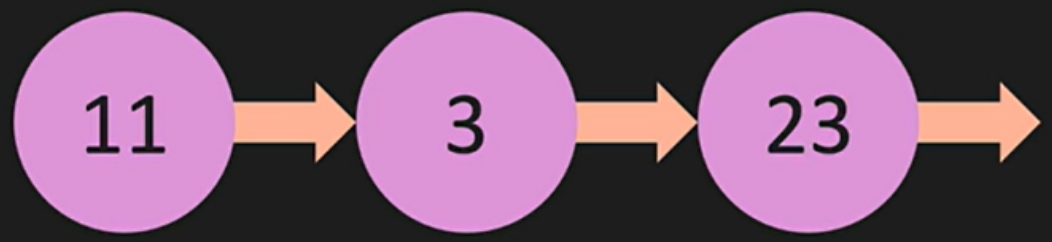
O(n)

	Linked Lists	Lists
Append	O(1)	O(1)
Pop	O(n)	O(1)
Prepend	O(1)	O(n)
Pop First	O(1)	O(n)
Insert	O(n)	O(n)
Remove	O(n)	O(n)
Lookup by Index	O(n)	O(1)
Lookup by Value	O(n)	O(n)


head
tail
11
3
23
7
None
4
None

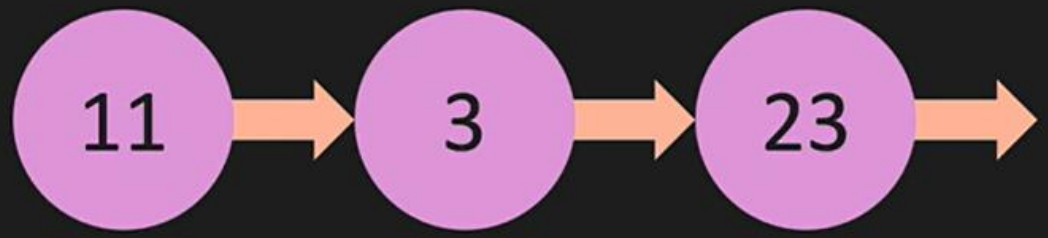

Value
4
None
Pointer

```
{
 "value": 4,
 "next": None
}
```




```
{
 "value": 7,
 "next":
}
```

```
{
 "value": 4,
 "next": None
}
```




```
{
 "value": 7,
 "next": {
          "value": 4,
          "next": None
         }
}
```

```
{
 "value": 11,
 "next": {
          "value": 3,
          "next": {
                   "value": 23,
                   "next": {
                            "value": 7,
                            "next": {
                                     "value": 4,
                                     "next": None
                                    }
                           }
                  }
          }
}
```

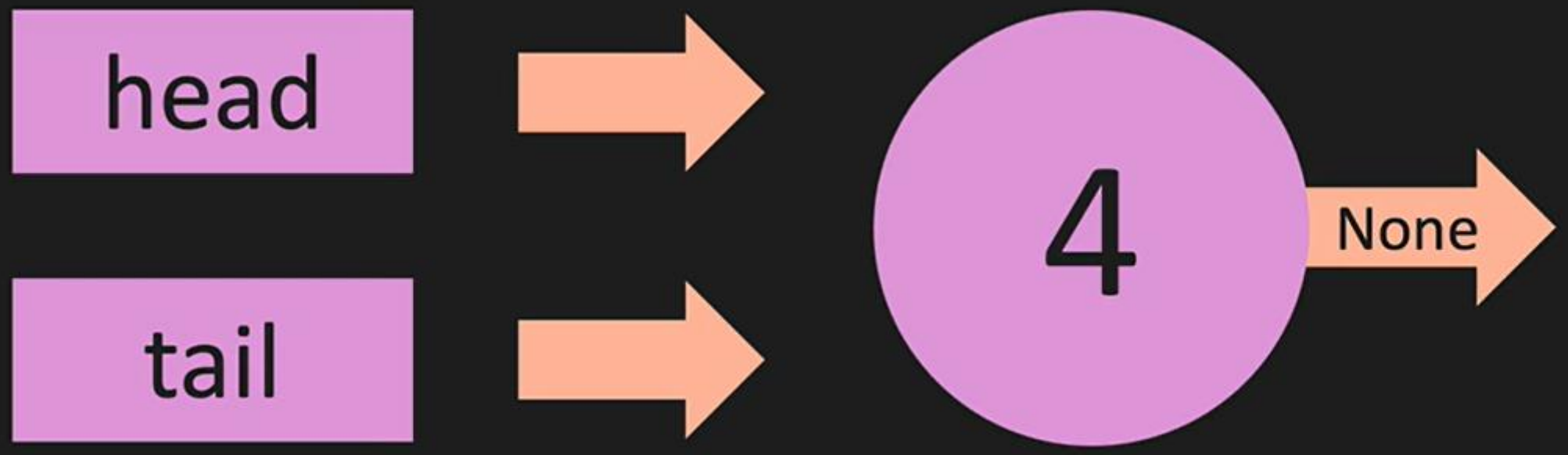
```python
class LinkedList:
    def __init__(self, value):
    def append(self, value):
    def prepend(self, value):
    def insert(self, index, value):
```

```
class LinkedList:
    def __init__(self, value):
        create new Node
    def append(self, value):
        create new Node
        add Node to end
    def prepend(self, value):
        create new Node
        add Node to beginning
    def insert(self, index, value):
        create new Node
        insert Node
```

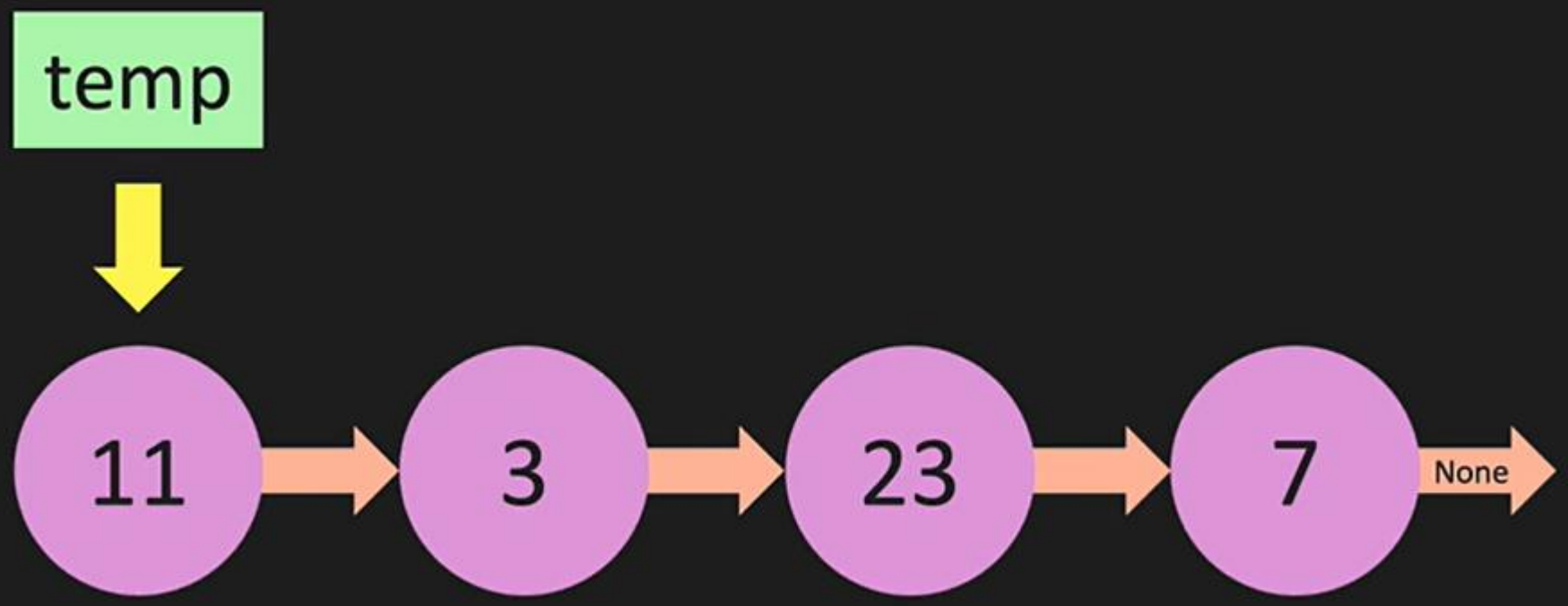
```python
class Node:
    def __init__(self, value):
        self.value = value
        self.next = None
```

```
{
 "value": 4,
 "next": None
}
```

