

COMP 250

INTRODUCTORY SCIENCE

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Week 5-2 : Singly Add WeChat: edu_assist_pro

Giulia Alberini, Fall 2020

WHAT ARE WE GOING TO DO IN THIS VIDEO?



- Singly Linked Lists Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Assignment Project Exam Help

<https://eduassistpro.github.io/>

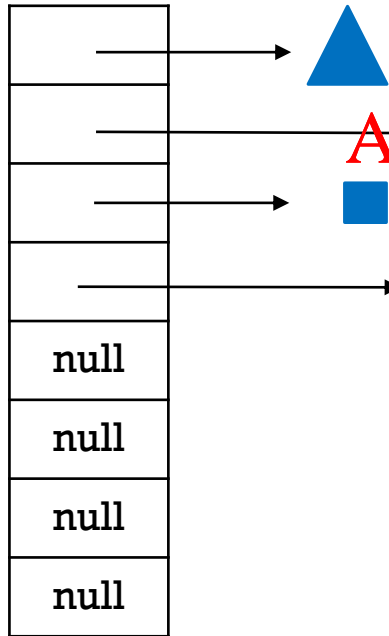
Add WeChat edu_assist_pro

IMPLEMENTATIONS

There are different implementations of a list:

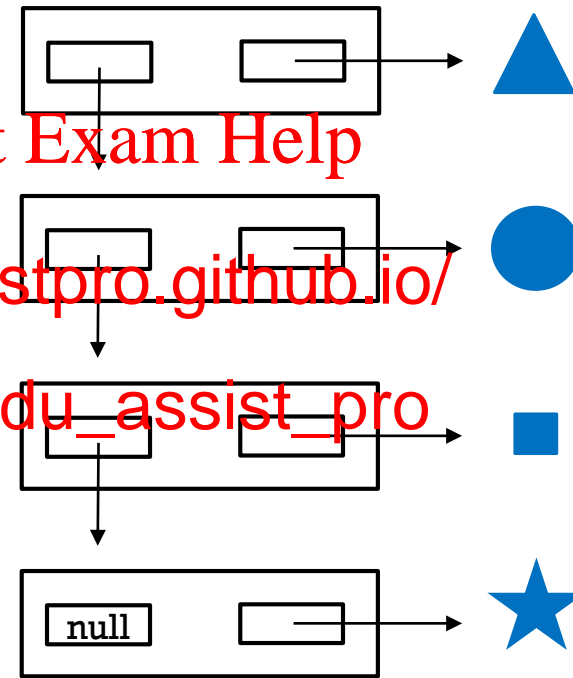
- Array list
 - Singly linked list
 - Doubly linked list
- Assignment Project Exam Help
<https://eduassistpro.github.io/>
n the list are linked using poi
Add WeChat edu_assist_pro

Array list



Linked list

"nodes"



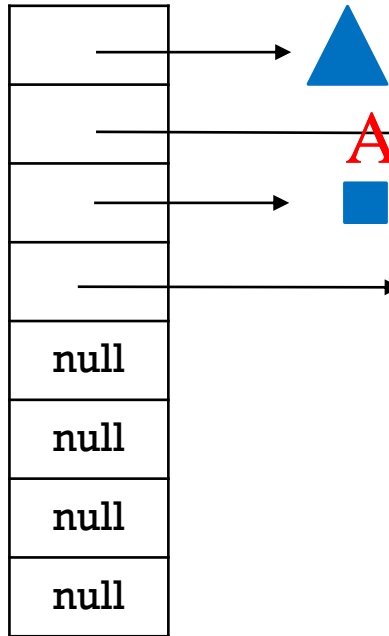
Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

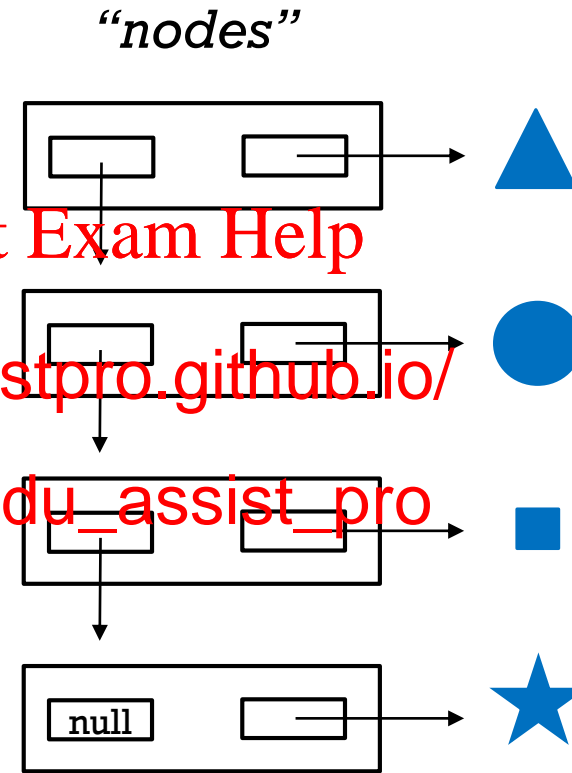
size = 4

Array list



Array slots are in consecutive locations (addresses) in memory, but objects (elements) can be anywhere.

Linked list



Linked list “nodes” and objects (elements) can be anywhere in memory.

SINGLY LINKED LIST NODE

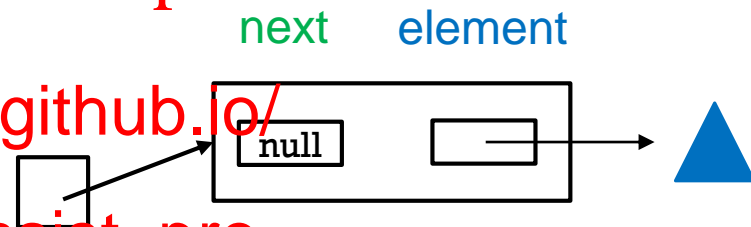
```
class SNode {  
    Shape element;  
    SNode next;  
}
```

