

COMP 250

Assignment Project Exam Help

INTRODUCER SCIENCE

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

Week 1-2:
Add WeChat edu_assist_pro

Giulia Alberini, Fall 2020
Slides adapted from Michael Langer's

WHAT ARE WE GOING TO DO IN THIS VIDEO?



- **Queues**

Assignment Project Exam Help

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

Add WeChat edu_assist_pro

ADT (ABSTRACT DATA TYPE)

- List

`add(i, e), remove(i), get(i), set(i), ..`

- Stack

`push(e), pop(), ..`

- Queue

`enqueue(e), dequeue()`

Assignment Project Exam Help

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

Add WeChat edu_assist_pro

QUEUE

dequeue
(remove from front)

enqueue
(add at back)

Assignment Project Exam Help

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

Add WeChat edu_assist_pro



e.g. Server



clients

EXAMPLES

- keyboard buffer

- CPU processes

Assignment Project Exam Help
(app rallel)

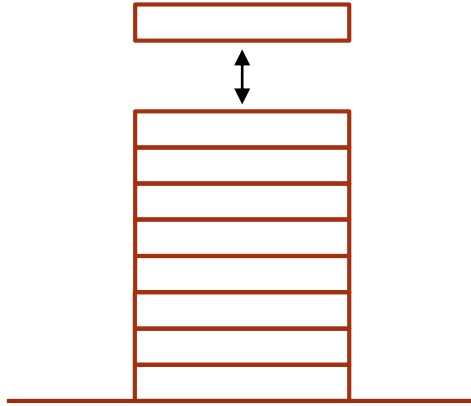
<https://eduassistpro.github.io/>

- web server

Add WeChat edu_assist_pro

- students in the zoom waiting room

- ...



Stack

Assignment Project Exam Help

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

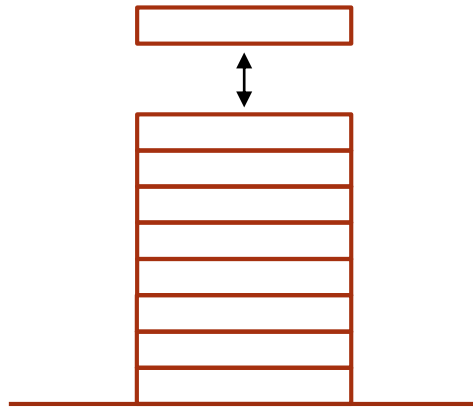
push(e)

Add WeChat edu_assist_pro

pop()

LIFO

(last in, first out)



Stack

push(e)

pop()

LIFO

(last in, first out)



Queue

enqueue(e)

dequeue()

FIFO

(first in, first out)

“first come, first serve”

Assignment Project Exam Help

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

Add WeChat edu_assist_pro

QUEUE EXAMPLE

enqueue(a)

enqueue(b)

dequeue()

a

a b

Assignment Project Exam Help

<https://eduassistpro.github.io/> *returns a*

Add WeChat edu_assist_pro

time

QUEUE EXAMPLE

enqueue(a)

enqueue(b)

dequeue()

enqueue(c)

enqueue(d)

enqueue(e)

Assignment Project Exam Help

<https://eduassistpro.github.io/> *returns a*

Add WeChat edu_assist_pro

a

a b

bcd

bcde

time

QUEUE EXAMPLE

enqueue(a)

enqueue(b)

dequeue()

enqueue(c)

enqueue(d)

enqueue(e)

dequeue()

enqueue(f)

enqueue(g)

Assignment Project Exam Help

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

Add WeChat edu_assist_pro

a

ab

returns a

bcde

cde

returns b

cdef

cdefg

time

HOW TO IMPLEMENT A QUEUE?

enqueue(e)

dequeue()

singly linked li

doubly linked list

array list

Assignment Project Exam Help

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

Add WeChat edu_assist_pro

HOW TO IMPLEMENT A QUEUE?

enqueue(e)

dequeue()

singly linked li

removeFirst()

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

doubly linked list

Add WeChat edu_assist_pro

array list

HOW TO IMPLEMENT A QUEUE?

enqueue(e)

dequeue()

singly linked li

removeFirst()

doubly linked list

array list

Assignment Project Exam Help

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

Add WeChat edu_assist_pro

(same, or

& removeLast())

HOW TO IMPLEMENT A QUEUE?

enqueue(e)

dequeue()

singly linked li

removeFirst()

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

doubly linked list

Add WeChat edu_assist_pro

(same, or

& removeLast())

array list

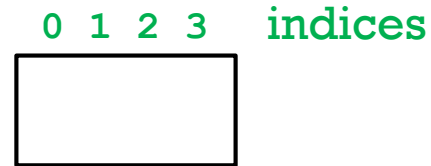
addLast(e)

removeFirst()

SLOW!

