

# CS210

# Discussion

Week 4

# Attendance



# Today

- Project notes
- Buffer
- Josephus
- Finish through exercise 5
  - Show us passing Gradescope tests to leave early

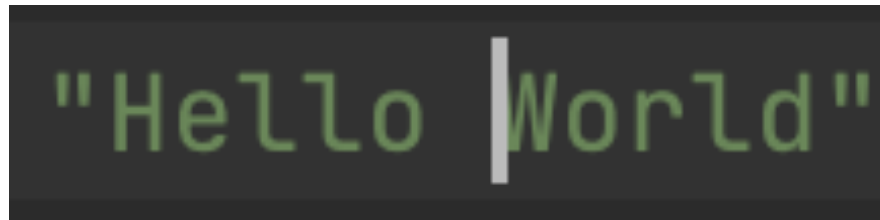
# Some Project Notes

- **Double check your performance requirements**
  - Count the runtime of called functions
- **Remove commented code**
- Use comments
- Be consistent with your style
  - Spacing
- Indent your code blocks
- Submit regrade requests

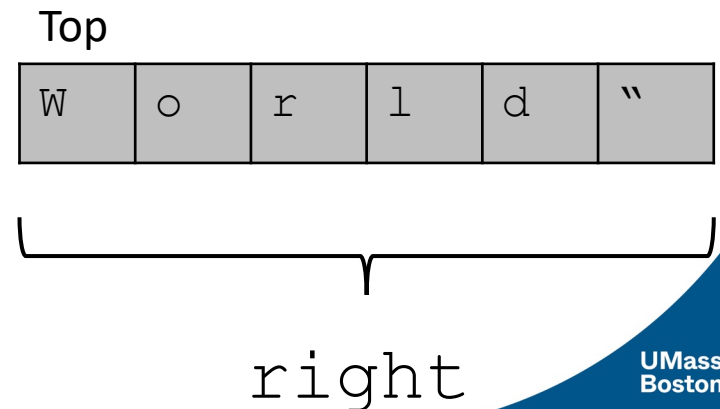
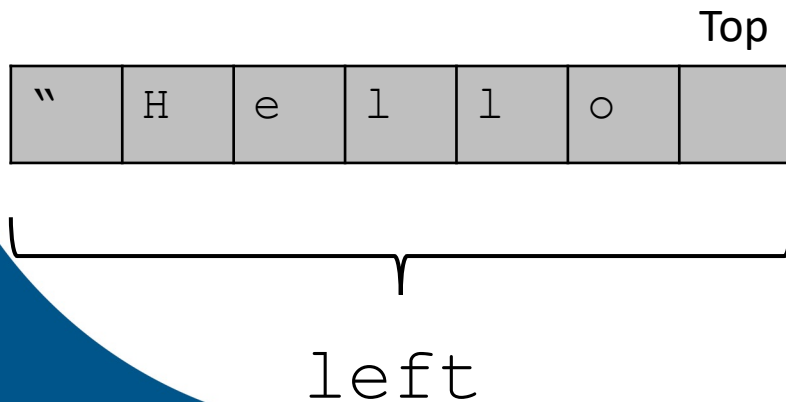
# Buffer

- Represent a text editor with two stacks
  - Implicit cursor (between the stacks)

```
protected LinkedStack<Character> left; // chars left of cursor  
protected LinkedStack<Character> right; // chars right of cursor
```



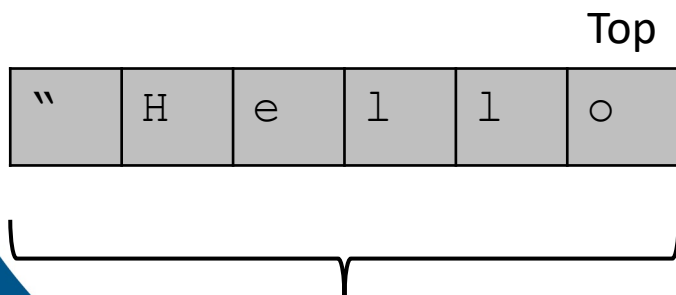
"Hello | World"