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

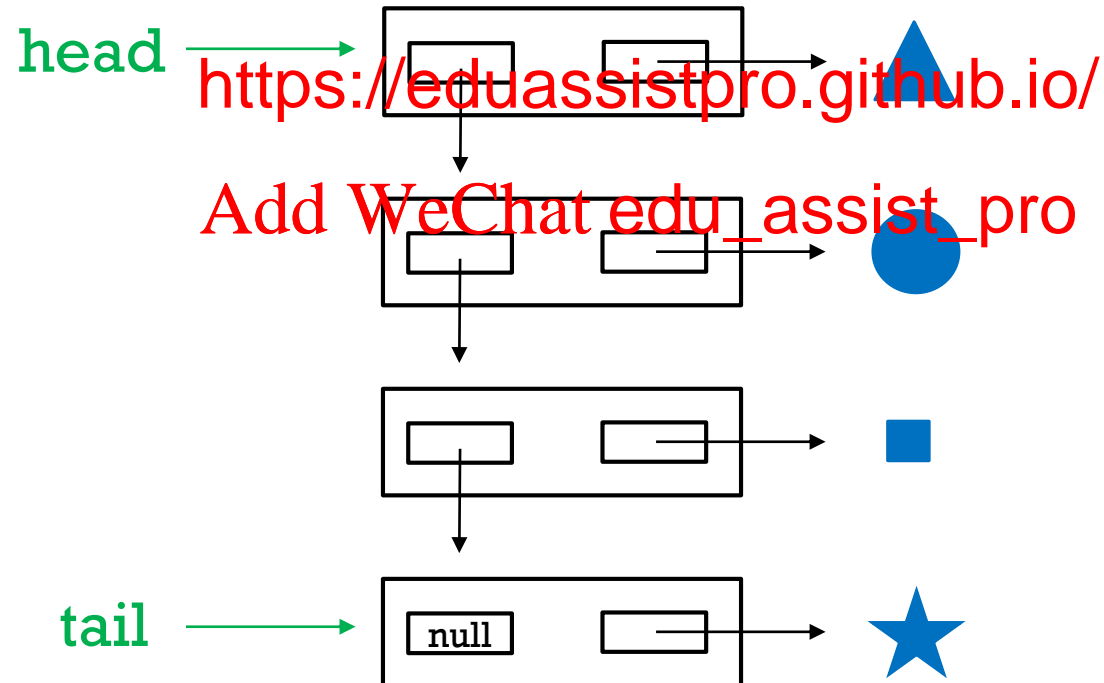
```
SNode myNode = new SNode();  
n.element = new Shape(▲);
```



SINGLY LINKED LIST

We think of a linked list as a sequence of nodes, along with a reference to the first (**head**) and last (**tail**) node.

Assignment Project Exam Help



SINGLY LINKED LIST

```
public class SLinkedList {  
    private SNode head;  
    private SNode tail;  
    private int size;  
    :  
    private class SNode {  
        Shape element;  
        SNode next;  
    }  
}
```

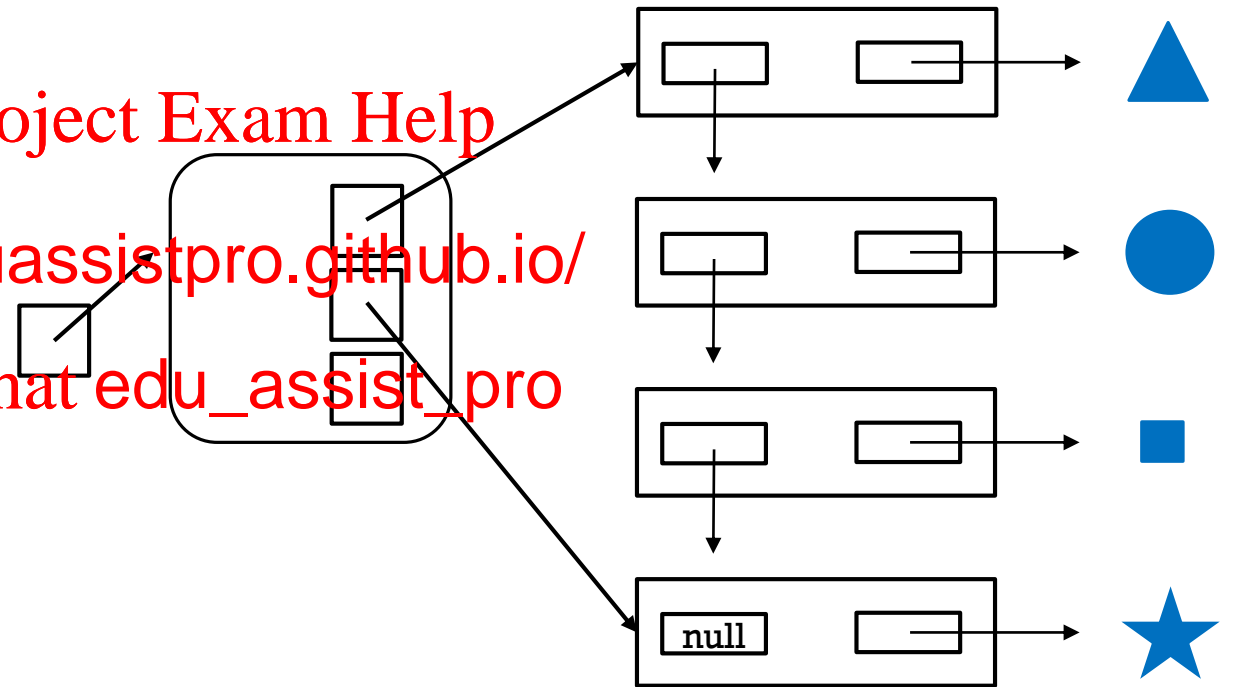
```
SLinkedList list = new SLinkedList();  
:
```

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

list



LINKED LIST OPERATIONS

- `addFirst (e)`
- `removeFirst()`
- `addLast (e)`
- `removeLast()`
-many other list operations