IMPLEMENTING A QUEUE WITH AN ARRAY LIST (BAD)

length = 4



enqueue (a)

enqueue (b

dequeue ()

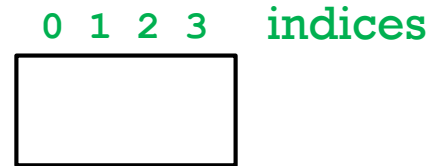
Assignment Project Exam Help

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

Add WeChat edu_assist_pro

IMPLEMENTING A QUEUE WITH AN ARRAY LIST (BAD)

length = 4



```
enqueue ( a )  
enqueue ( b )  
dequeue ( )  
enqueue ( c )  
enqueue ( d )  
enqueue ( e )  
dequeue ( )
```

Assignment Project Exam Help

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

Requires shift

Add WeChat edu_assist_pro

bc

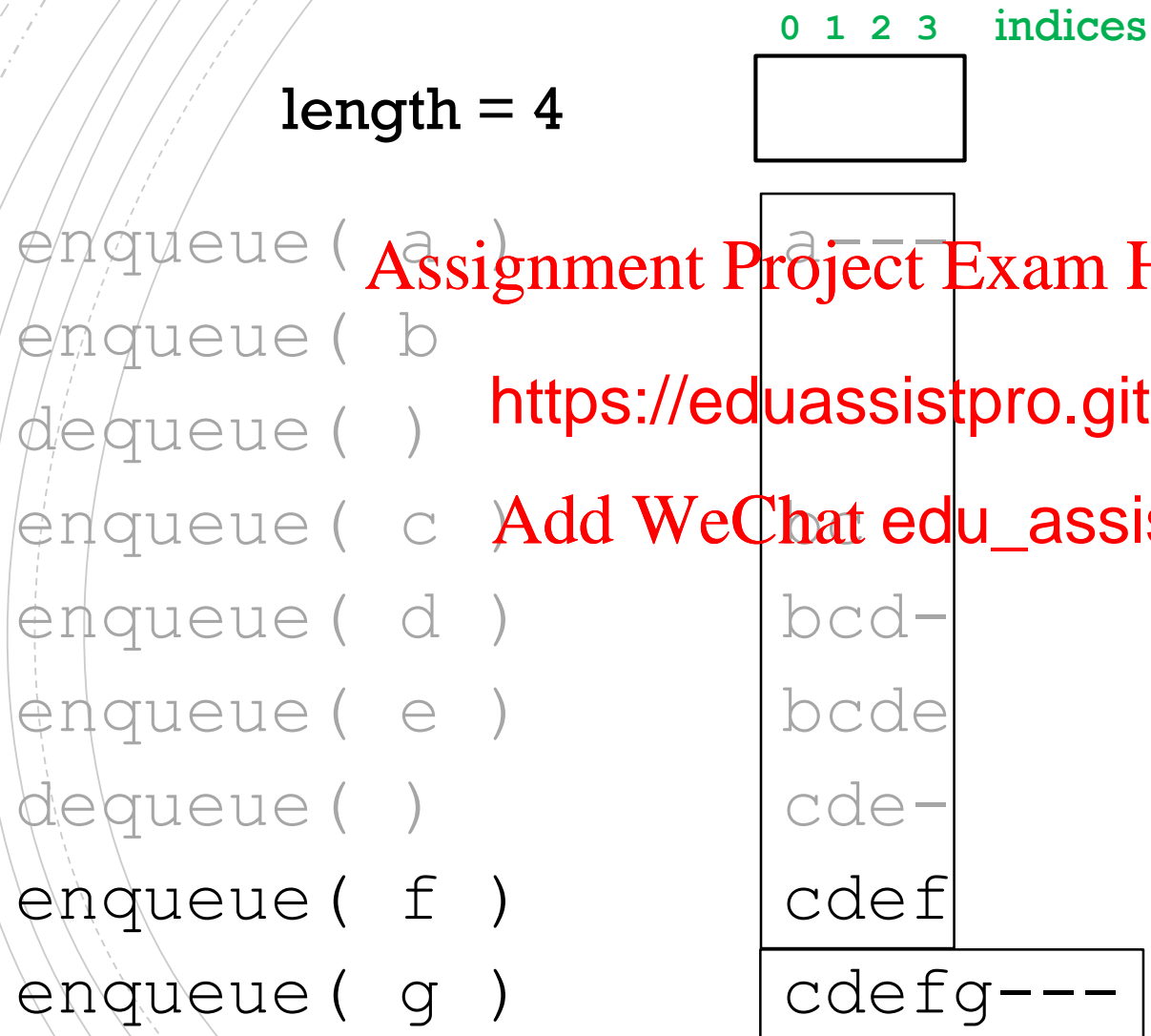
bcd-

bcde

cde-

Requires shift

IMPLEMENTING A QUEUE WITH AN ARRAY LIST (BAD)



Assignment Project Exam Help

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

Add WeChat edu_assist_pro

requires expansion

IMPLEMENTING A QUEUE WITH AN EXPANDING ARRAY (ALSO BAD)

Use **head** and **tail** indices
(**tail** = **head** + size - 1)

```
enqueue ( a )
enqueue ( b )
dequeue ( )
enqueue ( c )
enqueue ( d )
enqueue ( e )
```

0 1 2 3