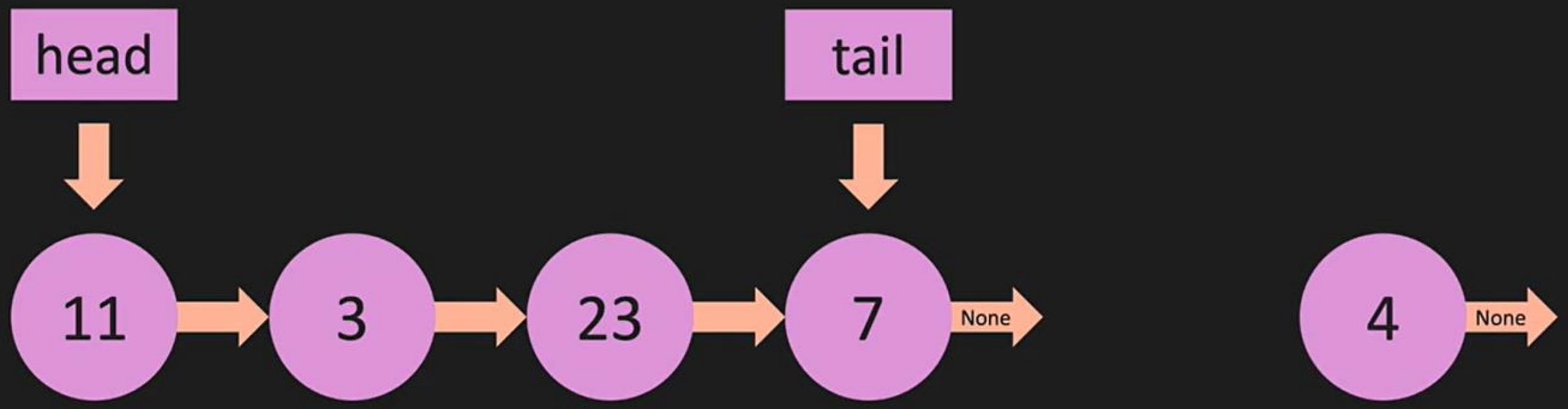
```python
class LinkedList:
    def __init__(self, value):
        new_node = Node(value)
        self.head = new_node
        self.tail = new_node
        self.length = 1
```

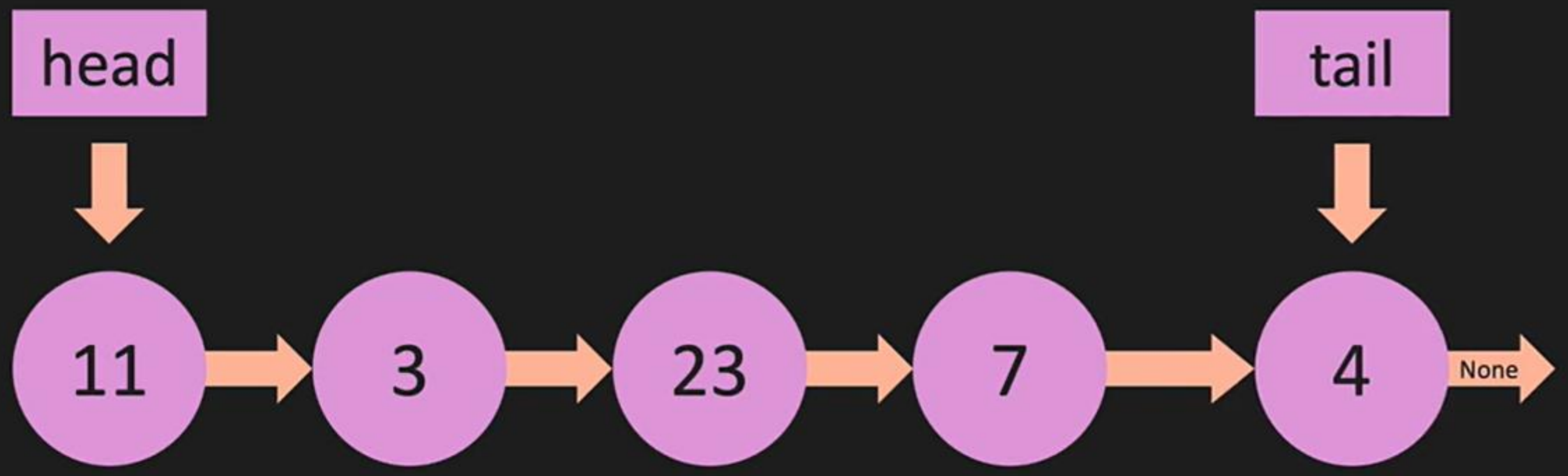


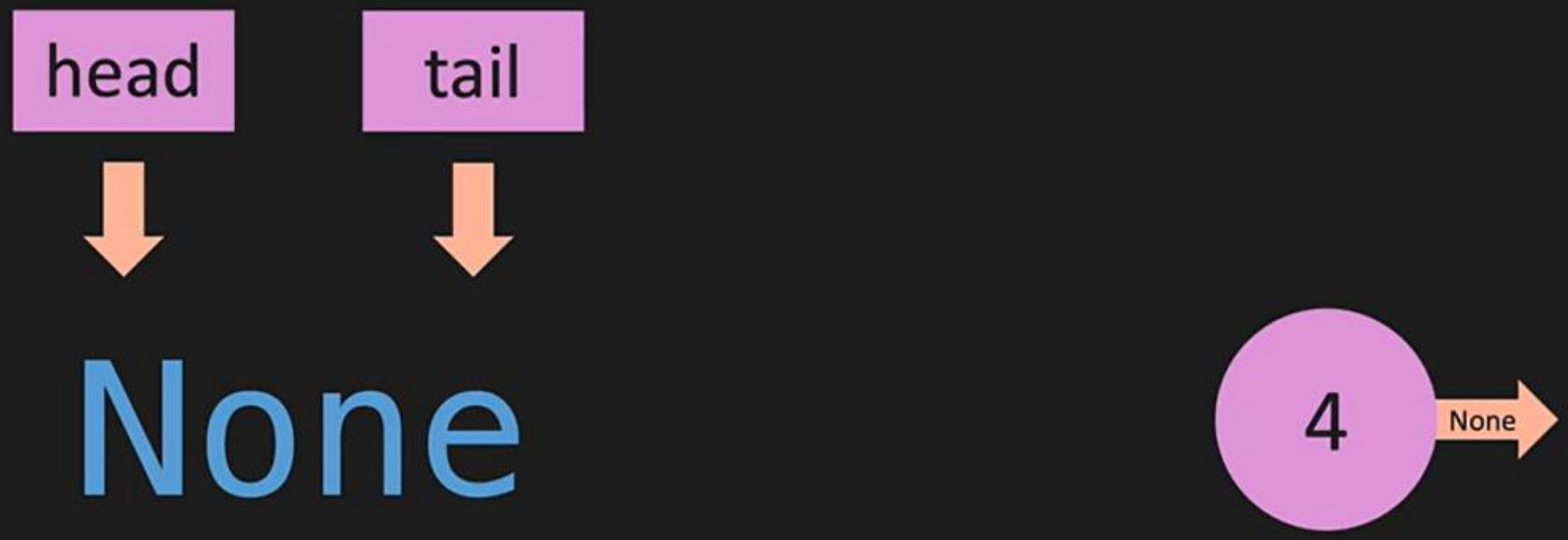
```python
def print_list(self):
    temp = self.head
    while temp is not None:
        print(temp.value)
        temp = temp.next
```