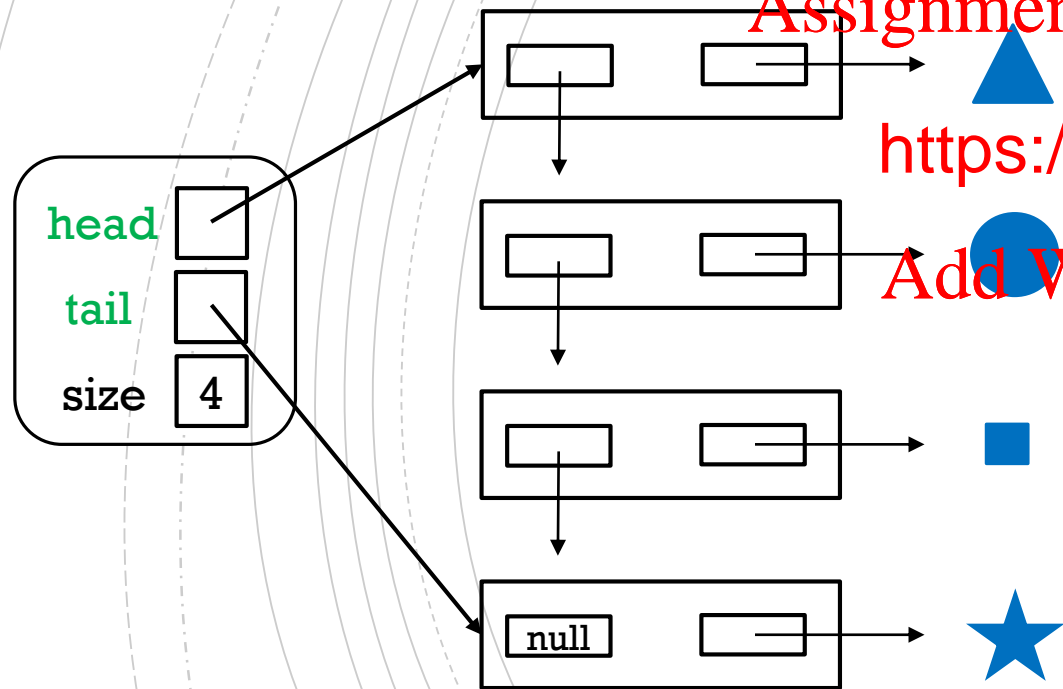
Assignment Project Exam Help

<https://eduassistpro.github.io/>

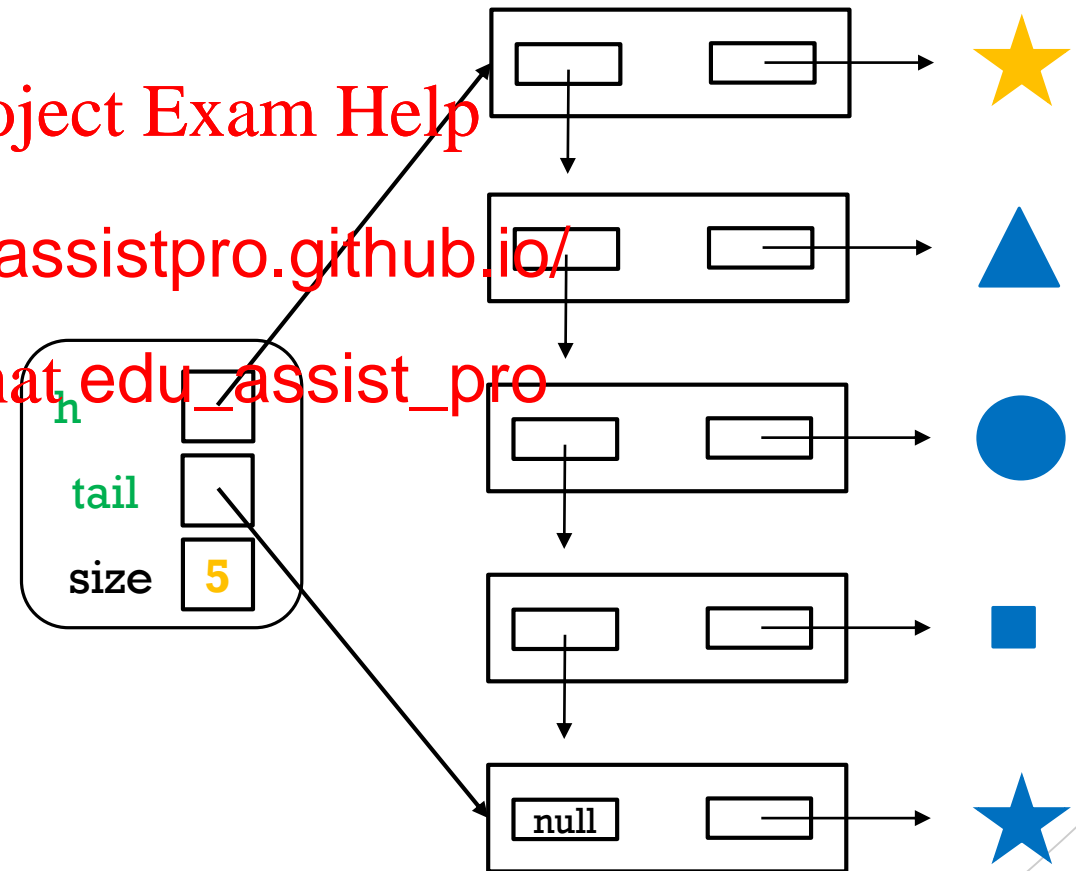
Add WeChat edu_assist_pro

LINKED LIST OPERATIONS – addFirst(★)

BEFORE



AFTER



addFirst(e) – PSEUDOCODE

```
SNode newNode = new SNode();
```

```
newNode.element = e;
```

```
newNode.next = head;
```

```
// edge case
```

```
if (head == null)
```

```
    tail = newNode;
```

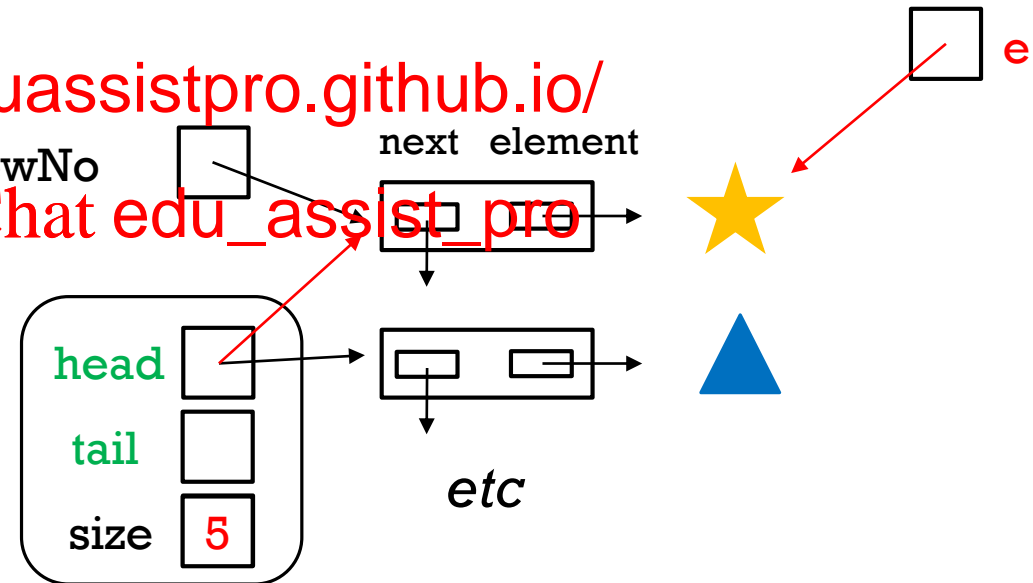
```
head = newNode;
```

```
size = size + 1;
```

