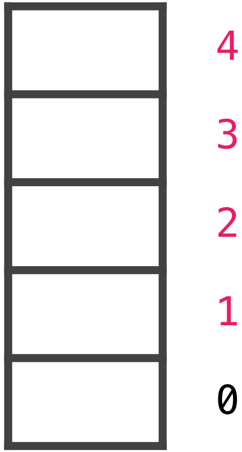
Lecture 6

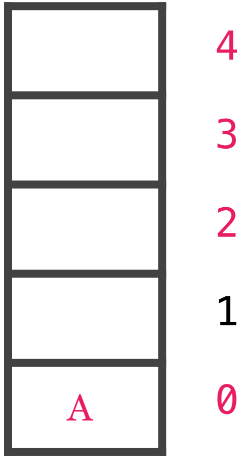
Queue, Stack ...

Stack

How push(element) in stack works

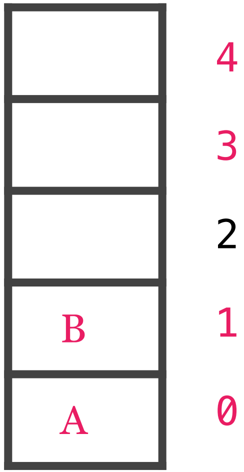


```
count = 0
```

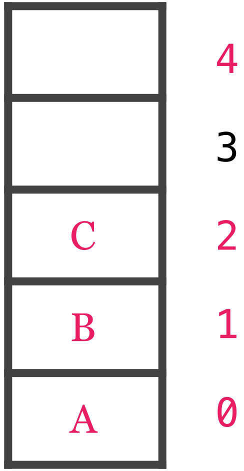


```
elements[count] = ‘A’
count = count + 1
```

How push(element) in stack works

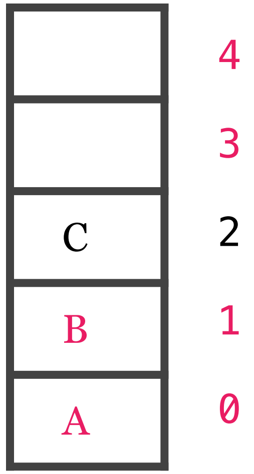


```
elements[count] = ‘B’
count = count + 1
```

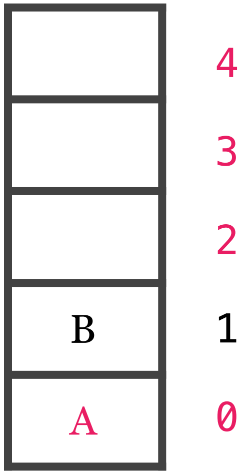


```
elements[count] = ‘C’
count = count + 1
```

How pop( ) in stack works

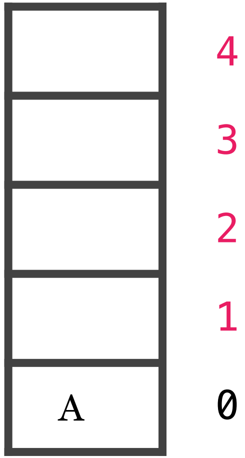


```
count = count - 1
return elements[count]
```

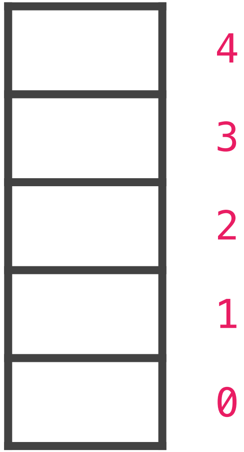


```
count = count - 1
return elements[count]
```