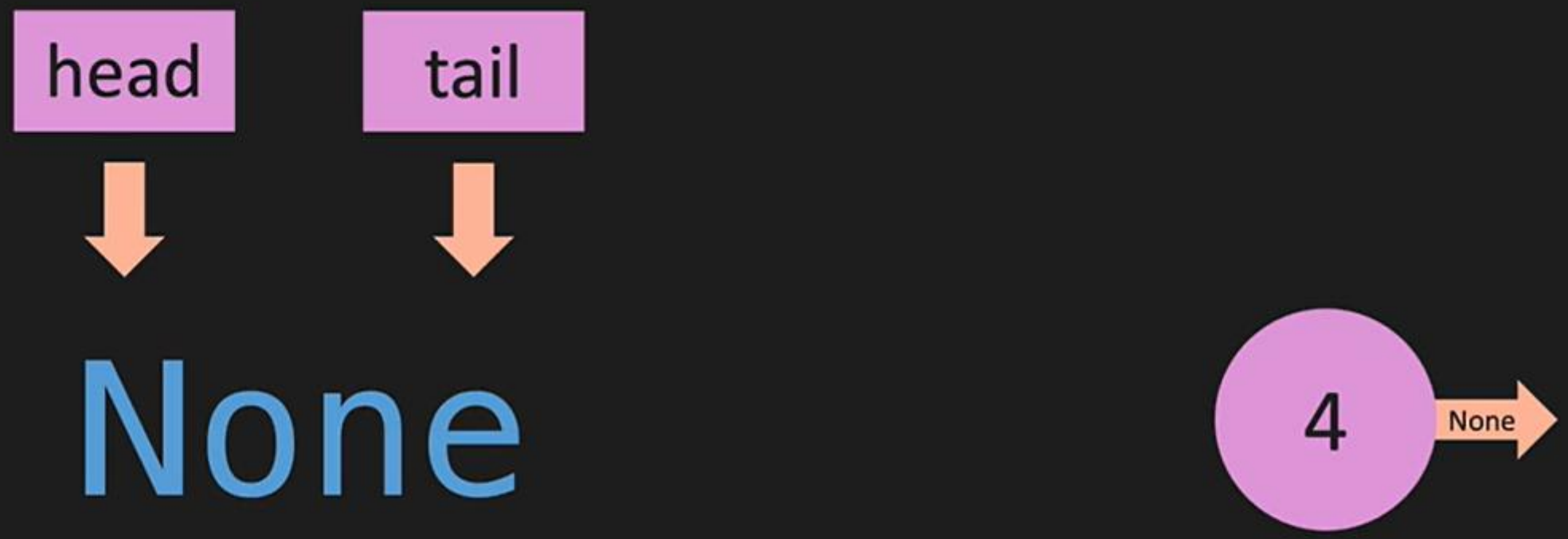
Appends

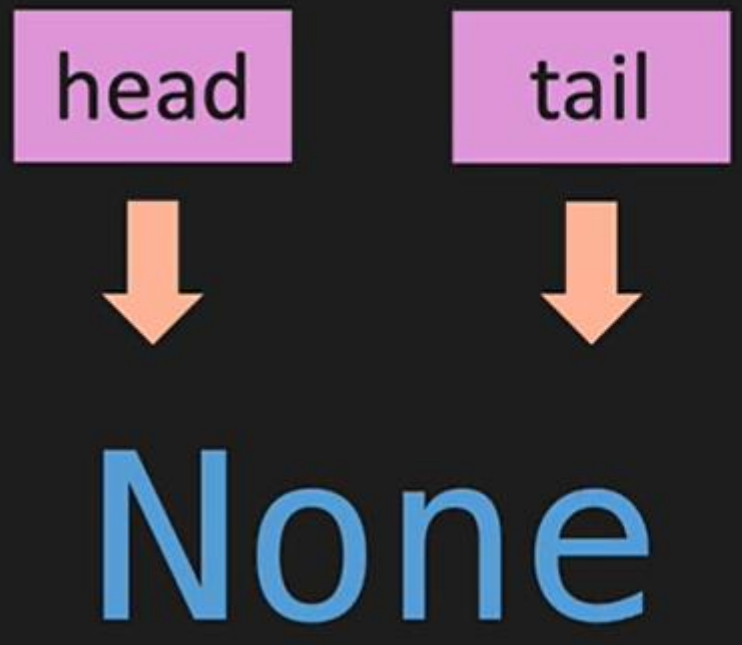



head
tail
11
3
23
7
4
None

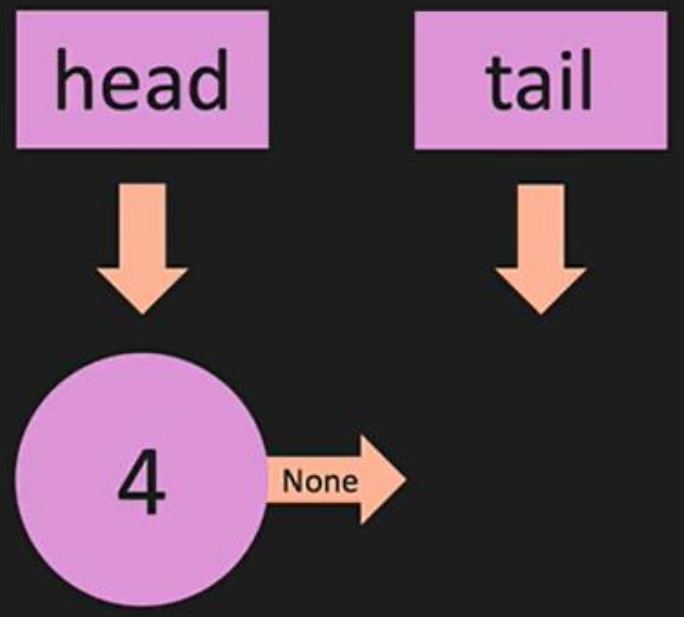

head
tail
None
4
None

```python
def append(self, value):
    new_node = Node(value)
```


head
tail
None
4
None

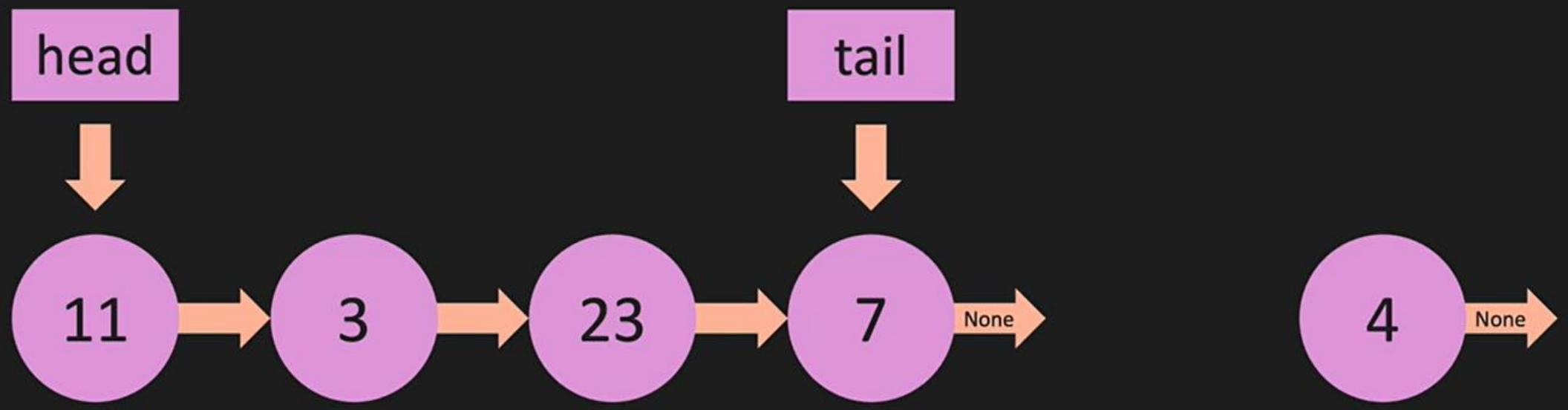



```
if self.head is None:
```

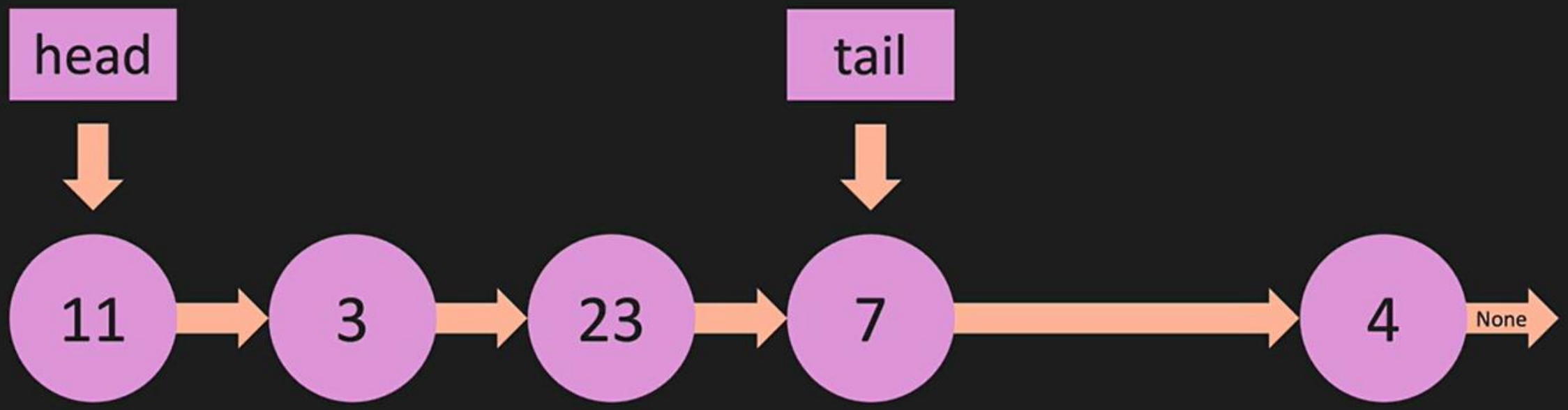



```
if self.head is None:
    self.head = new_node
    self.tail = new_node
```

```python
def append(self, value):
    new_node = Node(value)
    if self.head is None:
        self.head = new_node
        self.tail = new_node
```



