

COMP 250

Assignment Project Exam Help

INTRODUCING COMPUTER SCIENCE

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

Week 5-8 : October

Add WeChat edu_assist_pro

Giulia Alberini, Fall 2020

WHAT ARE WE GOING TO DO IN THIS VIDEO?



- Doubly 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

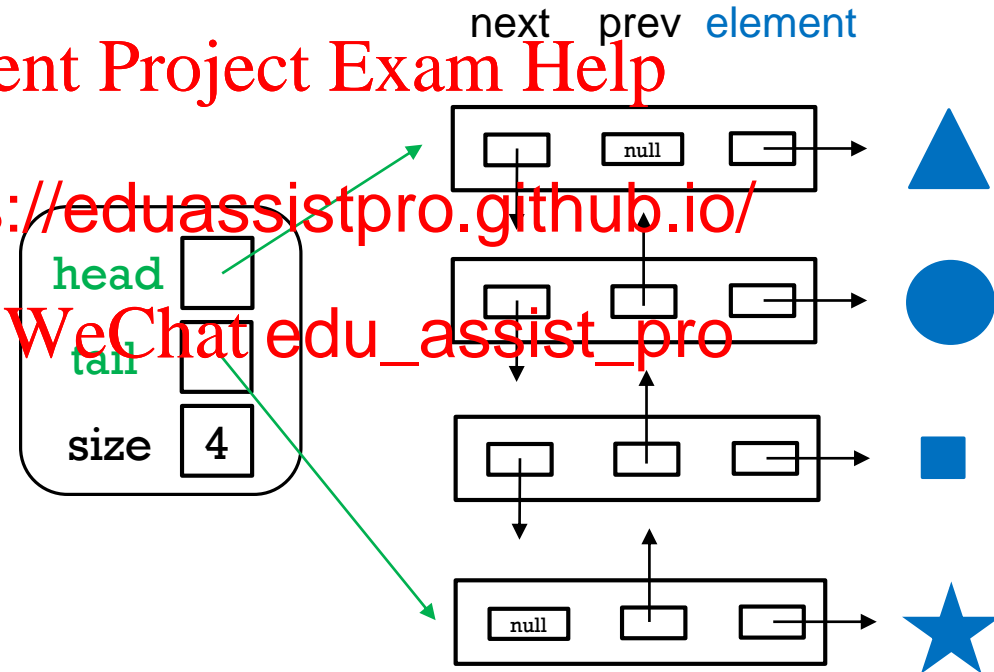
DOUBLY LINKED LIST

Each node has a reference to the next node *and* to the previous node.

Assignment Project Exam Help

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

Add WeChat edu_assist_pro



DOUBLY LINKED LIST NODE

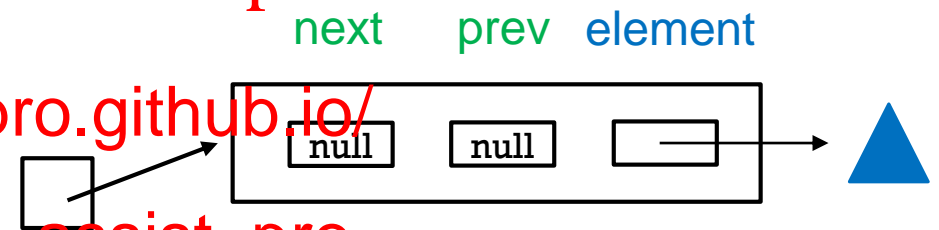
```
class DNode {  
    Shape element;  
    DNode next;  
    DNode prev;  
}
```