How pop( ) in stack works



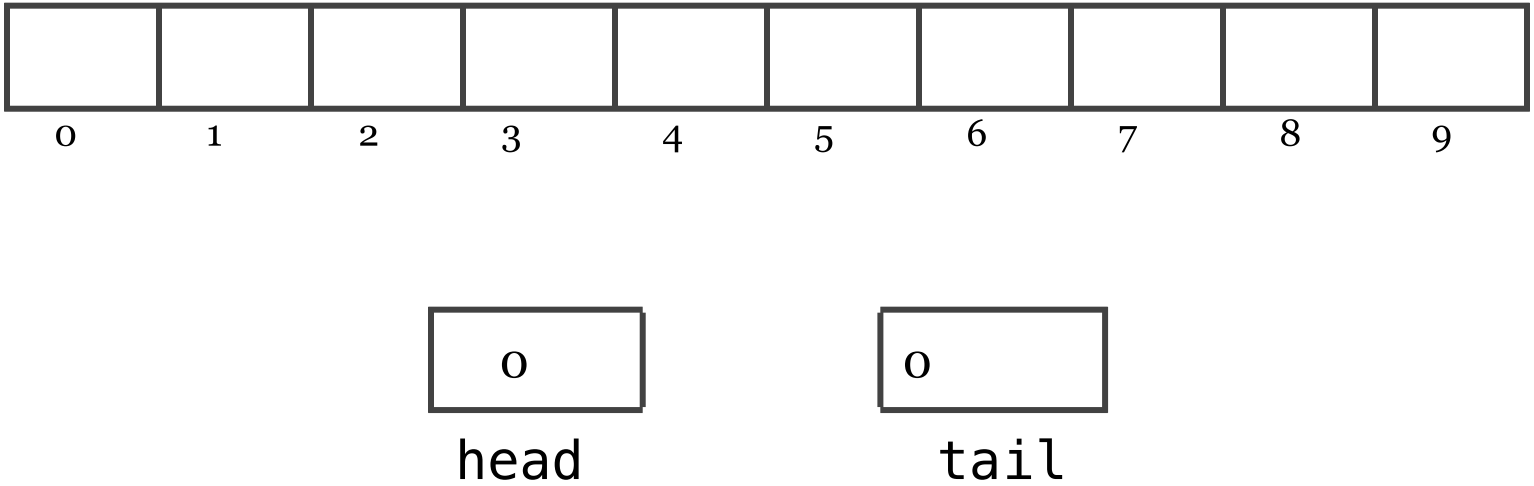
```
count = count – 1
return elements[count]
```



```
count = 0
```