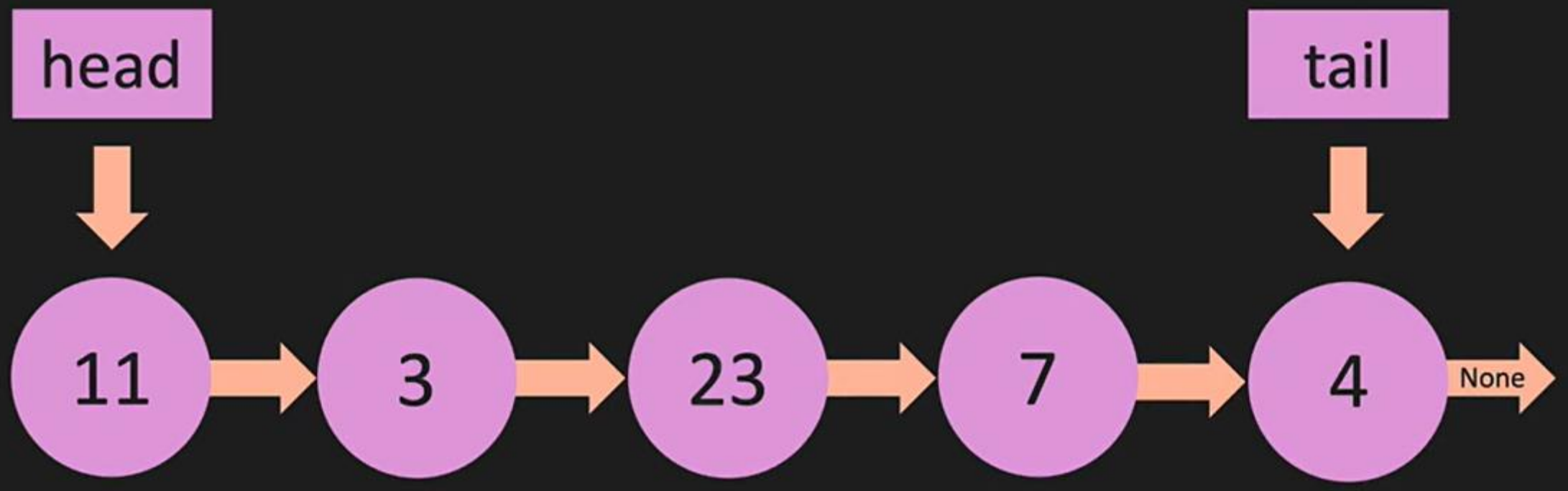

```
else:
    self.tail.next = new_node
```

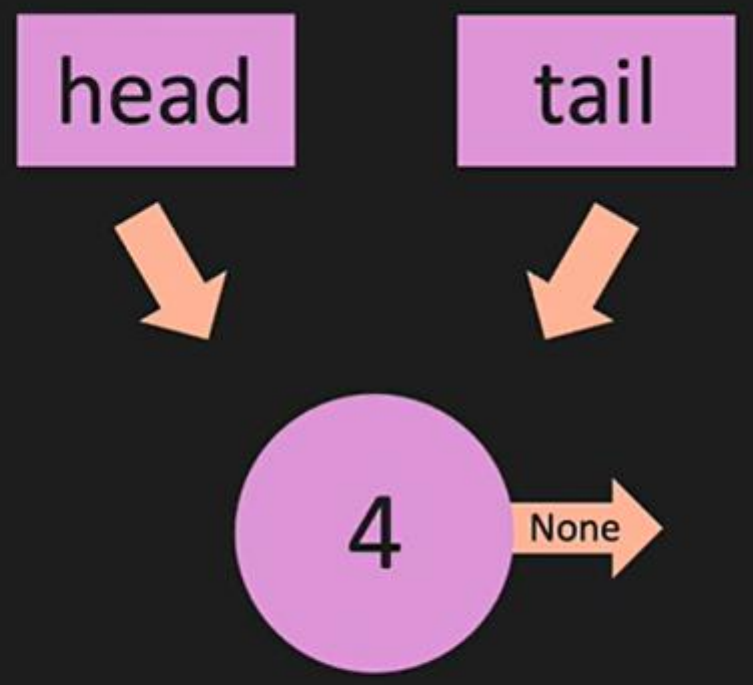


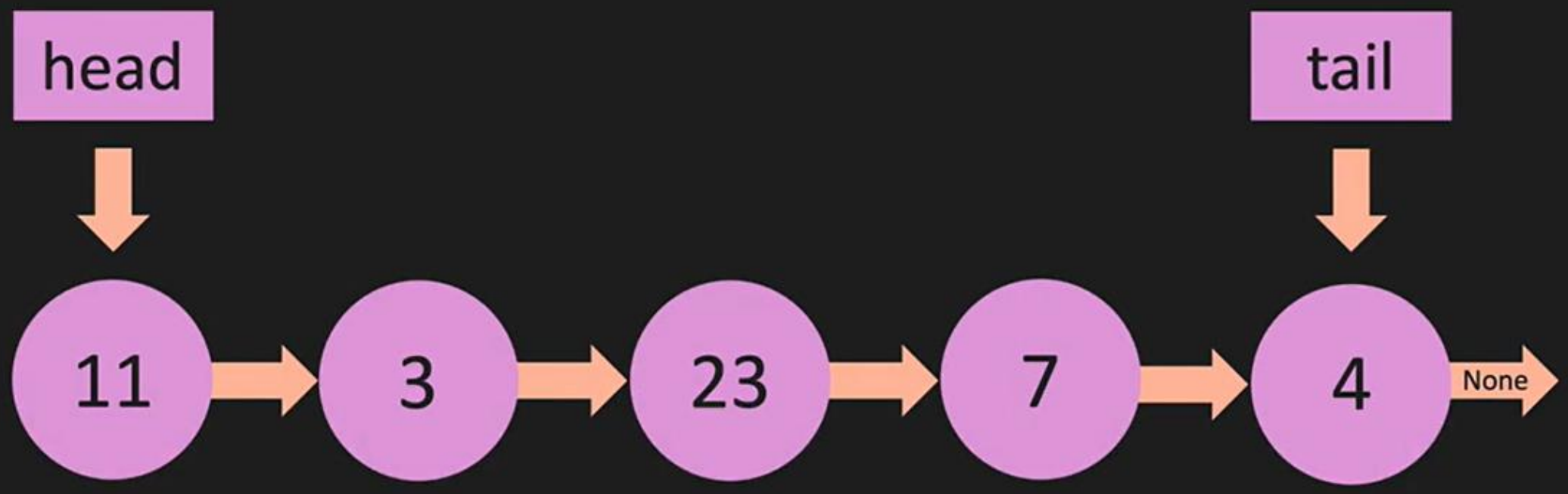

```
else:
    self.tail.next = new_node
    self.tail = new_node
```

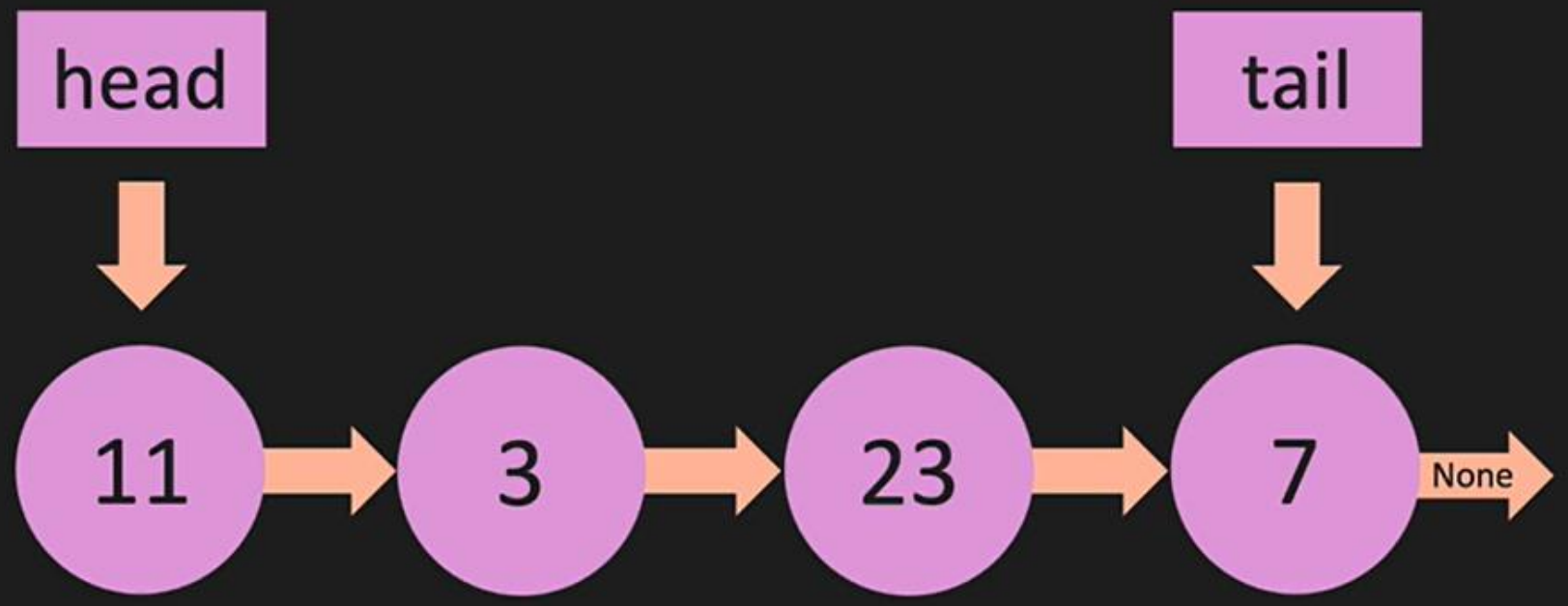
```python
def append(self, value):
    new_node = Node(value)
    if self.head is None:
        self.head = new_node
        self.tail = new_node
    else:
        self.tail.next = new_node
        self.tail = new_node
    self.length += 1
    return True
```