Assignment Project Exam Help

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

Add WeChat edu_assist_pro

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



DOUBLY LINKED LIST

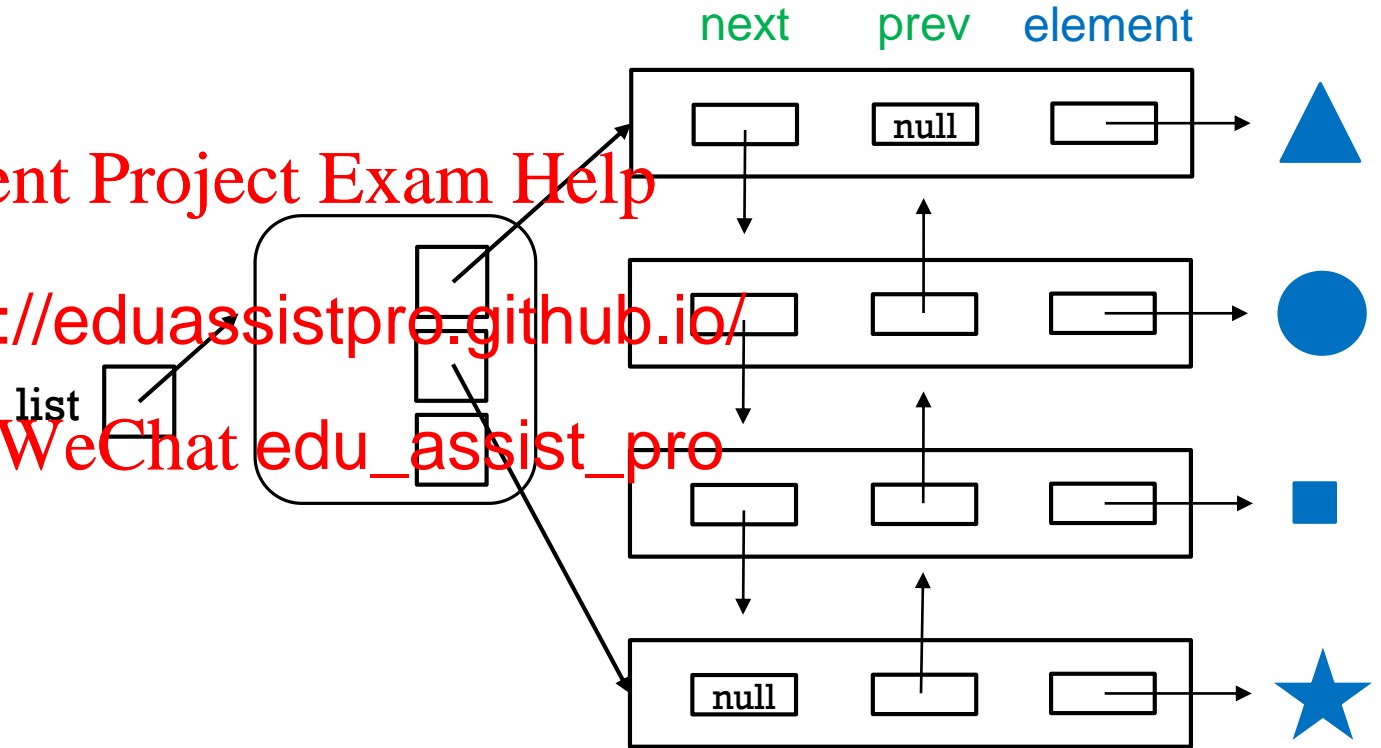
```
public class DLinkedList {  
    private DNode head;  
    private DNode tail;  
    private int size;  
    :  
    private class DNode {  
        Shape element;  
        DNode next;  
        DNode prev;  
    }  
}
```

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

Assignment Project Exam Help

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

Add WeChat edu_assist_pro



DOUBLY LINKED LIST – removeLast()