Assignment Project Exam Help

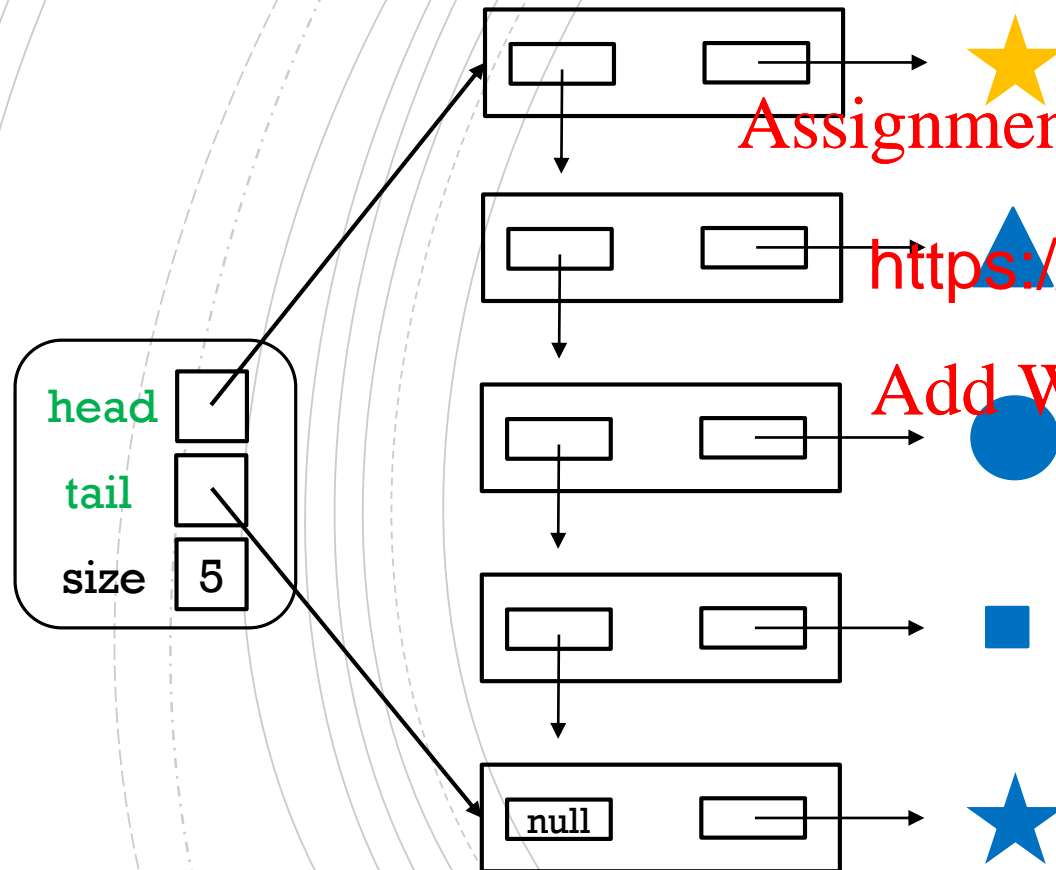
<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

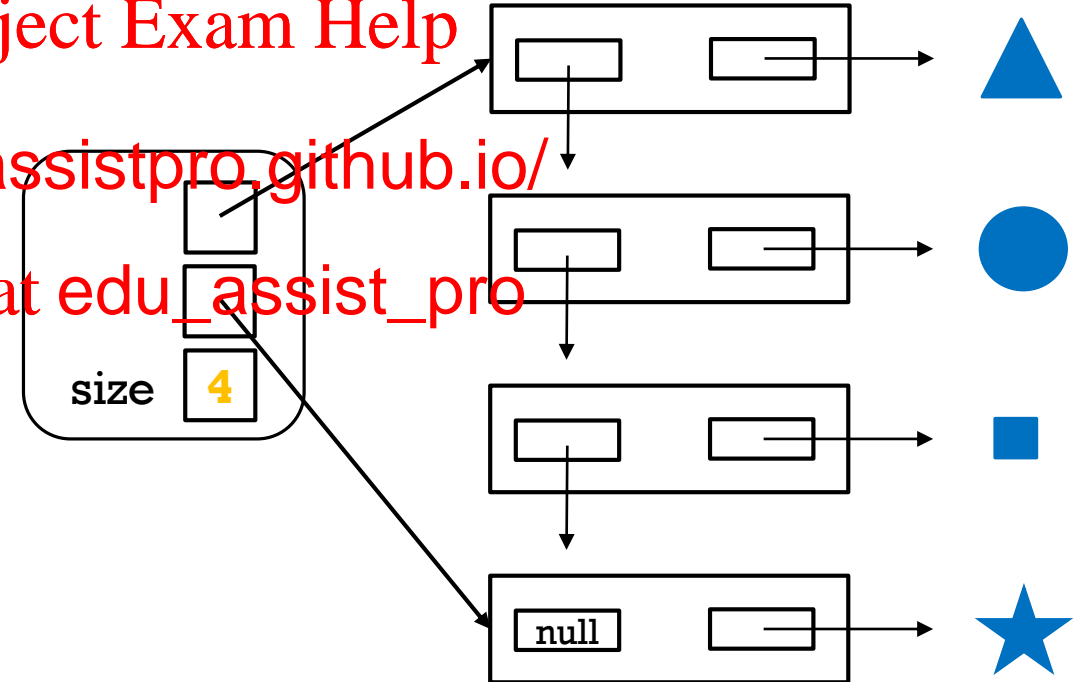


LINKED LIST OPERATIONS – removeFirst()