Pop



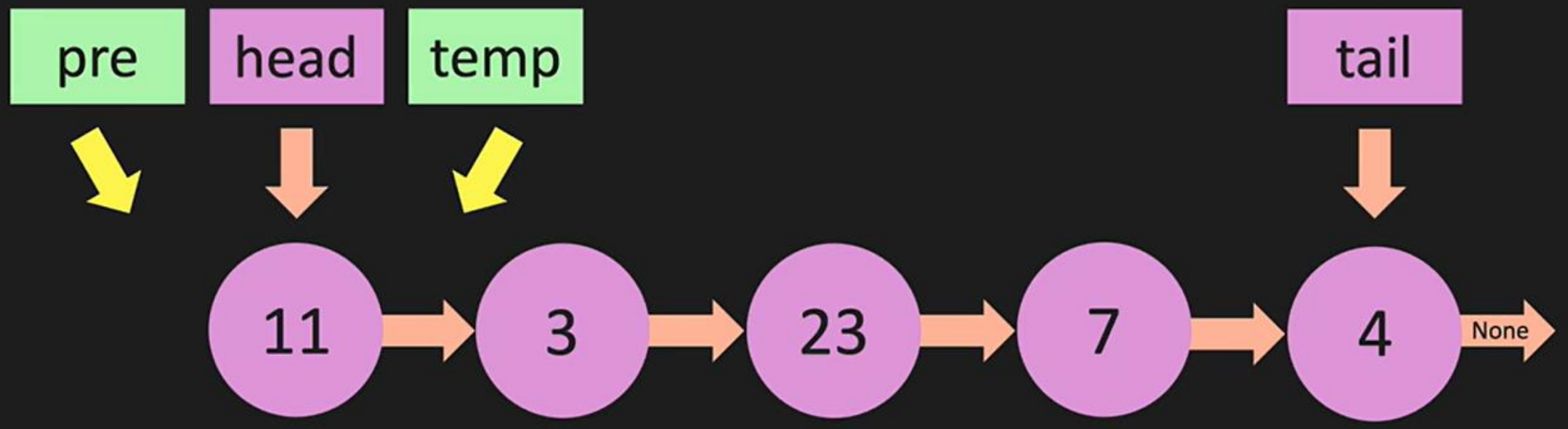

head
tail
4
None

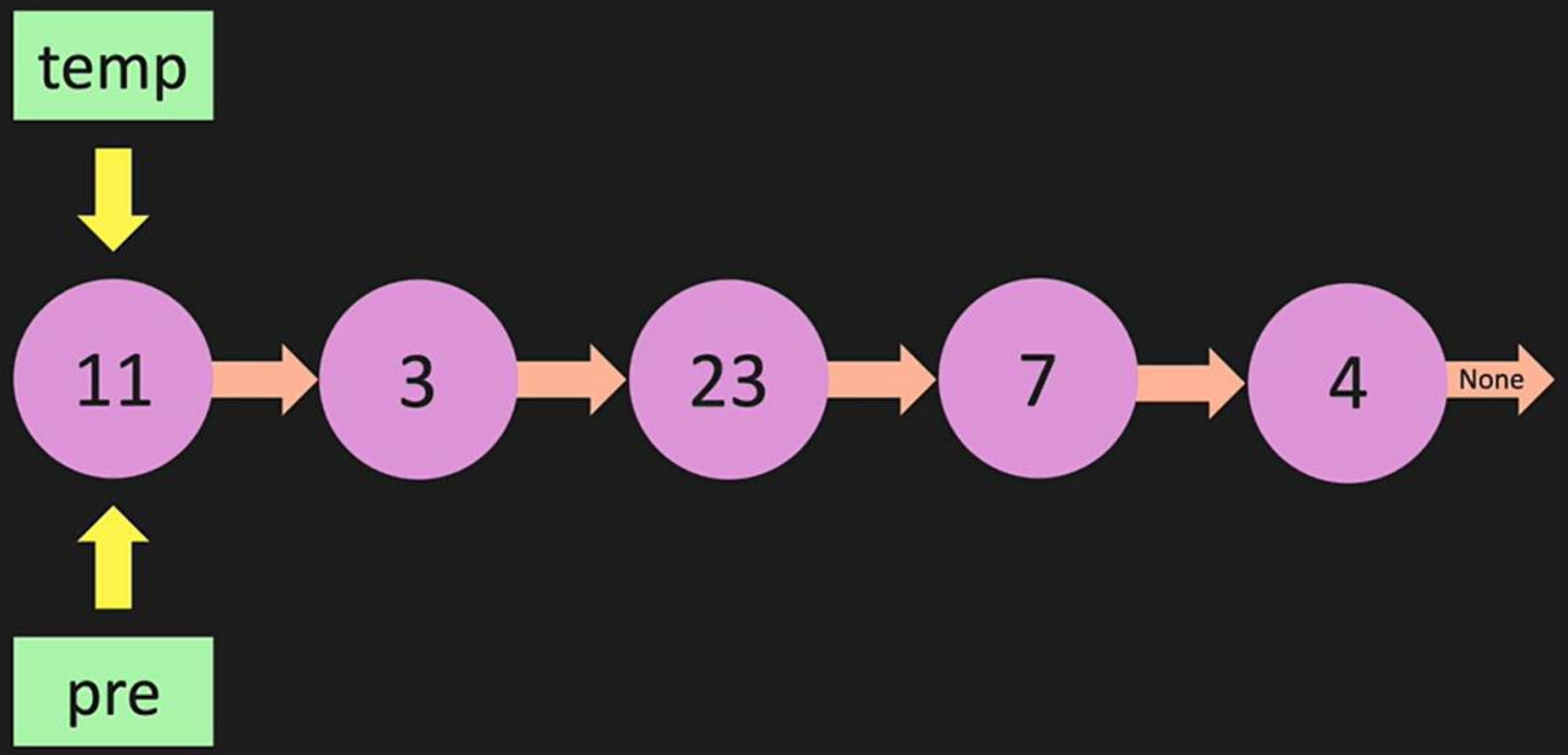

head
tail
11
3
23
7
4
None

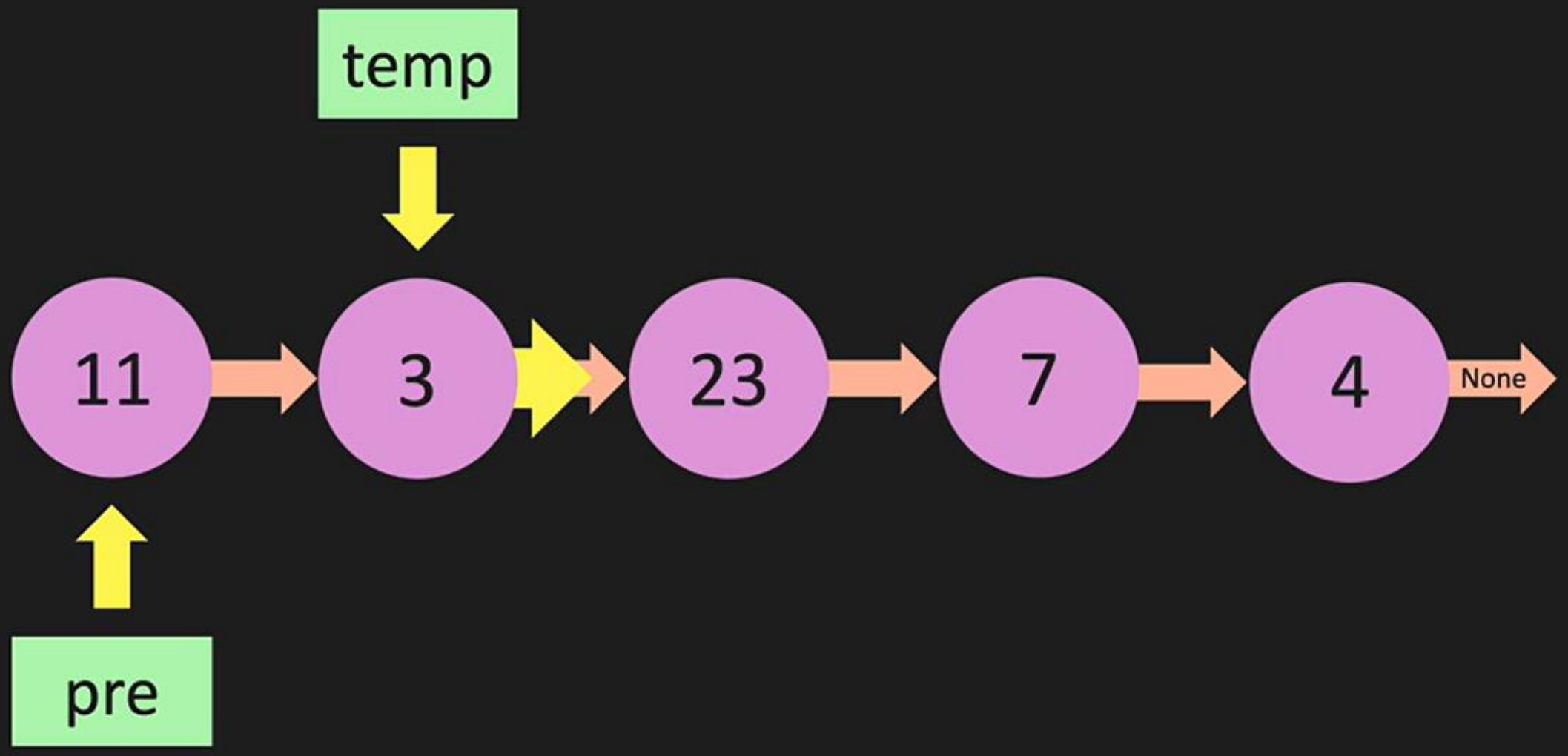

head
tail
11
3
23
7
None

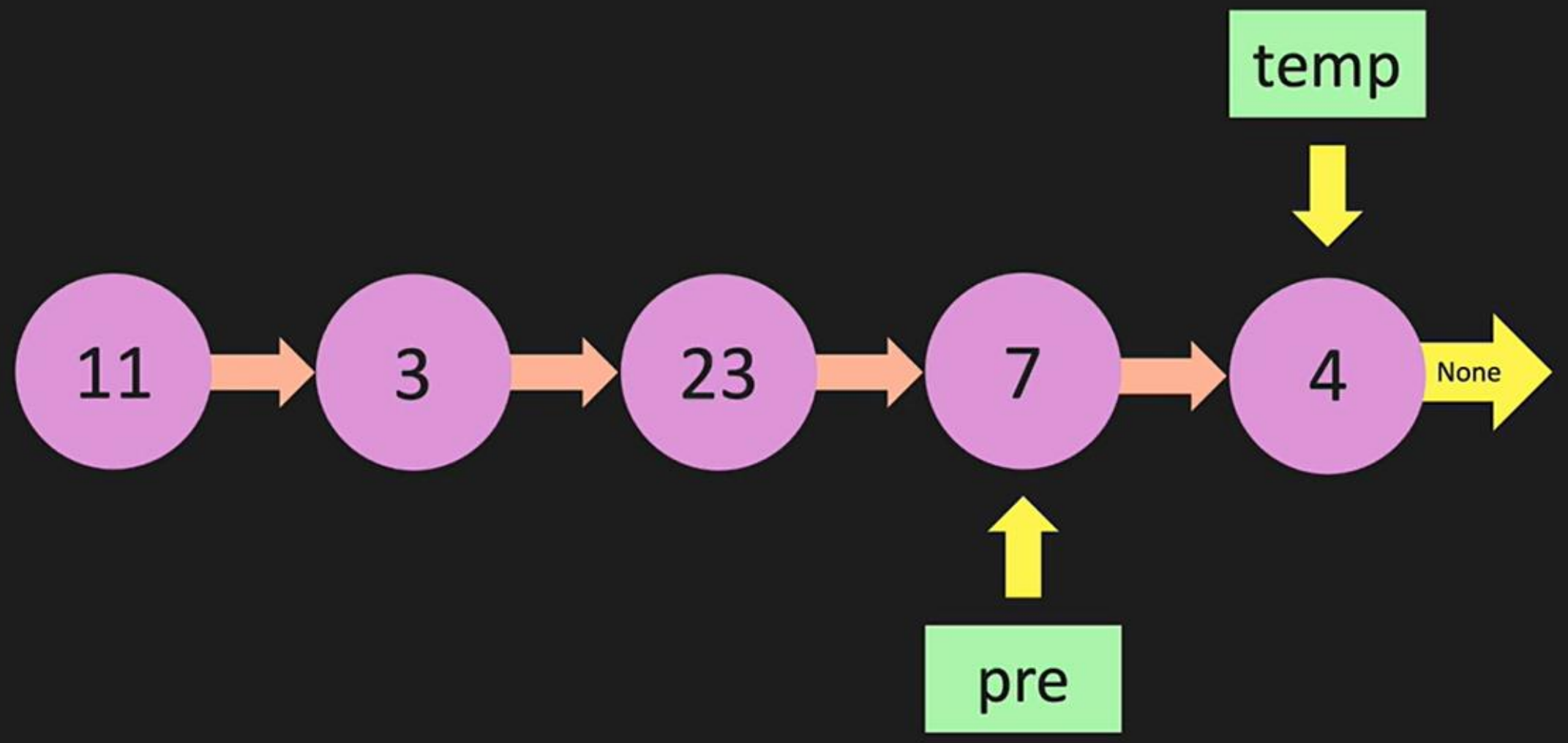
```
head: {
        "value": 11,
        "next": {
                "value": 3,
                "next": {
                        "value": 23,
                        "next": {
                                "value": 7,
                                "next": {
                                        "value": 4,