Queue

How enqueue(element) works



0 1 2 3 4 5 6 7 8 9

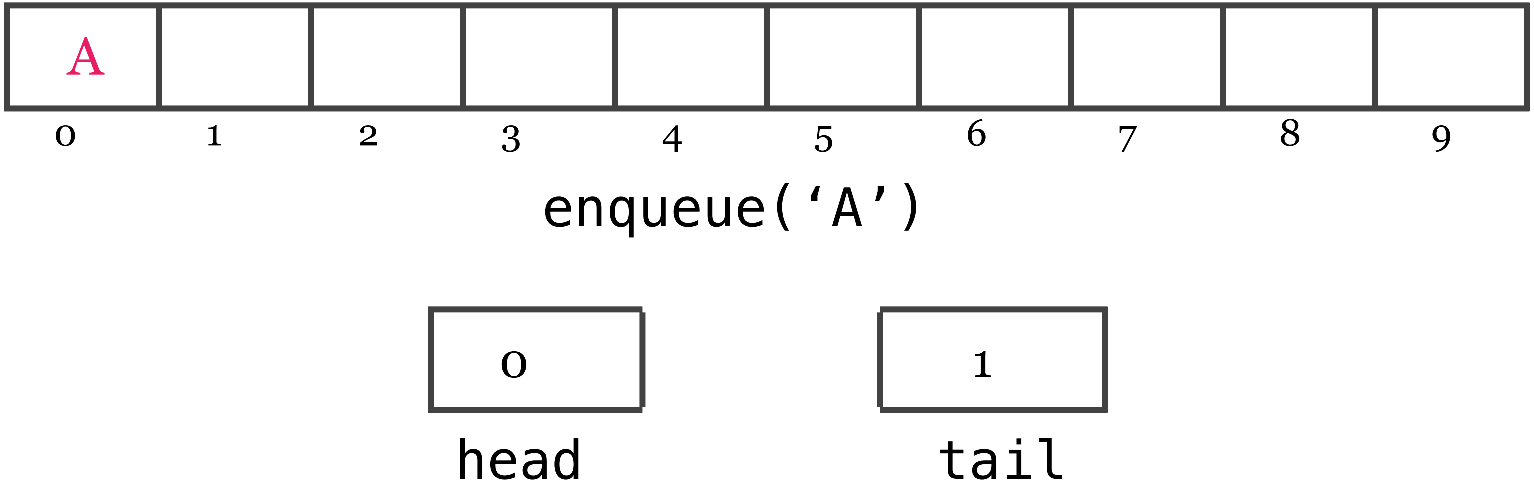
0

head

0

tail

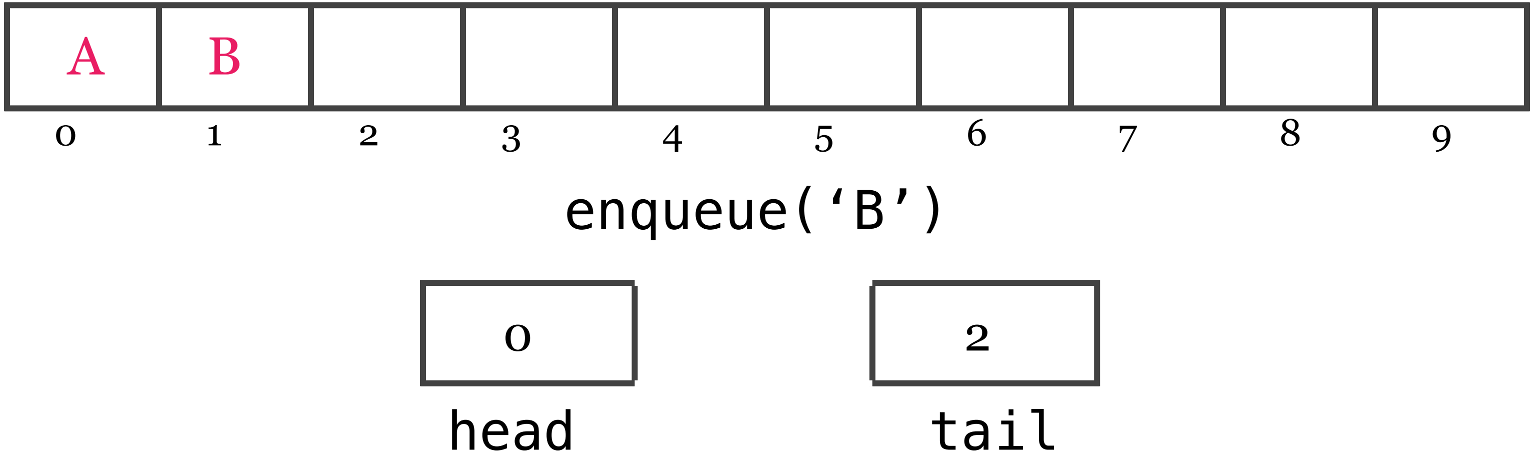
How enqueue(element) works