BEFORE



AFTER



Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

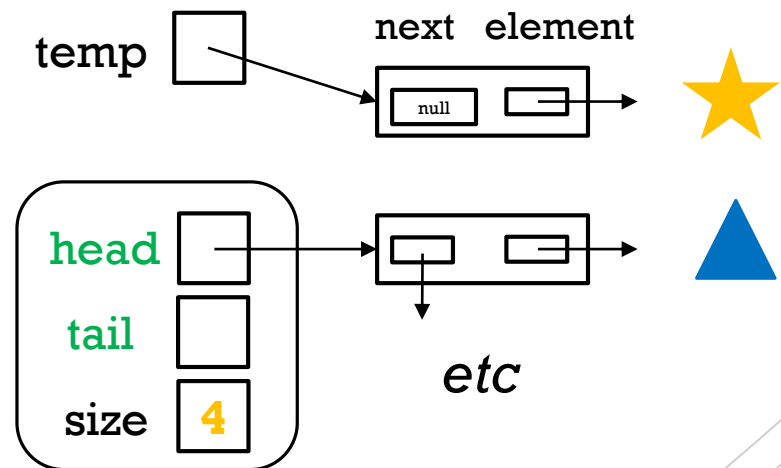
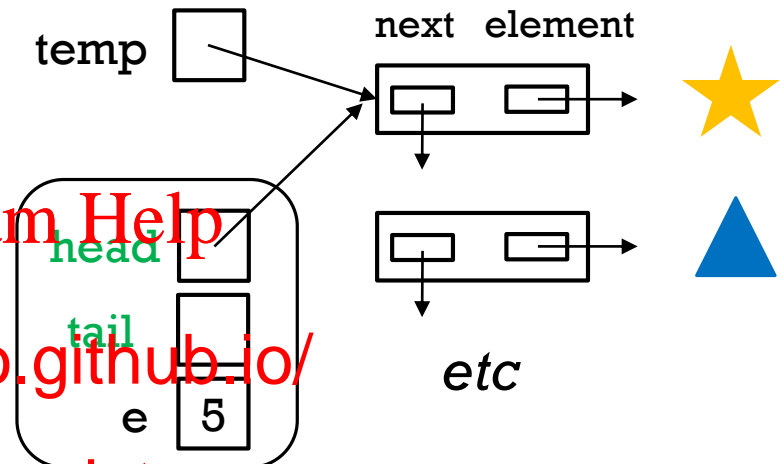
removeFirst() – PSEUDOCODE

```
SNode temp = head;  
head = temp.next;  
temp.next = null; // not required  
size = size - 1;  
return temp.element;
```

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



removeFirst() – EDGE CASES (SIZE IS 0 OR 1)

```
SNode temp = head;
```

```
if (size == 0)
```

```
    throw exception
```

```
head = temp.next;
```

```
temp.next = null;
```

```
size = size - 1;
```

```
if (size == 0)
```

```
    tail = null;
```

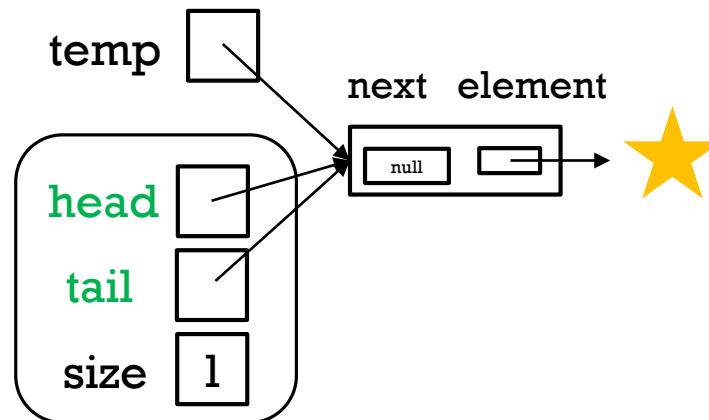
```
return temp.element;
```

Assignment Project Exam Help

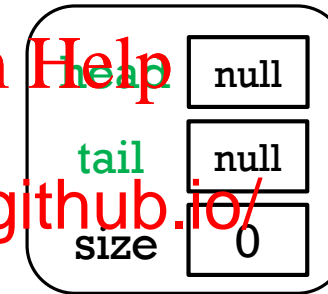
<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

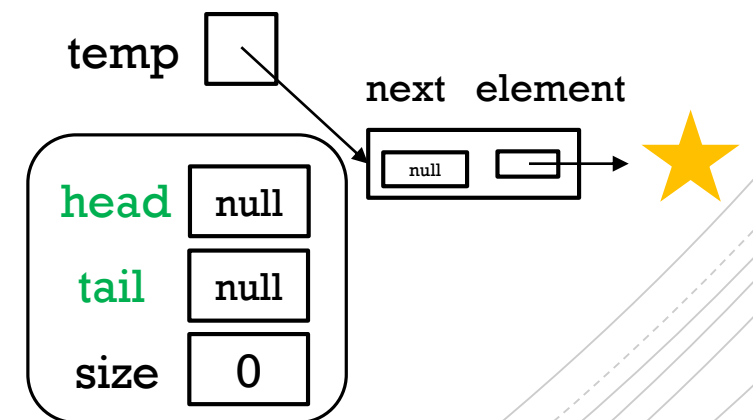
Size = 0
Size = 1



temp null



fore After



WORSE CASE TIME COMPLEXITY (N = LIST SIZE)

array list

linked list

addFirst()

removeFirst()

Assignment Project Exam Help

<https://eduassistpro.github.io/>

$O(N)$

Add WeChat edu_assist_pro

For arraylist with N elements,
recall that add(0, e) and remove(0)
required a loop with N iterations

For linked lists the implementation
of addFirst() and removeFirst()
does not depend on the number of
elements in the list

WORSE CASE TIME COMPLEXITY (N = LIST SIZE)

	array list	linked list
addFirst()	$O(1)$	$O(1)$
removeFirst()	$O(N)$	$O(1)$
addLast()	$O(1)^*$?
removeLast()	$O(1)$?

Assignment Project Exam Help

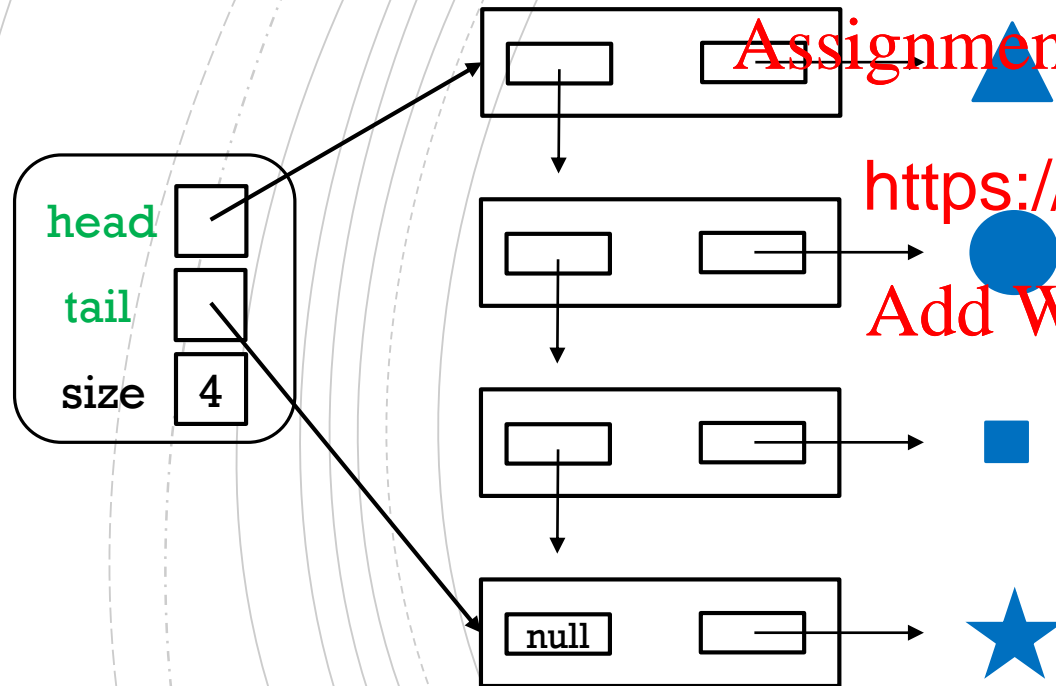
<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