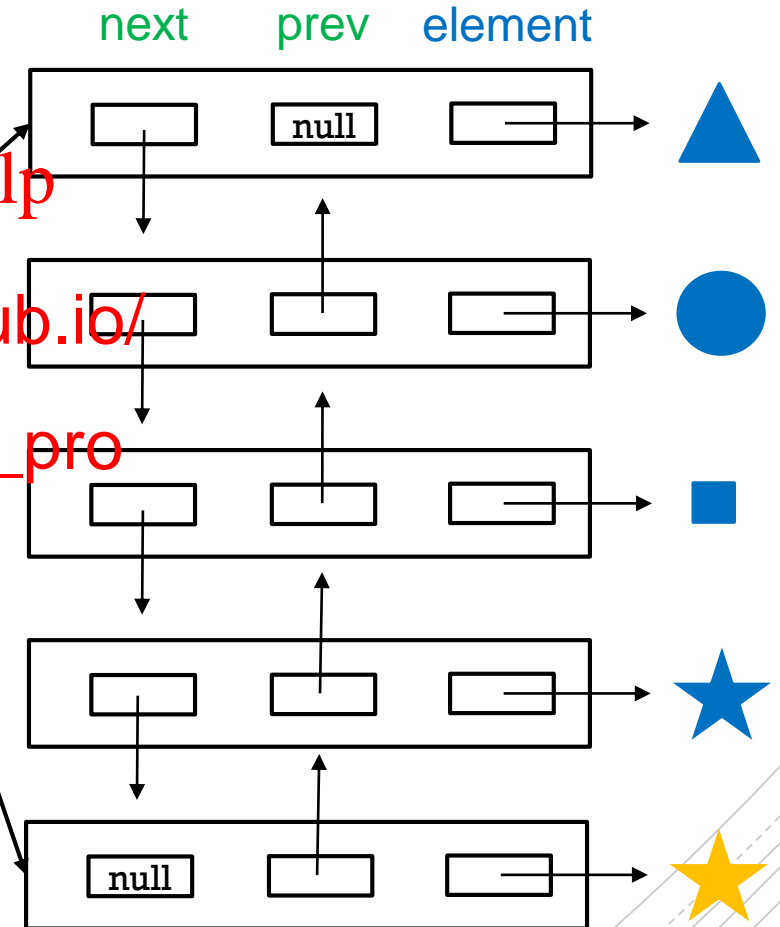
```
tail = tail.prev;  
tail.next.prev = null; // not necessary  
tail.next = null;  
size = size - 1;  
// to return the element,  
// you need to do a bit more work  
// edge cases for size = 0 and 1 to be added
```

Assignment Project Exam Help

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

Add WeChat edu_assist_pro

list



DOUBLY LINKED LIST – removeLast()

```
tail = tail.prev;  
tail.next.prev = null; // not necessary  
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
```

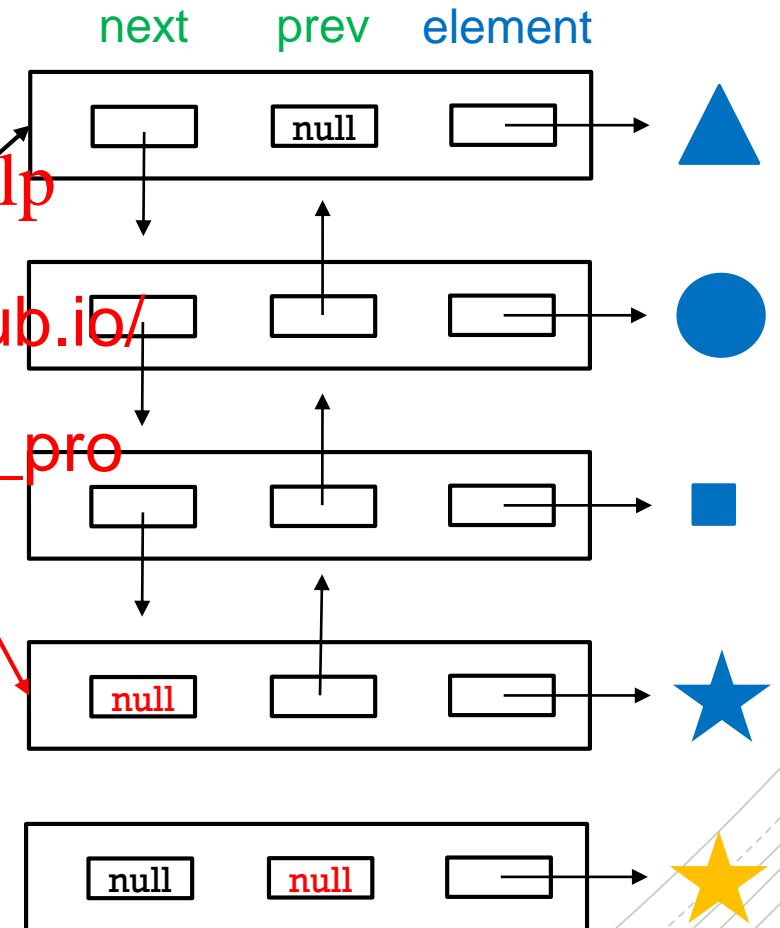
For a doubly linked list, removing the last element is much faster.