a	-	-	-

(0, 0)

, 1)

, 1)

, 2)

(1, 3)

Assignment Project Exam Help

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

Add WeChat edu_assist_pro

?

IMPLEMENTING A QUEUE WITH AN EXPANDING ARRAY (ALSO BAD)

Use **head** and **tail** indices

(**tail** = **head** + size - 1)

	0	1	2	3	
enqueue (a)	a	-	-	-	(0, 0)
enqueue (b)		b	-	-	(0, 1)
dequeue ()					(1, 1)
enqueue (c)		c	-	-	(1, 2)
enqueue (d)			d	-	(1, 3)
enqueue (e)				e	(1, 4)
dequeue ()					(2, 4)
enqueue (f)				f	(2, 5)
enqueue (g)					(2, 6)

Assignment Project Exam Help

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

Add WeChat edu_assist_pro

Make
bigger
array and
copy to it.

IMPLEMENTING A QUEUE WITH AN EXPANDING ARRAY

- to dequeue: retrieve the element at head and increase the index head

Assignment Project Exam Help

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

- to enqueue: add the element at tail

Add WeChat edu_assist_pro

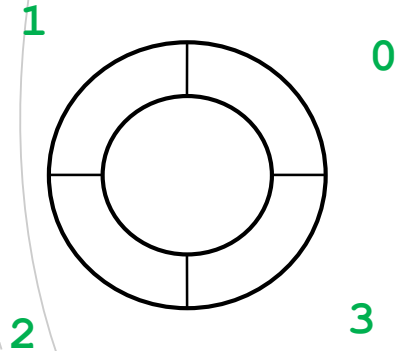
An expanding array is an inefficient usage of space!

A better idea is...