A									
0	1	2	3	4	5	6	7	8	9

```
enqueue(‘A’)
```

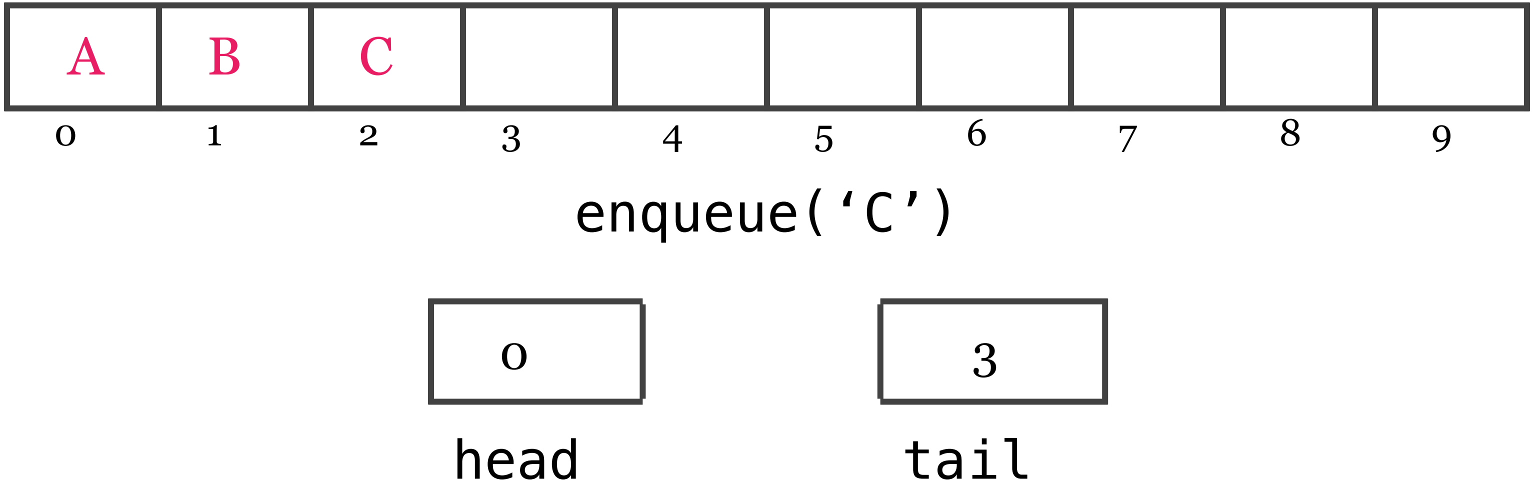
How enqueue(element) works



0	1	2	3	4	5	6	7	8	9
A	B								

enqueue(‘B’)