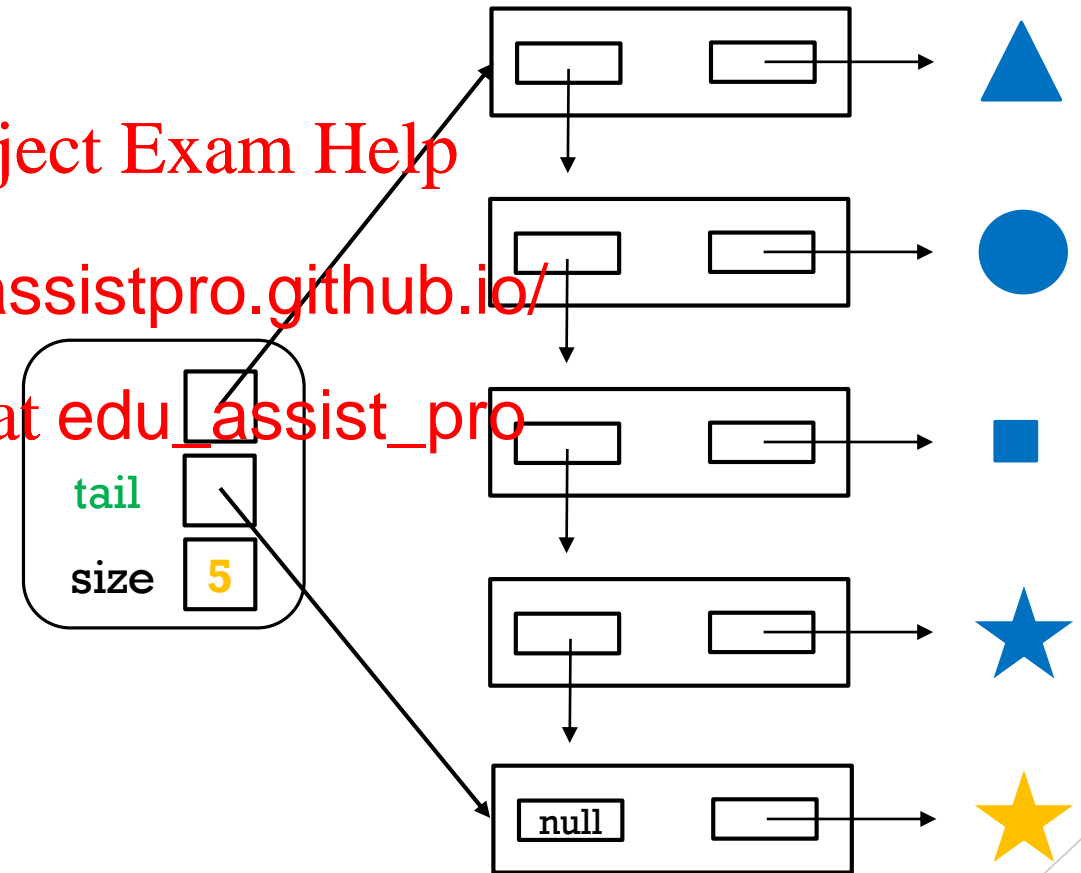
*if array is not full

LINKED LIST OPERATIONS – addLast(★)

BEFORE



AFTER



Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

addLast (e) – PSEUDOCODE

```
// create a new node
```

```
SNode newNode = new SNode();
```

```
newNode.element = e;
```

```
// add it at the end
```

```
tail.next = newNode;
```

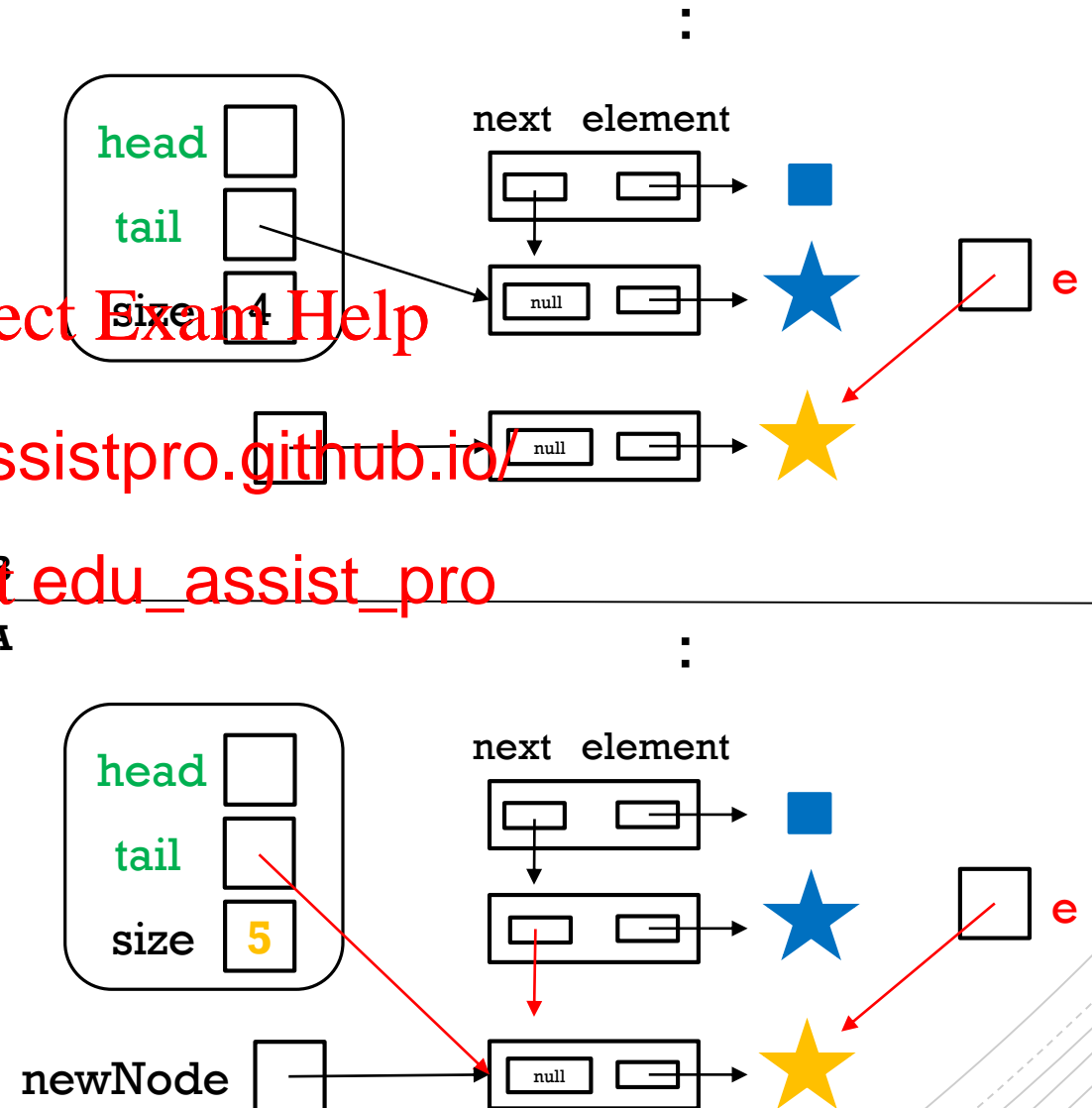
```
tail = tail.next;
```

```
size = size + 1;
```