CIRCULAR ARRAY

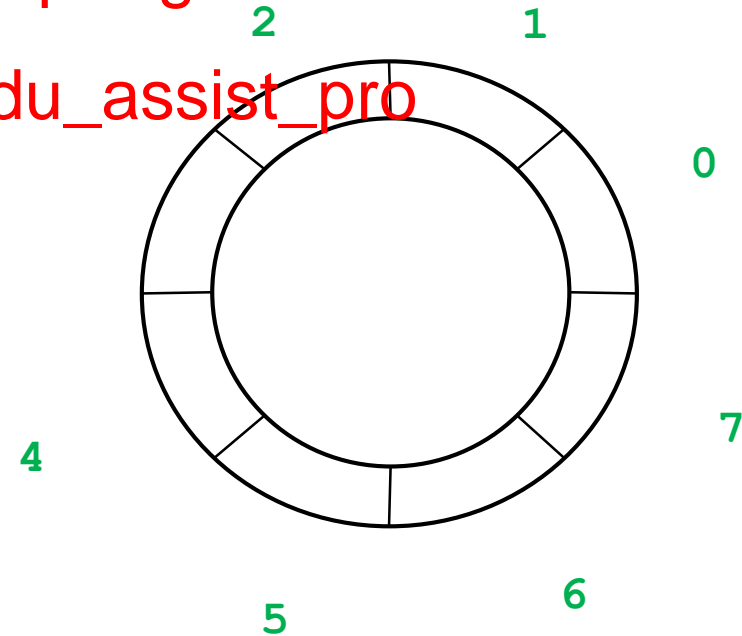
length = 4

0123



length = 8

01234567



Assignment Project Exam Help

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

Add WeChat edu_assist_pro

IMPLEMENTING A QUEUE WITH A CIRCULAR ARRAY (GOOD)

$$\text{tail} = (\text{head} + \text{size} - 1) \bmod \text{length}$$

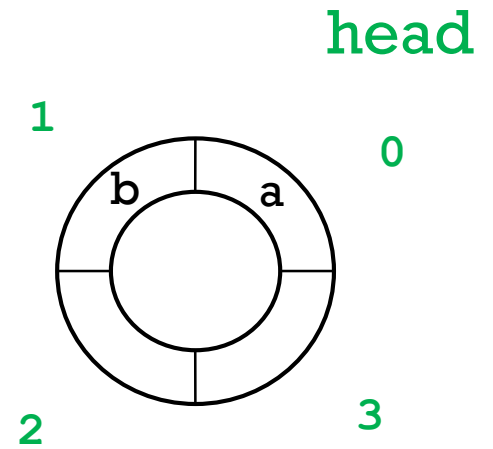
```
enqueue( a )  
enqueue( b )  
dequeue() ?
```

Assignment Project Exam Help

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

Add WeChat edu_assist_pro

head=0 tail=1



IMPLEMENTING A QUEUE WITH A CIRCULAR ARRAY (GOOD)

$$\text{tail} = (\text{head} + \text{size} - 1) \bmod \text{length}$$

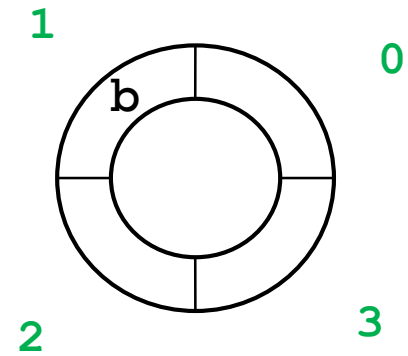
```
enqueue( a )  
enqueue( b )  
dequeue()  
enqueue( c ) ?
```

Assignment Project Exam Help

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

Add WeChat edu_assist_pro

head=1 tail=1



IMPLEMENTING A QUEUE WITH A CIRCULAR ARRAY (GOOD)

$\text{tail} = (\text{head} + \text{size} - 1) \bmod \text{length}$

```
enqueue ( a )  
enqueue ( b )  
dequeue ()  
enqueue ( c )  
enqueue ( d ) ?
```

Assignment Project Exam Help

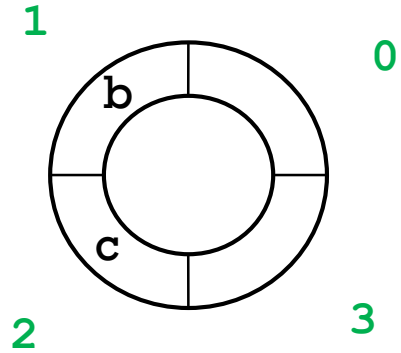
<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

head=1 tail=2

head

tail



IMPLEMENTING A QUEUE WITH A CIRCULAR ARRAY (GOOD)

$$\text{tail} = (\text{head} + \text{size} - 1) \bmod \text{length}$$

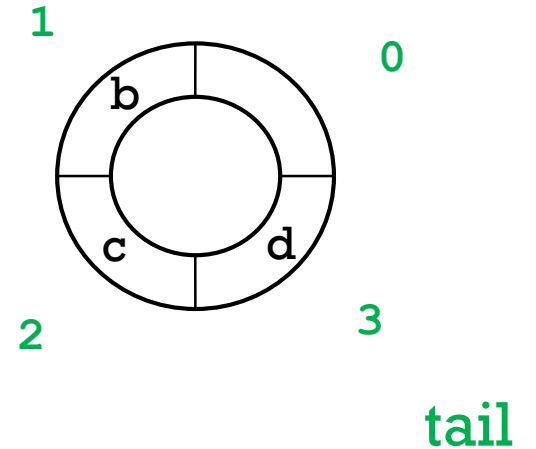
```
enqueue( a )  
enqueue( b )  
dequeue( )  
enqueue( c )  
enqueue( d )  
enqueue( e ) ?
```