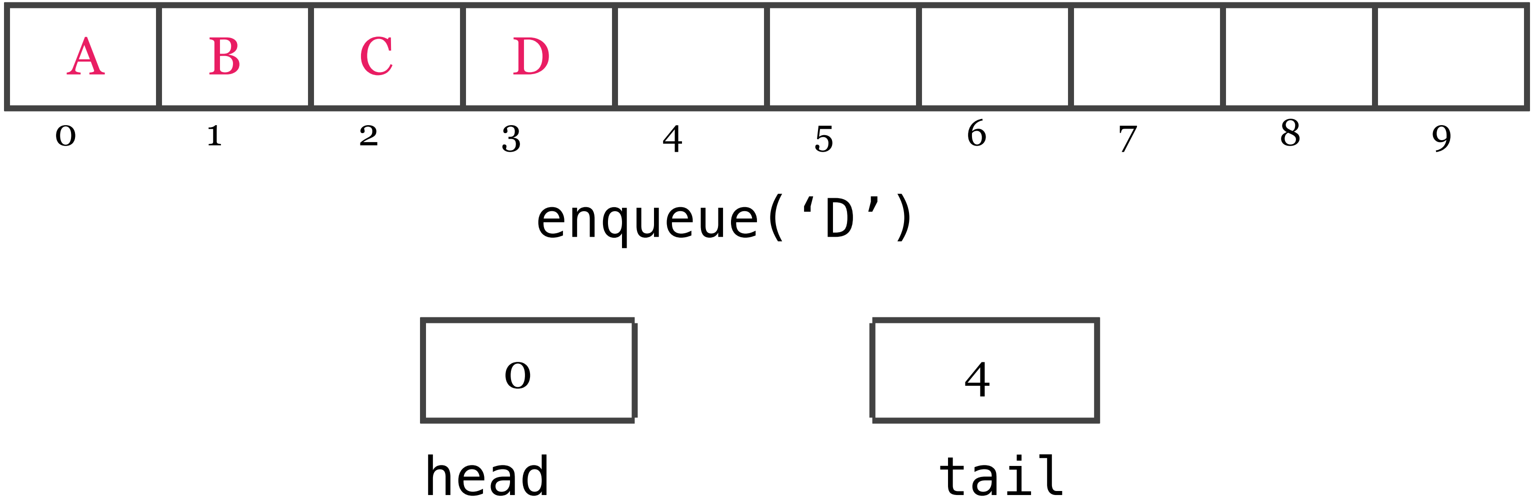
How enqueue(element) works



0	1	2	3	4	5	6	7	8	9
A	B	C							

`enqueue(‘C’)`

How enqueue(element) works

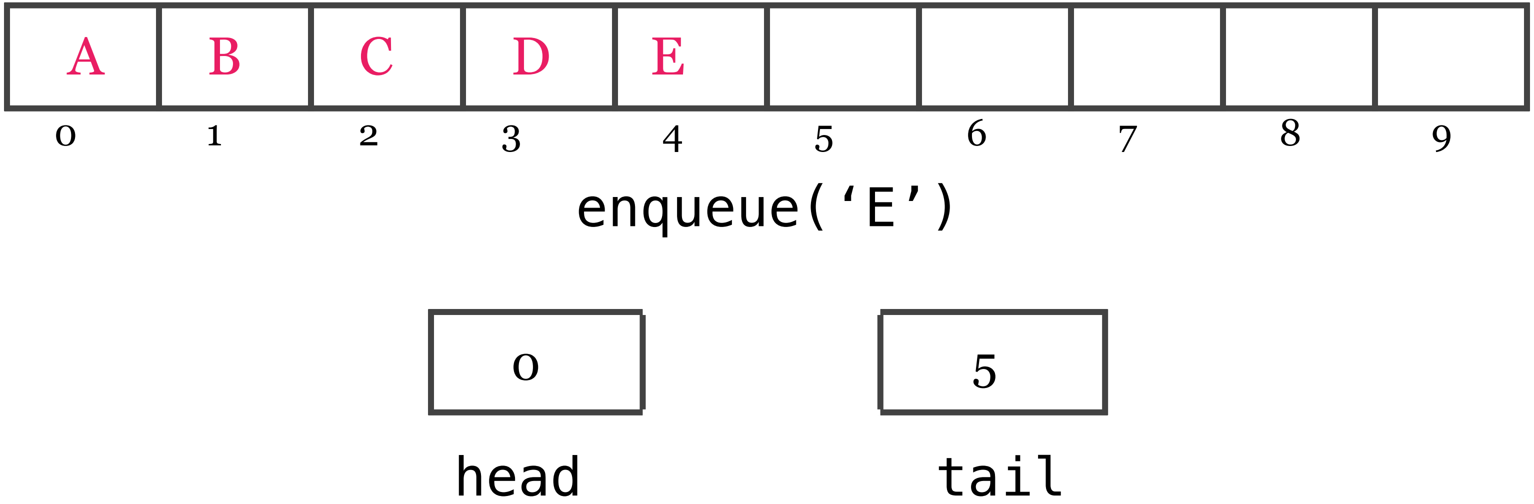


A	B	C	D						
0	1	2	3	4	5	6	7	8	9

enqueue(‘D’)

0	4
head	tail

How enqueue(element) works



A	B	C	D	E					
0	1	2	3	4	5	6	7	8	9

enqueue(‘E’)