Assignment Project Exam Help

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

Add WeChat edu_assist_pro

list



WORSE CASE TIME COMPLEXITY (N = LIST SIZE)

	array list	SLinkedList	DLinkedList
addFirst()			O(1)
removeFirst()			O(1)
addLast()	O(1)	O(1)	O(1)
removeLast()	O(1)	O(N)	O(1)

Assignment Project Exam Help

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

Add WeChat edu_assist_pro

OTHER LIST OPERATIONS

Many list operations require access to a specific node i

Assignment Project Exam Help

`get(i)`

`set(i,e)`

`add(i,e)`

`remove(i)`

:

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

Add WeChat edu_assist_pro

LINKED LISTS

Suppose we want to access general node i in a linked list.

Two issues arise: **Assignment Project Exam Help**

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

- Edge cases ($i = 0, i = \text{size} - 1$) require special code.
This is a pain and can lead to costly bugs.
- How long does it take to access node i ?

AVOID EDGE CASES WITH "DUMMY NODES"

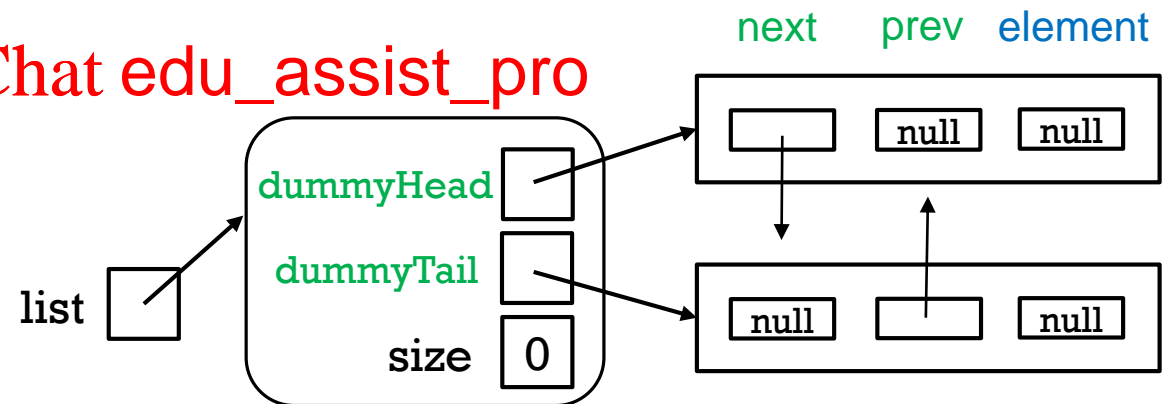
```
public class DLinkedList {  
    private DNode dummyHead;  
    private DNode dummyTail;  
    private int size;  
    :  
    public DLinkedList() {  
        dummyHead = new DNode();  
        dummyTail = new DNode();  
        dummyHead.next = dummyTail;  
        dummyTail.prev = dummyHead;  
        size = 0;  
    }  
}
```

```
// empty list  
DLinkedList list = new DLinkedList();
```

Assignment Project Exam Help

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

Add WeChat edu_assist_pro



AVOID EDGE CASES WITH "DUMMY NODES"