tail:                                   "next": None
                                        }
                                }
                        }
                }
        }
```

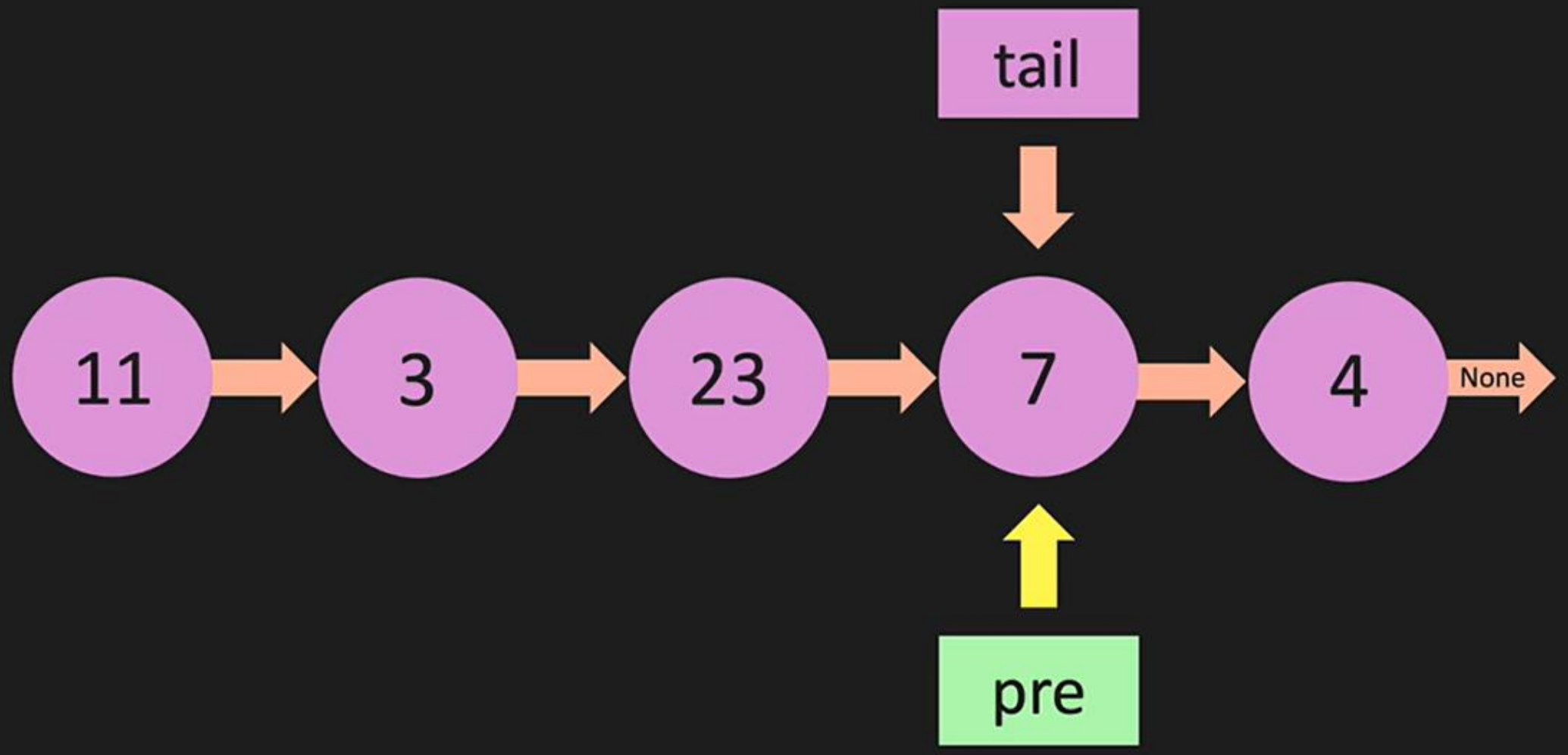
```
head: {
        "value": 11,
        "next": {
                "value": 3,
                "next": {
                        "value": 23,
                        "next": {
                                "value": 7,
tail:                           "next": None
                                }
                        }
                }
        }
```