Assignment Project Exam Help

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

Add WeChat edu_assist_pro

head=1 tail=3



IMPLEMENTING A QUEUE WITH A CIRCULAR ARRAY (GOOD)

$$\text{tail} = (\text{head} + \text{size} - 1) \bmod \text{length}$$

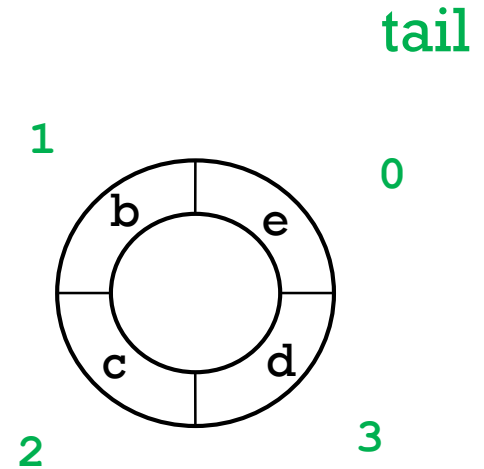
```
enqueue ( a )  
enqueue ( b )  
dequeue ()  
enqueue ( c )  
enqueue ( d )  
enqueue ( e )  
dequeue () ?
```

Assignment Project Exam Help

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

Add WeChat edu_assist_pro

head=1 tail=0



IMPLEMENTING A QUEUE WITH A CIRCULAR ARRAY (GOOD)

$$\text{tail} = (\text{head} + \text{size} - 1) \bmod \text{length}$$

```
enqueue ( a )  
enqueue ( b )  
dequeue ()  
enqueue ( c )  
enqueue ( d )  
enqueue ( e )  
dequeue ()
```

Assignment Project Exam Help

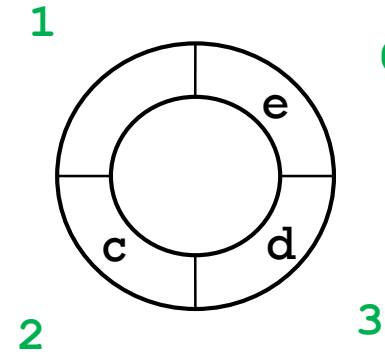
<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

head=2 tail=0

head

tail



IMPLEMENTING A QUEUE WITH A CIRCULAR ARRAY (GOOD)

$\text{tail} = (\text{head} + \text{size} - 1) \bmod \text{length}$

```
enqueue ( a )  
enqueue ( b )  
dequeue ()  
enqueue ( c )  
enqueue ( d )  
enqueue ( e )  
dequeue ()  
enqueue ( f )  
enqueue ( g ) ?
```