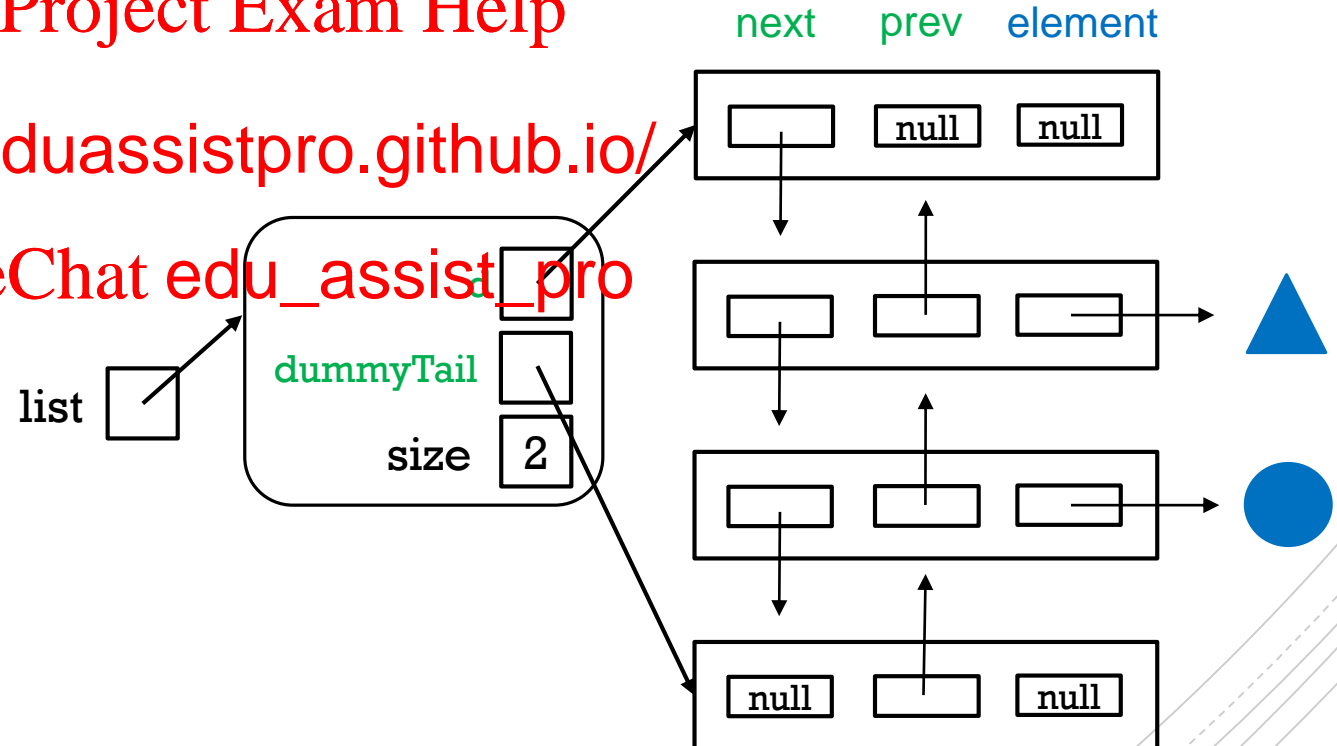
```
public class DLinkedList {  
    private DNode dummyHead;  
    private DNode dummyTail;  
    private int size;  
    :  
    public DLinkedList() {  
        dummyHead = new DNode();  
        dummyTail = new DNode();  
        dummyHead.next = dummyTail;  
        dummyTail.prev = dummyHead;  
        size = 0;  
    }  
}
```

```
DLinkedList list = new DLinkedList();  
// add 2 elements...
```

Assignment Project Exam Help

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

Add WeChat edu_assist_pro



HOW DO WE ACCESS A NODE? – get()