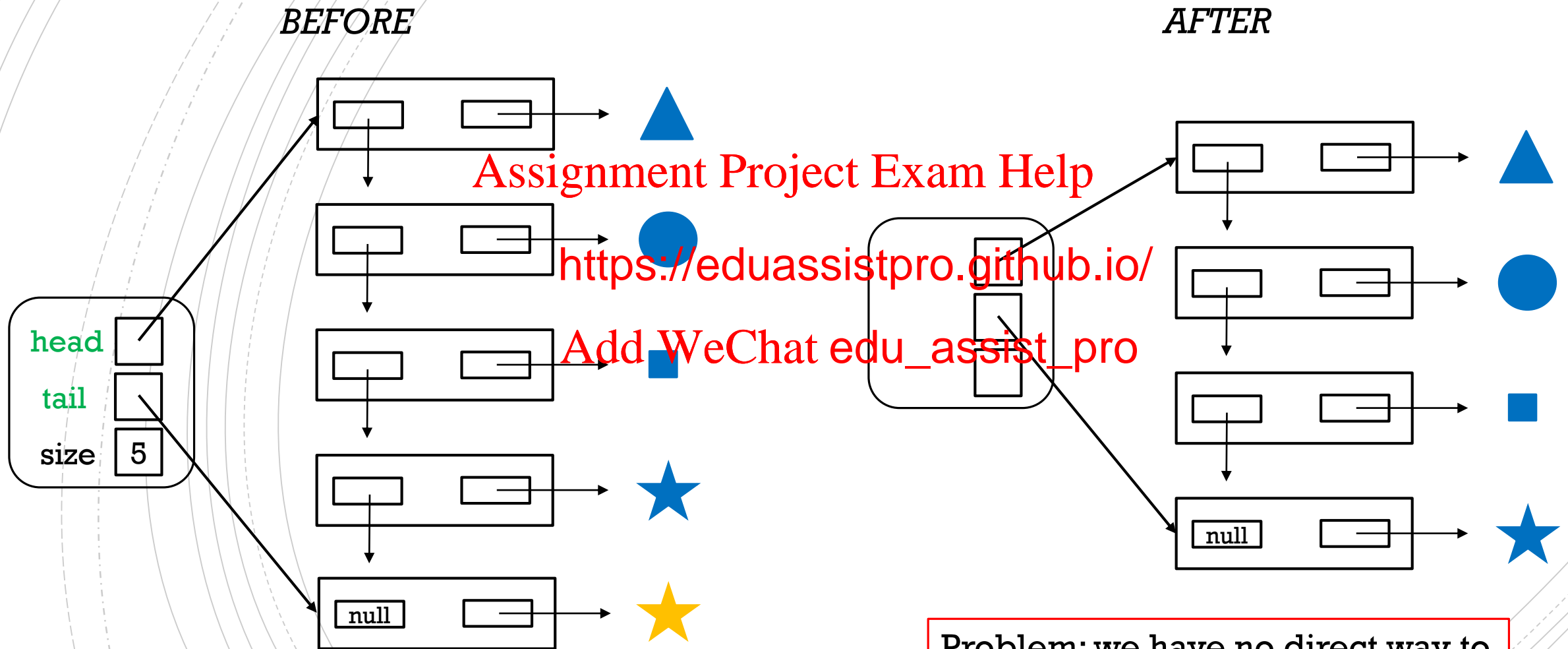
Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



LINKED LIST OPERATIONS – removeLast()



removeLast() – PSEUDOCODE

```
SNode tmp = head;
```

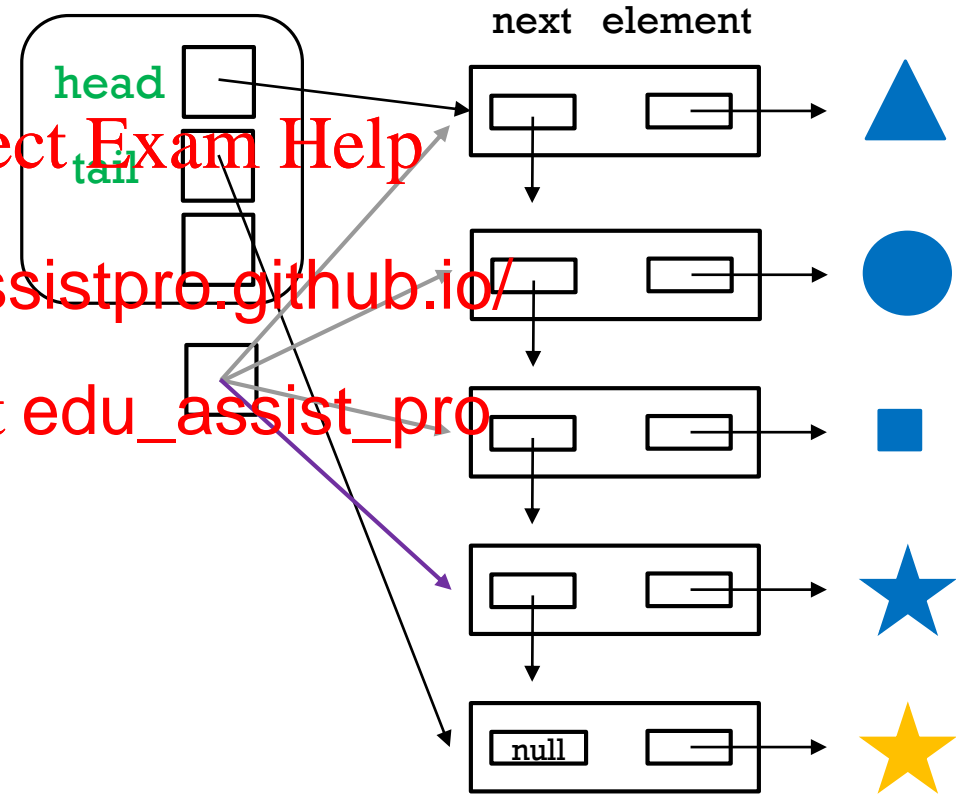
```
while (tmp.next != tail)
```

```
tmp = tmp.next;
```