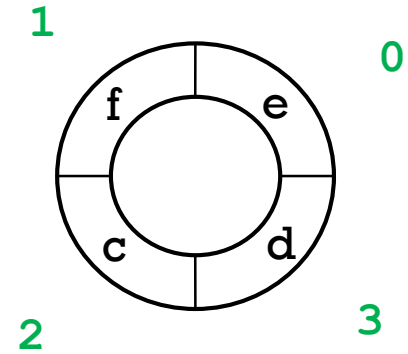
Assignment Project Exam Help

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

Add WeChat edu_assist_pro

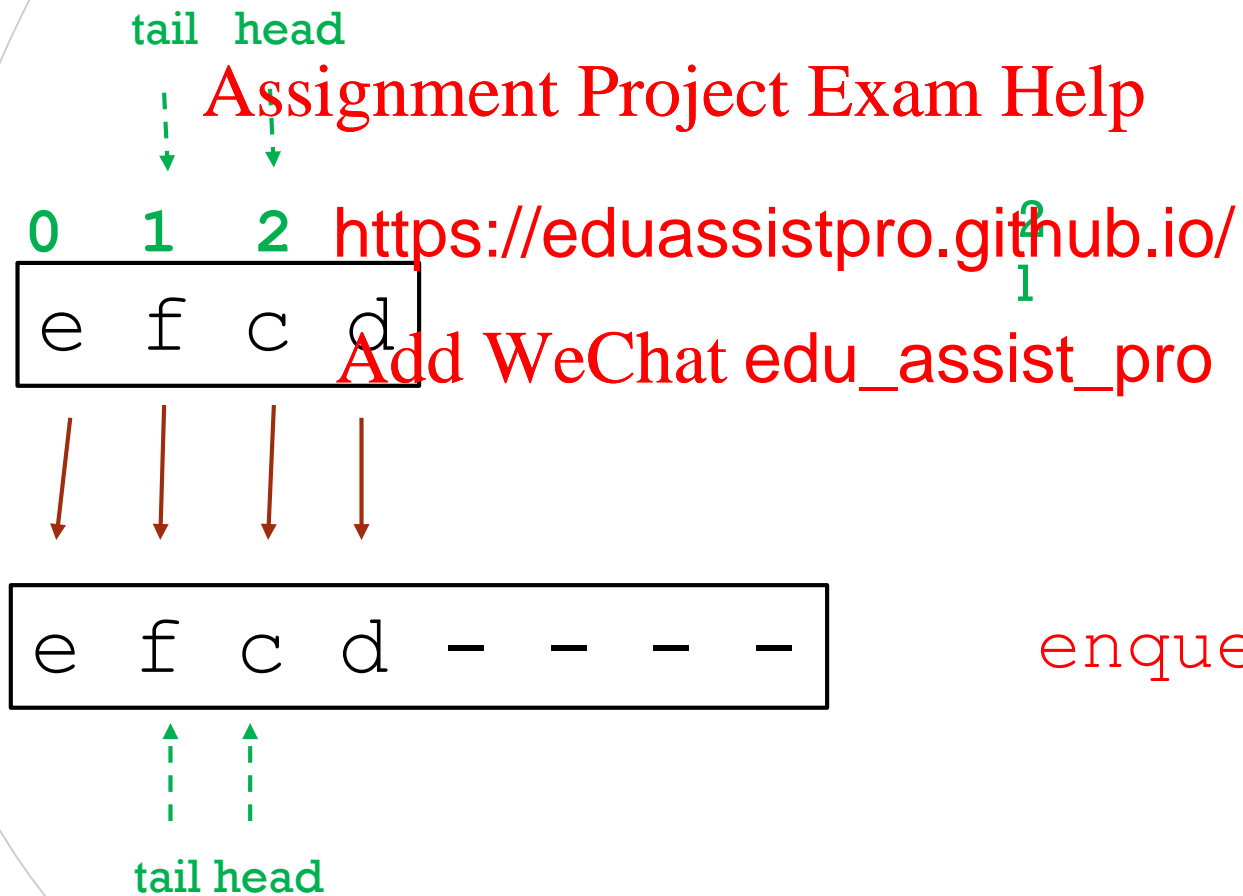