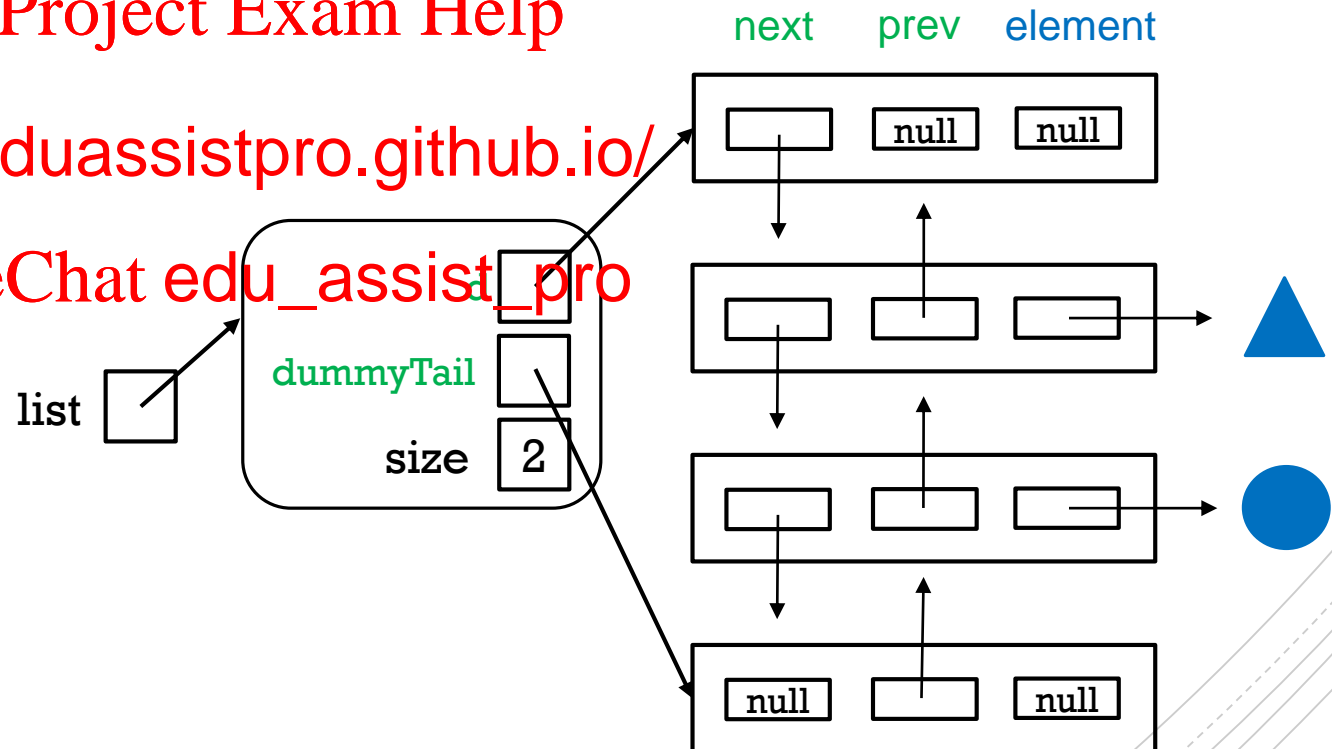
```
public Shape get(int i) {  
    DNode node = getNode(i);  
    return node.element;  
}
```

```
private DNode getNode(int i) {  
    // verify that 0<=i<size omitted  
    DNode node = dummyHead.next;  
    for(int k=0; k<i; k++)  
        node = node.next;  
    return node;  
}
```