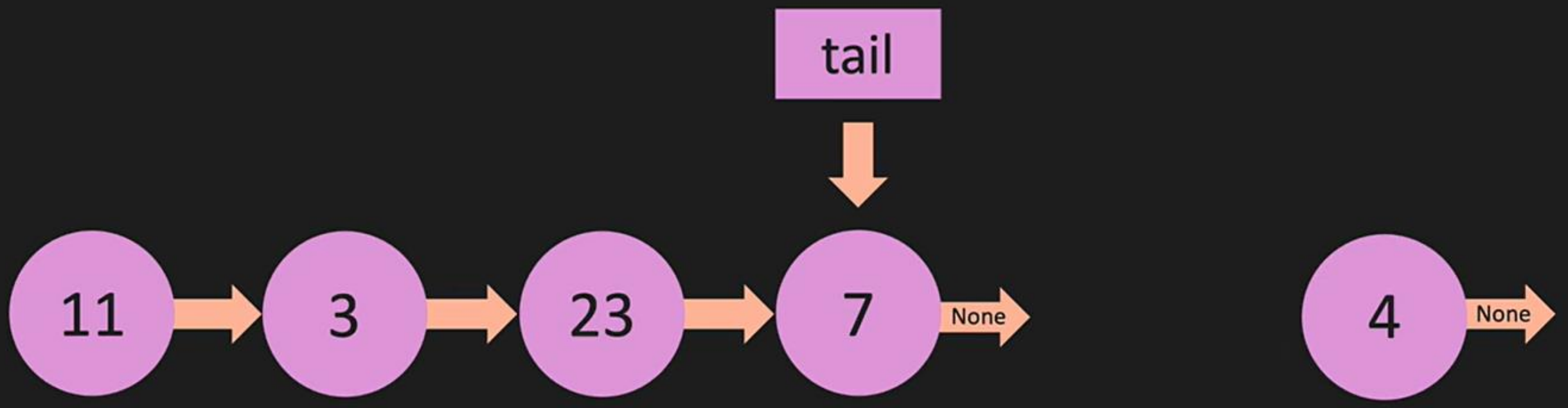

pre
head
temp
tail
11
3
23
7
4
None


temp
11
3
23
7
4
None
pre

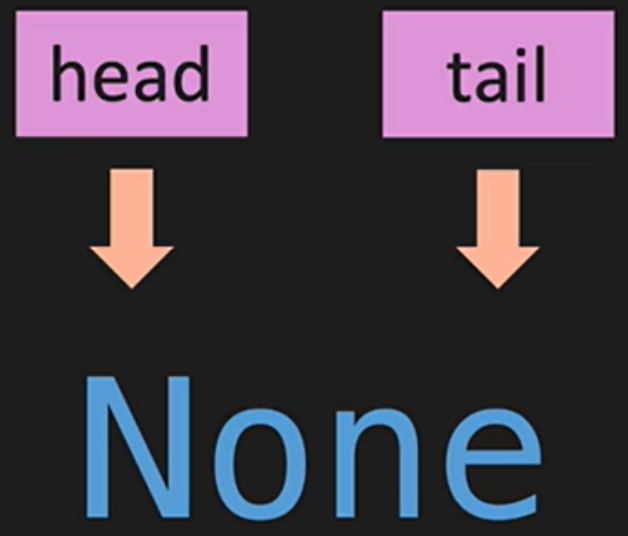

temp
11
3
23
7
4
None
pre


temp
11
3
23
7
4
None
pre


tail
11
3
23
7
4
None
pre

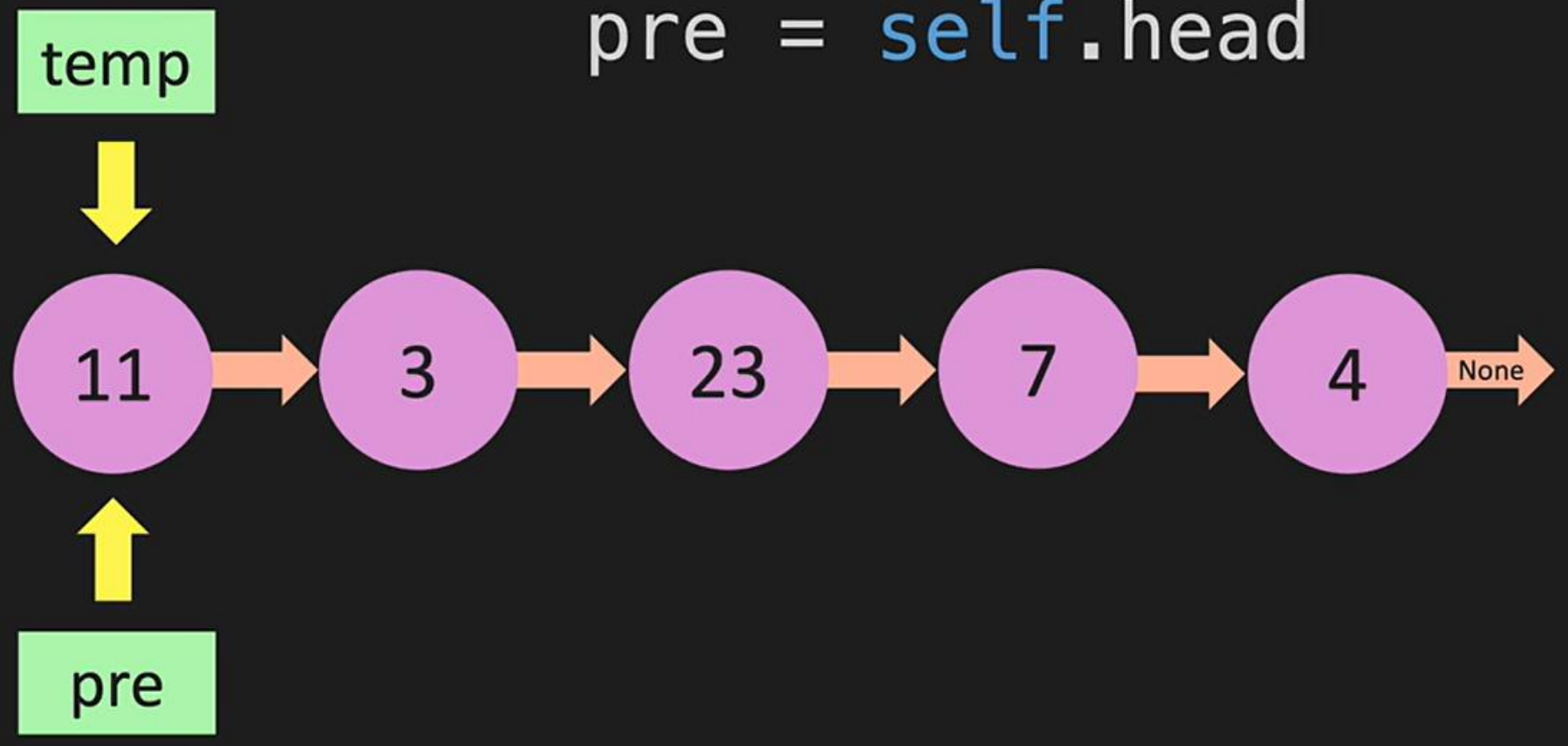

tail
11
3
23
7
None
4
None

```
def pop(self):
```

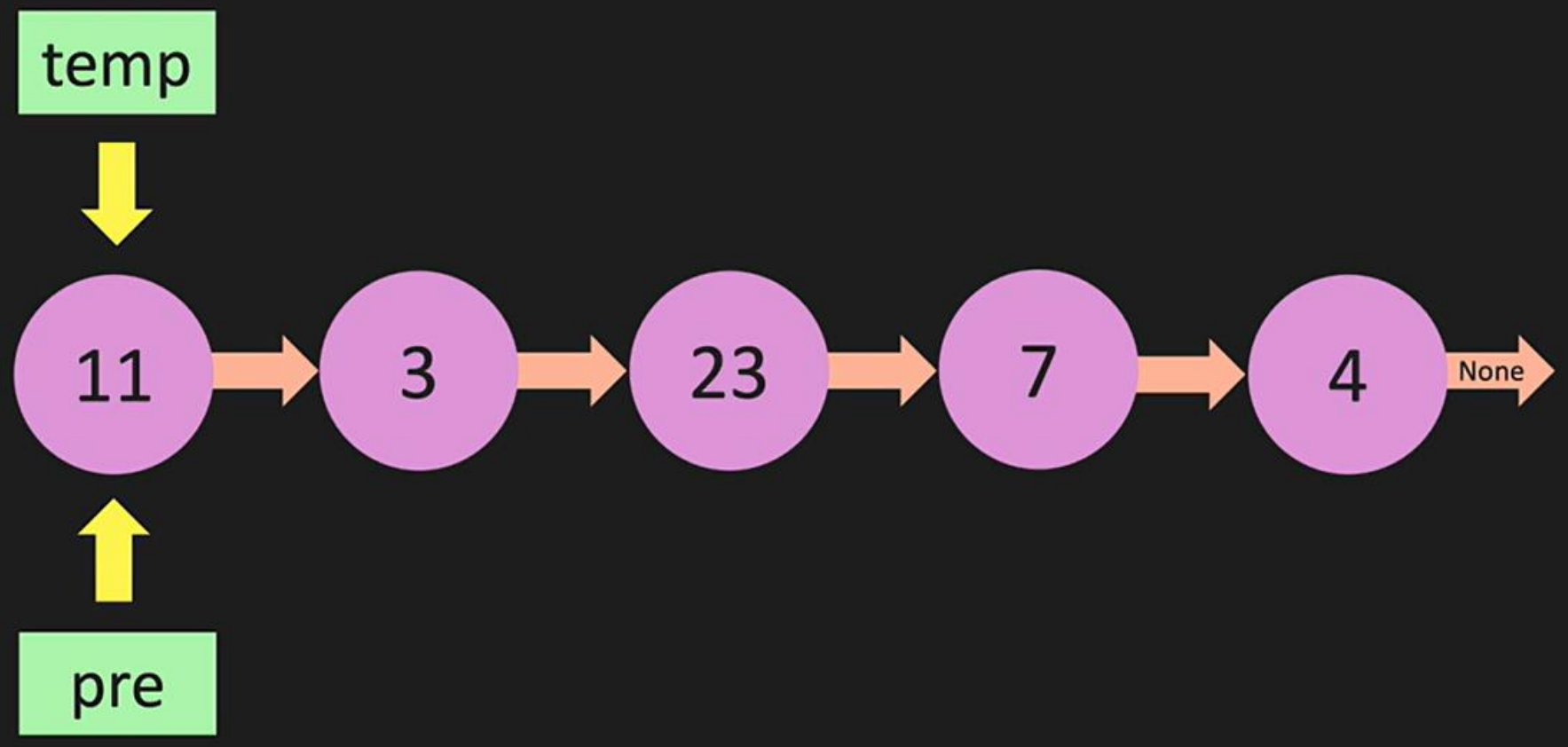


```
def pop(self):
    if self.length == 0:
        return None
```

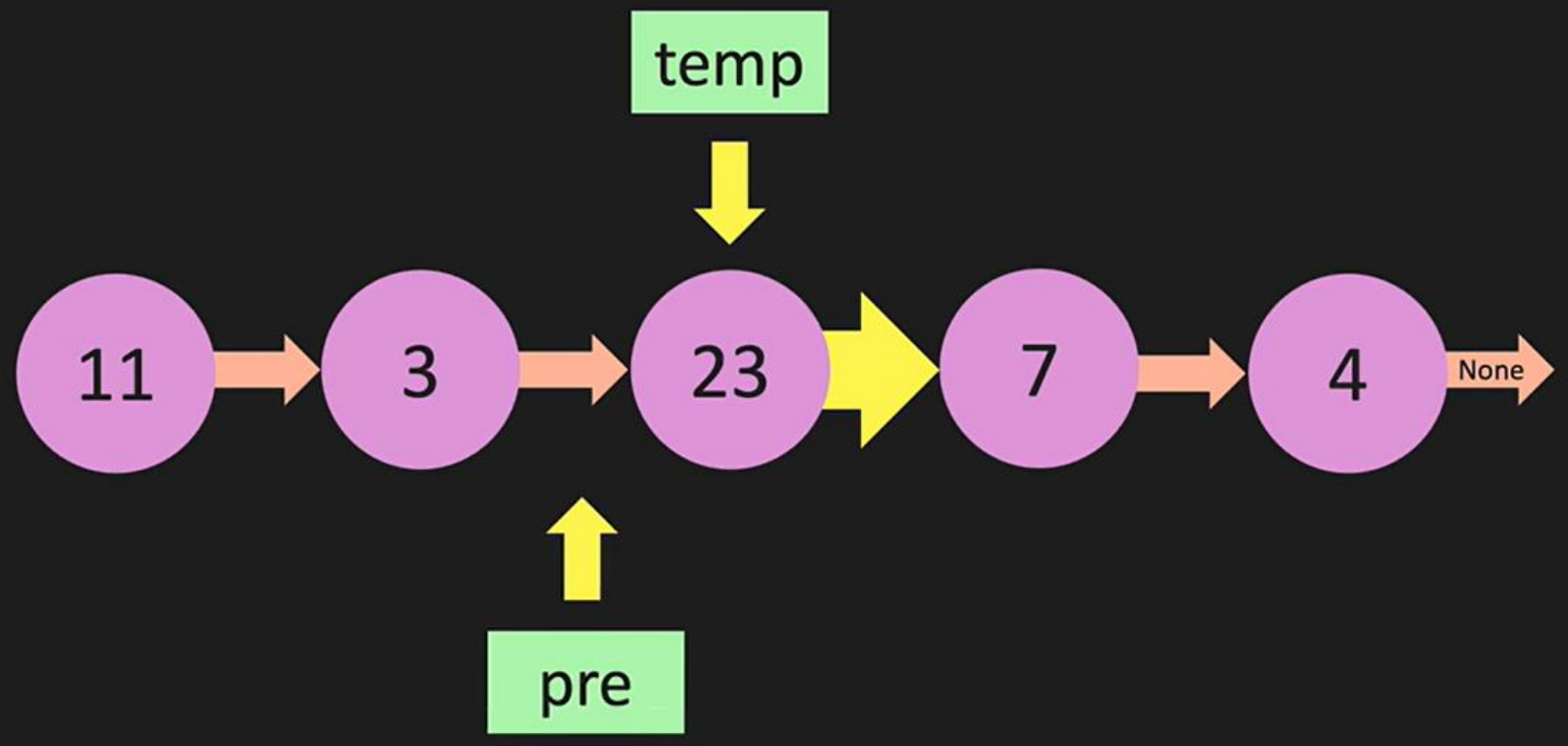
```
temp = self.head
pre = self.head