head=2 tail=1

head



INCREASE THE LENGTH OF THE ARRAY AND COPY

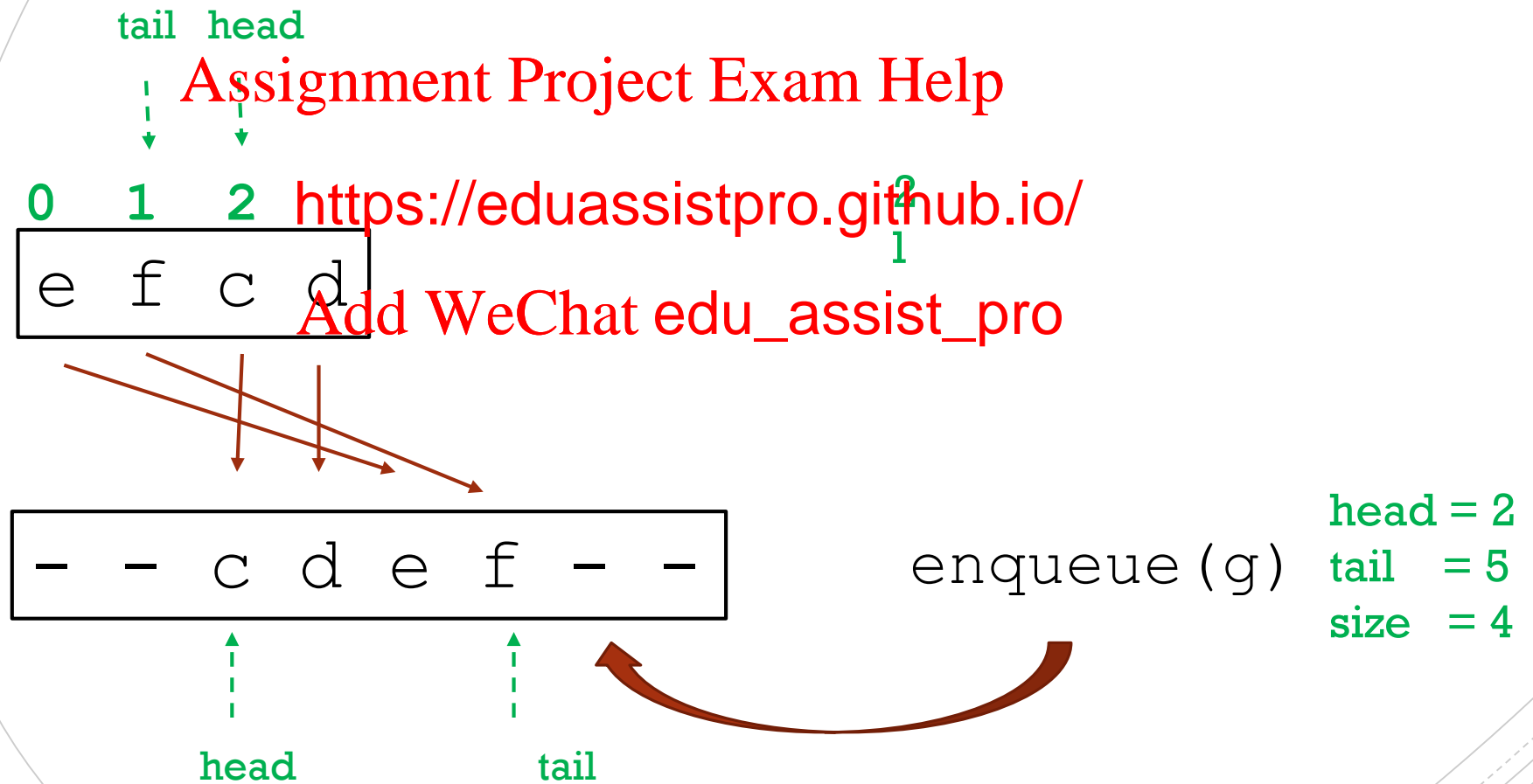
Be careful, on how you copy! The following would not work:



enqueue(g) ?