Assignment Project Exam Help

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

Add WeChat edu_assist_pro



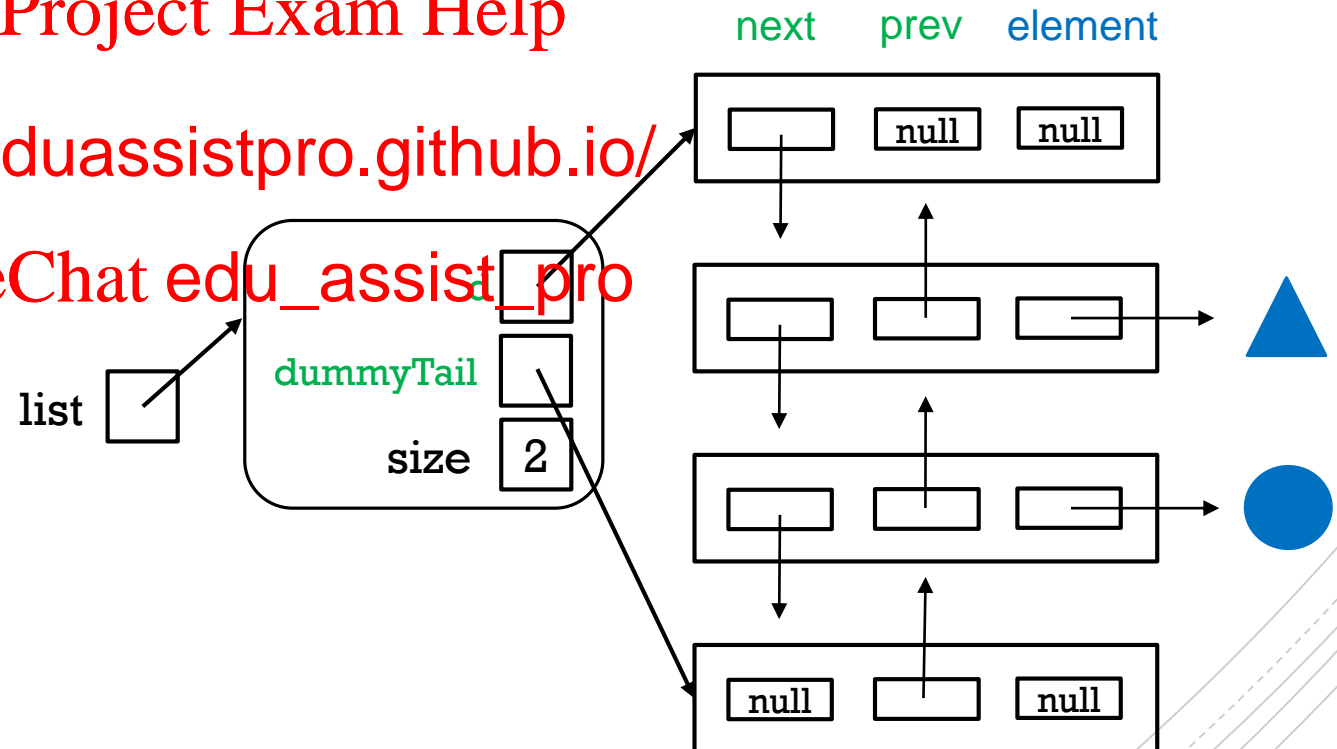
CAN WE SPEED THIS UP? – getNode()

```
private DNode getNode(int i) {  
    // verify that 0<=i<size omitted  
    DNode node;  
    if (i < size/2) {  
        node = dummyHead.next;  
        for(int k=0; k<i; k++)  
            node = node.next;  
    }  
    else {  
        node = dummyTail.prev;  
        for(int k=size -1; k>i; k--)  
            node = node.prev;  
    }  
    return node;  
}
```

Assignment Project Exam Help

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

Add WeChat edu_assist_pro



JAVA LINKEDLIST CLASS

<https://docs.oracle.com/javase/8/docs/api/java/util/LinkedList.html>

It uses a *doubly linked list* as the underlying data structure.

Assignment Project Exam Help

It has some methods that <https://eduassistpro.github.io/>

- addFirst()
- removeFirst()
- addLast()
- removeLast()

Add WeChat edu_assist_pro

Why ?

Q: What is the time complexity of the following ?

```
DLinkedList list = new DLinkedList( ) ;
```

```
for (k = 0; k < list.size(); k++) {  
    // some operation  
}
```

s some constant

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

Add WeChat edu_assist_pro

Q: What is the time complexity of the following ?

```
DLinkedList list = new DLinkedList( ) ;
```

```
for (k = 0; k < list.size(); k++) {  
    // some constant time operation  
}
```

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

Add WeChat edu_assist_pro

A: $1 + 1 + 1 + \dots + 1 = N \Rightarrow O(N)$

where '1' means constant.

Q: What is the time complexity of the following ?

```
for (k = 0; k < list.size(); k++) // size == N
```

Assignment Project Exam Help

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

Assuming here that getNode(1) is at the head.