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



removeLast() – PSEUDOCODE

```
SNode tmp = head;
```

```
while (tmp.next != tail)
```

```
    tmp = tmp.next;
```

```
tail = tmp;
```

```
tail.next = null;
```

```
size = size - 1;
```

```
// to return the element,
```

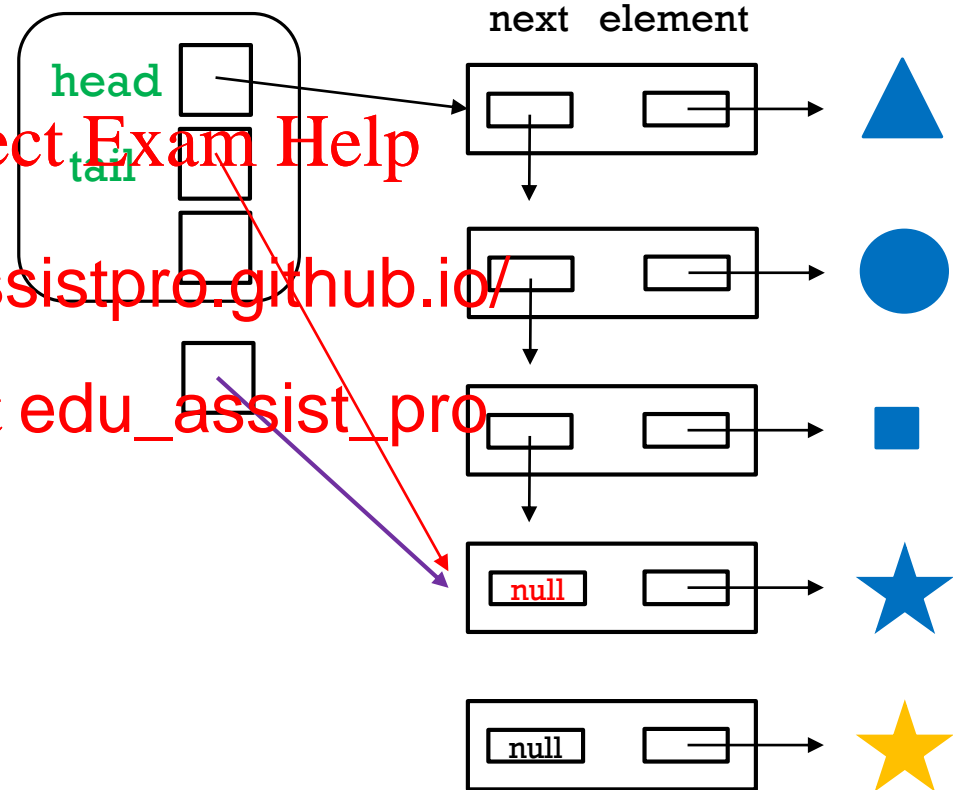
```
// you need to do a bit more work
```

```
// edge cases for size = 0 and 1 to be added
```

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



removeLast () – EDGE CASES (SIZE IS 0 OR 1)

```
if (size == 0)
```

```
    throw exception
```

```
if (size == 1)
```

```
    head = null;
```

```
    tail = null;
```

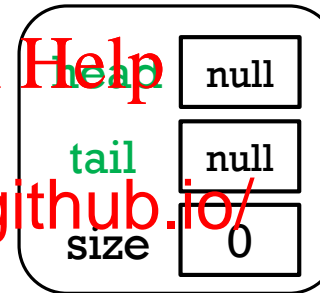
```
else {
```

```
    ...
```

```
}
```

```
size = size - 1;
```

temp null

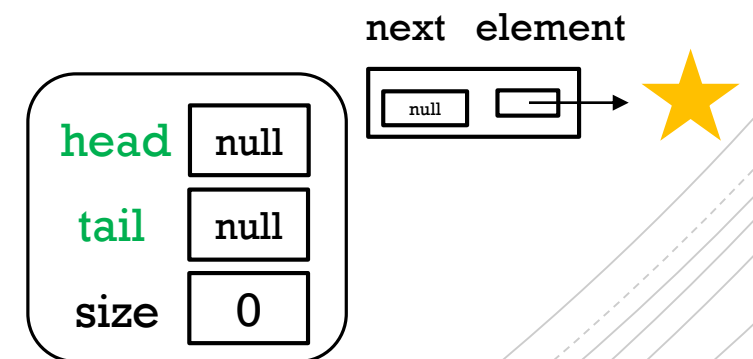
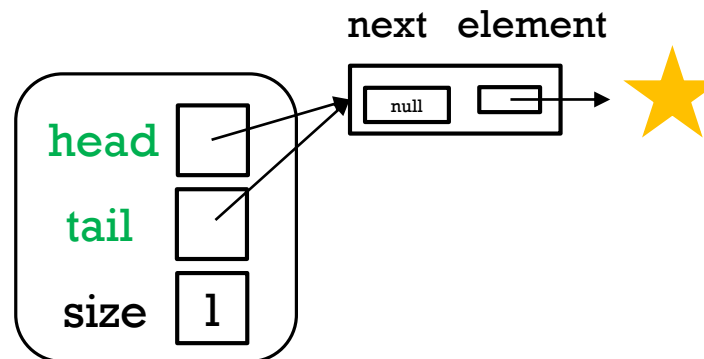


<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Size = 0
Size = 1

fore After



WORSE CASE TIME COMPLEXITY (N = LIST SIZE)

	array list	linked list
addFirst()		$O(1)$
removeFirst()	$O(N)$	$O(1)$
addLast()	$O(1)^*$	$O(1)$
removeLast()	$O(1)$	$O(N)$

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

*if array is not full

An orange paint roller with a red handle, positioned horizontally. The roller is covered in orange paint, which is dripping down the left side. The text "Coming Soon" is written in white on the orange surface of the roller.

Coming Soon

Assignment Project Exam Help

In the next

- Doubly <https://eduassistpro.github.io/>
Add WeChat edu_assist_pro