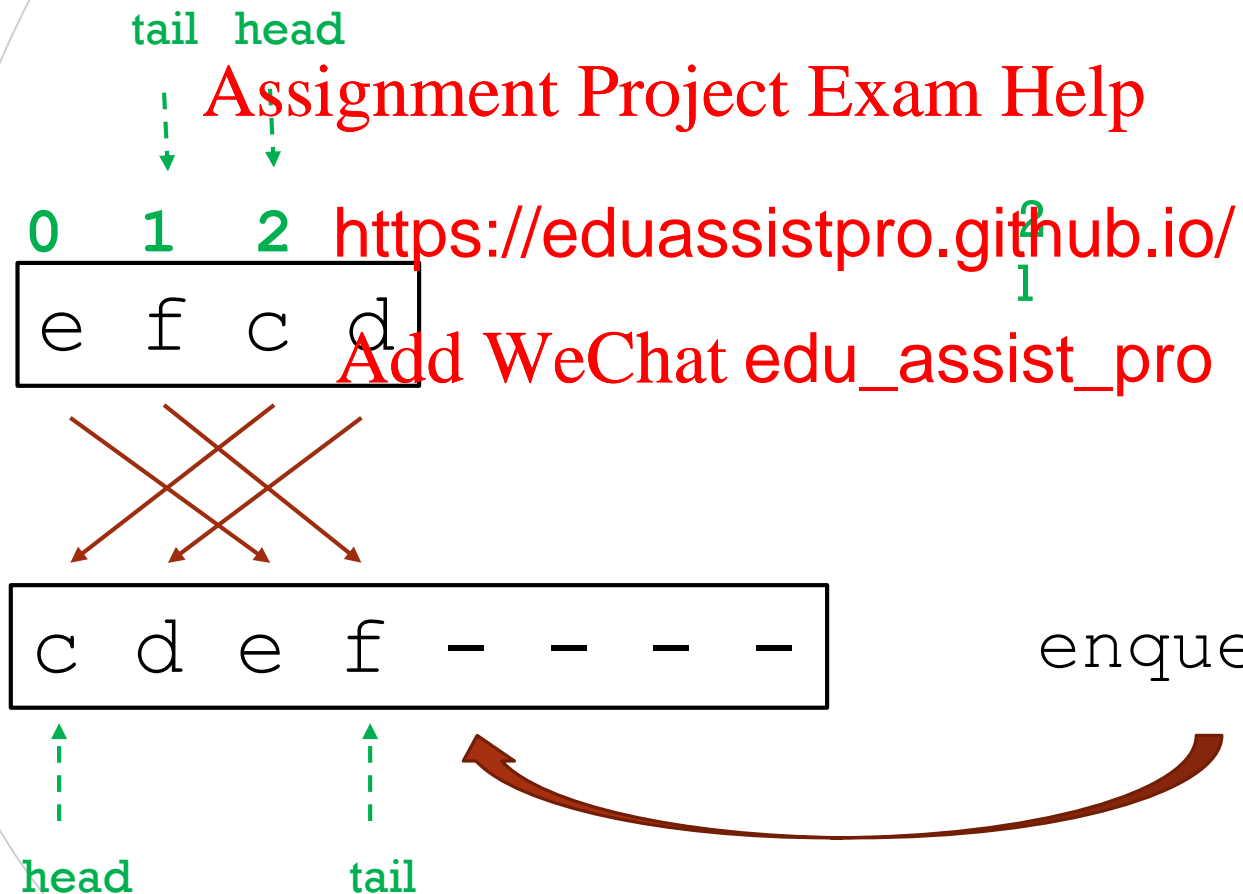
INCREASE THE LENGTH OF THE ARRAY AND COPY

Instead you can copy so that the head remains in the same position.



INCREASE THE LENGTH OF THE ARRAY AND COPY

OR you can copy so that the head moves to position 0.



head = 0
tail = 3
size = 4

Assignment Project Exam Help

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

Add WeChat edu_assist_pro

ENQUEUE(e)

```
enqueue( element ){
    if (size == queue.length) {
        // increase length of array
        create a bigger array e.g. 2*length
        for i = 0 to queue.length-1
            tmp[i] = queue[(head + i) % queue.length]
        head = 0
        queue = tmp
    }
    queue[(head + size) % length] = element
    size++
}
```

Assignment Project Exam Help
<https://eduassistpro.github.io/>
Add WeChat edu_assist_pro

Note that we don't have a tail variable here. Instead, recall that $tail = (head + size - 1) \bmod length$, and note that the new element is added in position $(tail + 1) \bmod length$.

DEQUEUE()

```
dequeue() {  
    if size <= 0  
        ra  
  
    element = queue[head]  
    size = size - 1  
    head = (head + 1) mod length  
    return element  
}
```

Assignment Project Exam Help

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

Add WeChat edu_assist_pro

WHAT IF SIZE IS 0?

What is the relation between `head` and `tail` when size is equal to 0?

`tail = (head + size - 1) mod length`

`https://eduassistpro.github.io/`

Initial state `tail = 0`

Add WeChat `edu_assist_pro`

WHAT IF SIZE IS 0?

What is the relation between **head** and **tail** when size is equal to 0?

Assignment Project Exam Help
 $\text{tail} = (\text{head} + \text{size} - 1) \% \text{length}$

<https://eduassistpro.github.io/> **add tail size)**

Initial state **---** **3, 0)**
enqueue (a) **a---** **0, 1)**
enqueue (b) **ab--** **(0, 1, 2)**
dequeue () **-b--** **(1, 1, 1)**
dequeue () **--** **(2, 1, 0)**

tail **head**

ADT (ABSTRACT DATA TYPE)

Defines a data type by the values and operations from the user's perspective only. It ignores the details of the implementation.

Assignment Project Exam Help

Examples:

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

Add WeChat edu_assist_pro

- list
- stack
- queue
- ...

An orange paint roller with a red handle, positioned horizontally. The roller is partially 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

- Back to J <https://eduassistpro.github.io/>
Add WeChat edu_assist_pro