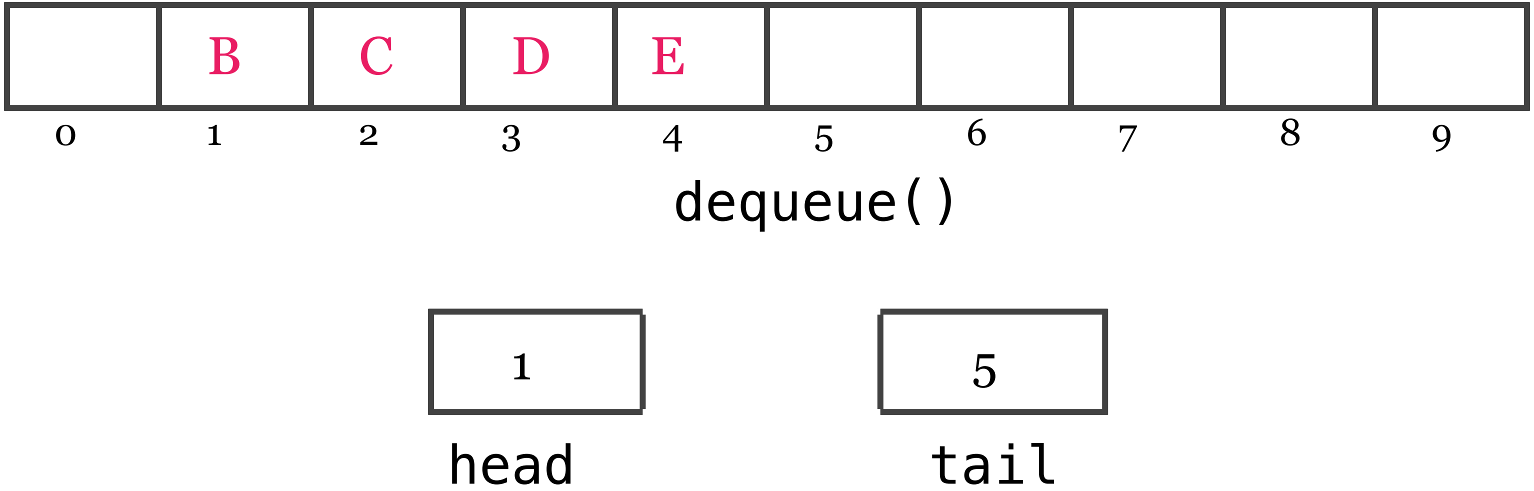
0

head

5

tail

How dequeue( ) works

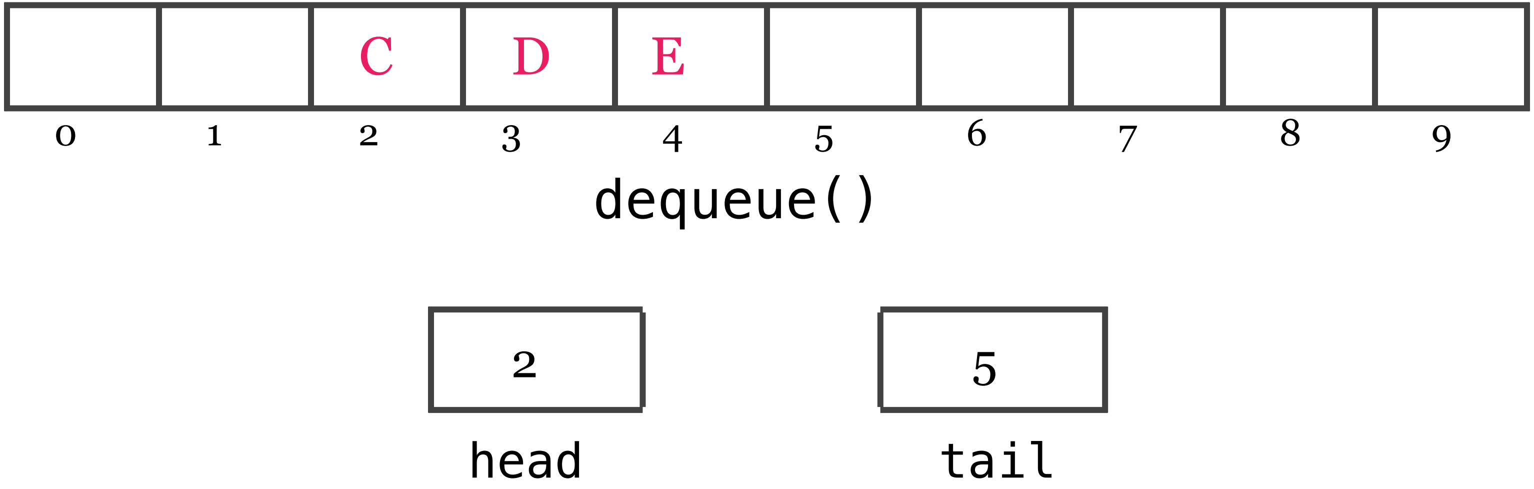


	B	C	D	E					
0	1	2	3	4	5	6	7	8	9

dequeue()