Add WeChat edu_assist_pro

Q: What is the time complexity of the following ?

```
for (k = 0; k < list.size(); k++) // size == N
```

Assignment Project Exam Help

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

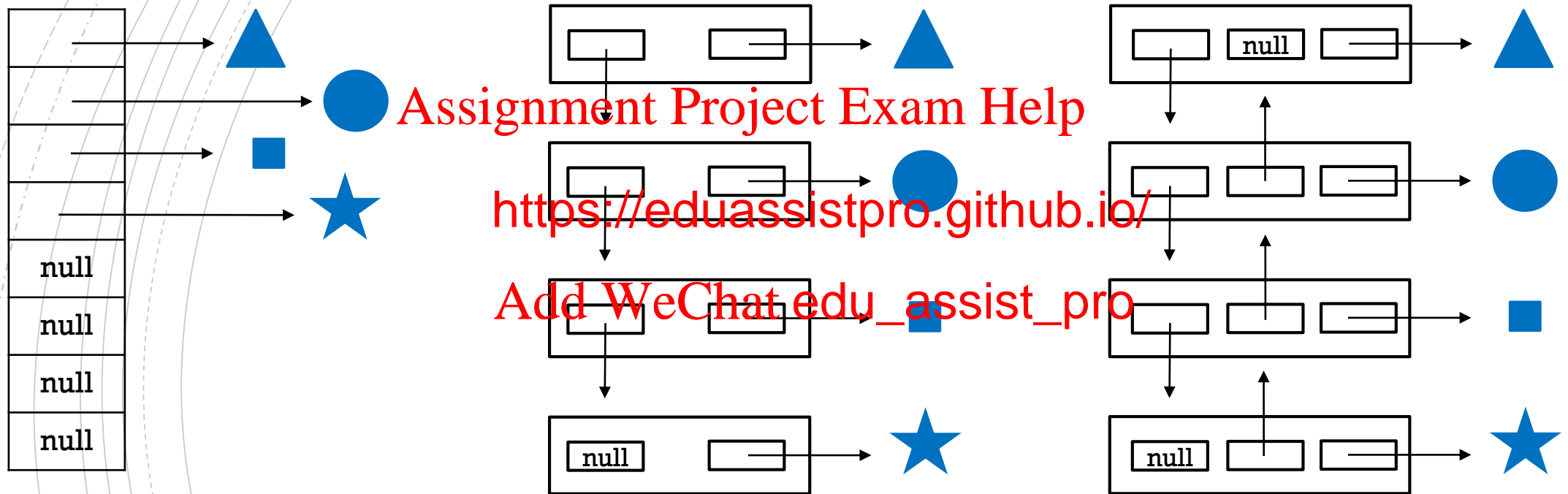
Assuming here that `getNode(1)` at the head.

A: 1 + 2 + 3 + N

$$= \frac{N(N+1)}{2} \Rightarrow O(N^2)$$

In 3 weeks we'll talk about a more efficient way to iterate through elements in a (Java) LinkedList!

WHAT ABOUT "SPACE COMPLEXITY" ?



All three data structures use space $O(N)$ for a list of size N .
But linked lists use $2x$ (single) or $3x$ (double).

ARRAY LIST VERSUS LINKED LIST ?

Array lists and linked lists both take $O(N)$ time to add or remove from an arbitrary position in the list.

In practice and with linked lists are faster. But the reasons are subtle and have to do with how computer memory works, in particular, contiguous memory allocation. You will learn about that topic in COMP 273.

Assignment Project Exam Help

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

Add WeChat edu_assist_pro

DO YOU EVER NEED LINKED LISTS ?

Yes. Even if you prefer ArrayLists, you still need to understand Li are special cases of a general a <https://eduassistpro.github.io/> structure called a tree which we will be discussing.

Add WeChat edu_assist_pro

An orange paint roller with a red handle, positioned horizontally. The roller is partially filled with orange paint, and there are orange paint splatters and drips around it. The text "Coming Soon" is written in white on the orange part of the roller.

Coming Soon

Assignment Project Exam Help

In the next

- **Quadrat**
- **Asymptotic notations**

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

Add WeChat **edu_assist_pro**