```



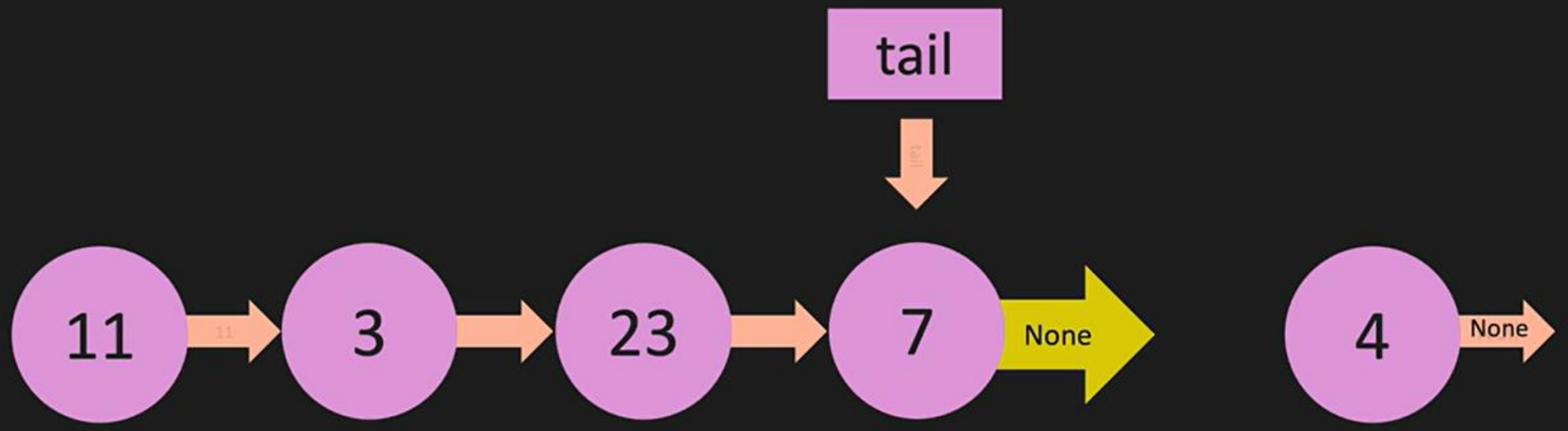
```
while(temp.next):
```



```
while(temp.next):
    pre = temp
    temp = temp.next
```



```
while(temp.next):
    pre = temp
    temp = temp.next
self.tail = pre
self.tail.next = None
```



```python
def pop(self):
    if self.length == 0:
        return None
    temp = self.head
    pre = self.head
    while(temp.next):
        pre = temp
        temp = temp.next
    self.tail = pre
    self.tail.next = None
    self.length -= 1
    if self.length == 0:
```