1	5
head	tail

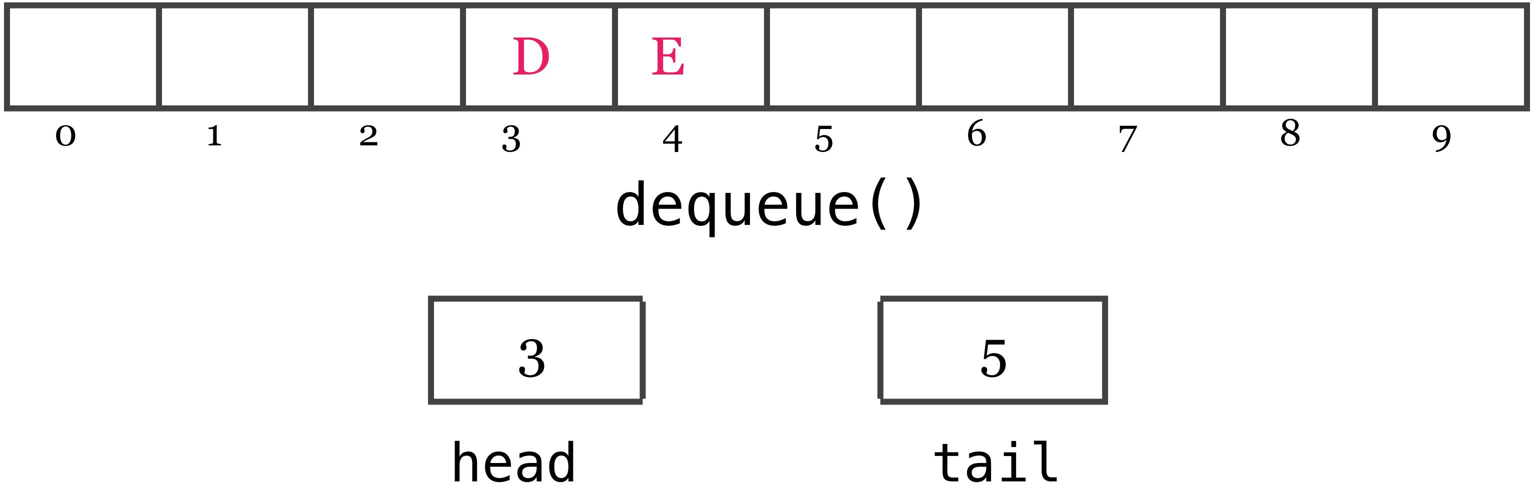
How dequeue( ) works



0	1	2	3	4	5	6	7	8	9
		C	D	E					

`dequeue()`

How dequeue( ) works

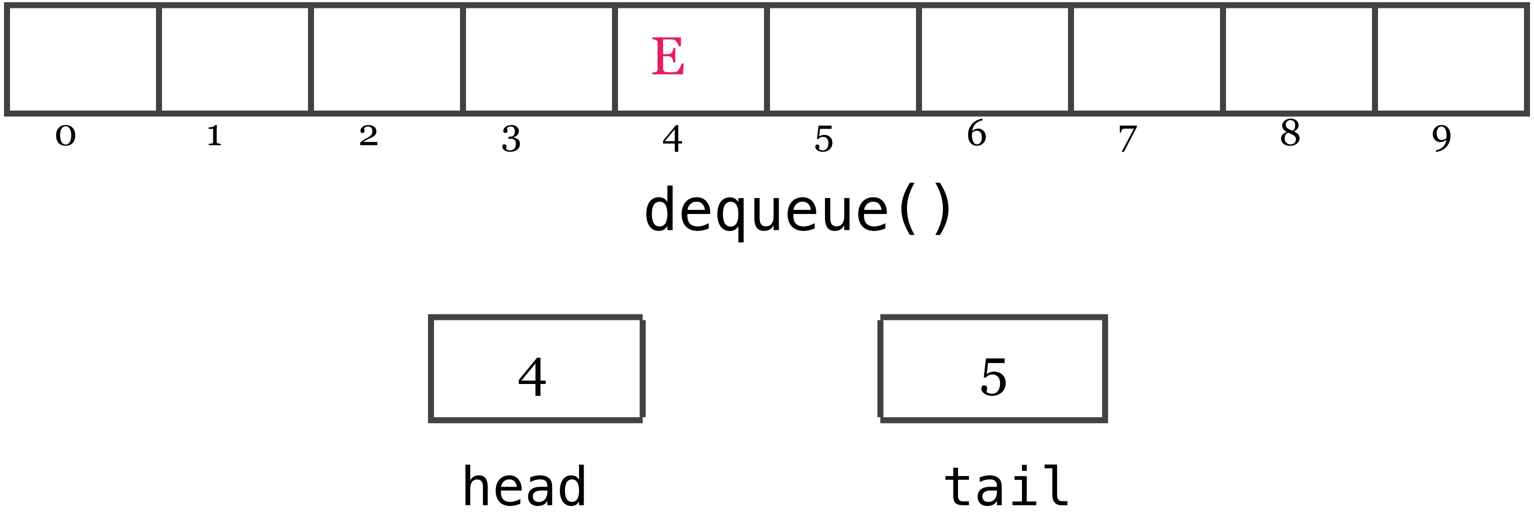


0	1	2	3	4	5	6	7	8	9
			D	E					

dequeue()

head	tail
3	5

How dequeue( ) works

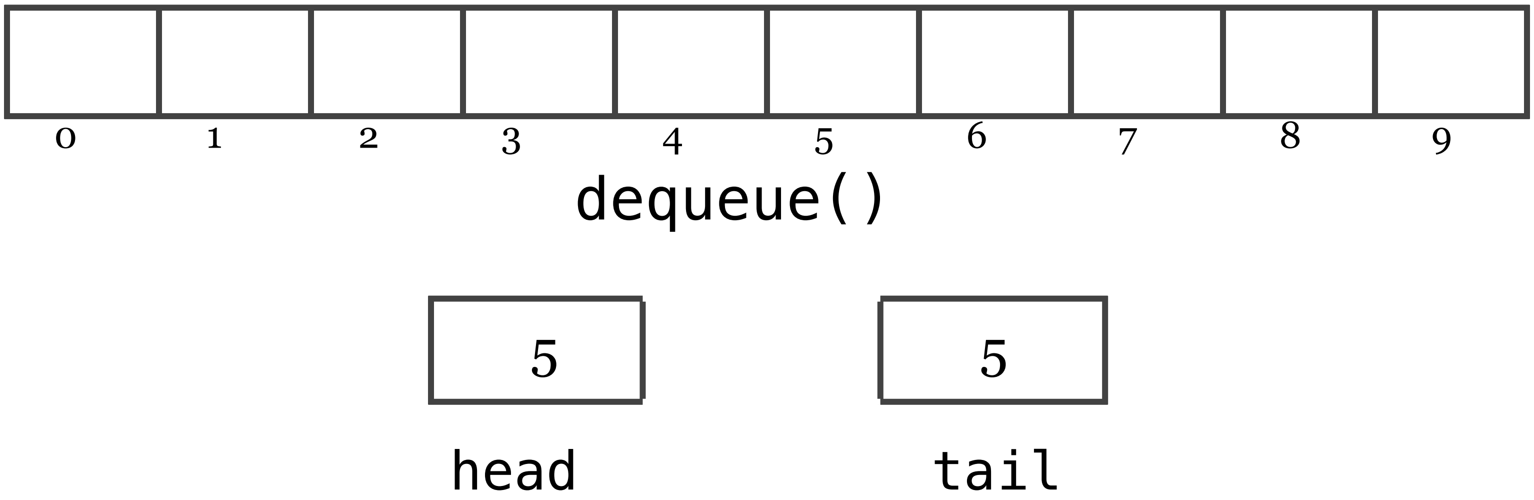


0	1	2	3	4	5	6	7	8	9
				E					

`dequeue()`

head	tail
4	5

How dequeue( ) works



0 1 2 3 4 5 6 7 8 9

dequeue()

5

head

5

tail

Limitations of array implementation of queue and stack

1. Space wastage in basic queue implementation.

2. Once maximum limit is reached stack and queue cannot add more nodes if it is necessary.

All of these problems can be resolved with little bit more logic. However, there are some better and elegant solution to this problem, and it involves linked list.