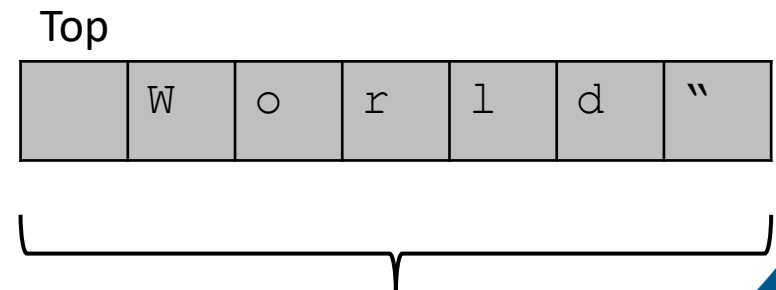
# Buffer

- Represent a text editor with two stacks
  - Implicit cursor (between the stacks)

```
protected LinkedStack<Character> left; // chars left of cursor  
protected LinkedStack<Character> right; // chars right of cursor
```



left



right

# Buffer

- Difference between pop and peek?
- Note that LinkedStacks are Iterable
  - For each loop can be used

```
for (Character c : left) {  
    StdOut.println(c);  
}
```

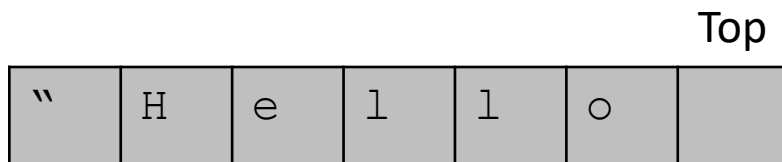
- Doesn't change the contents of the stack!!

```
dsa.Stack<Item> extends java.  
  
boolean isEmpty()  
  
int size()  
  
void push(Item item)  
  
Item peek()  
  
Item pop()  
  
Iterator<Item> iterator()
```

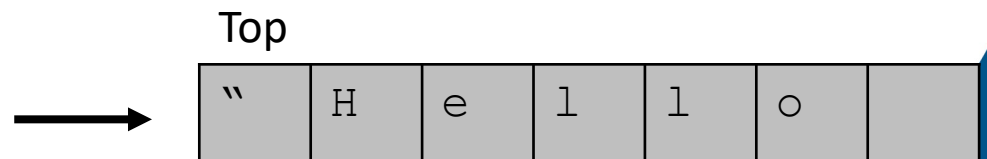
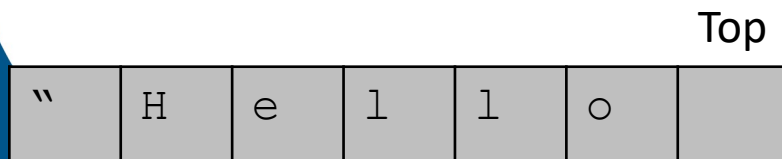
# Buffer

- `StringBuilder` is used to build strings
- `sb.append()`
  - Can take chars

```
public String toString() {  
    // A buffer to store the string representation.  
    StringBuilder sb = new StringBuilder();  
  
    // Push chars from left into a temporary stack.  
    ...  
}
```



→ olleH"





# Josephus

- Go through a circle of elements and keep removing the ' $m^{\text{th}}$ ' element
- Elements are numbers 1 to  $n$

```
>_ ~/workspace/exercise2
```

```
$ java Josephus 7 2
```

```
2
```

```
4
```

```
6
```

```
1
```

```
5
```

```
3
```

```
7
```

```
dsa.Queue<Item> extends java
```

```
boolean isEmpty()
```

```
int size()
```

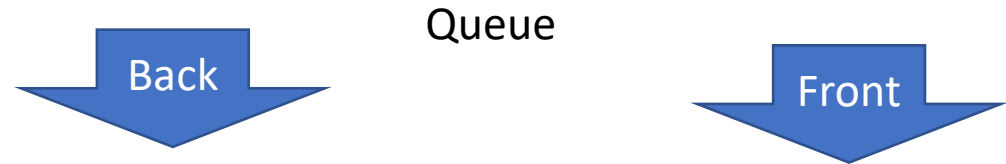
```
void enqueue(Item item)
```

```
Item peek()
```

```
Item dequeue()
```

```
Iterator<Item> iterator()
```

# Josephus



7	6	5	4	3	2	1
---	---	---	---	---	---	---

```
>_ ~/workspace/exercise2
```

```
$ java Josephus 7 2
```

```
2
```

```
4
```

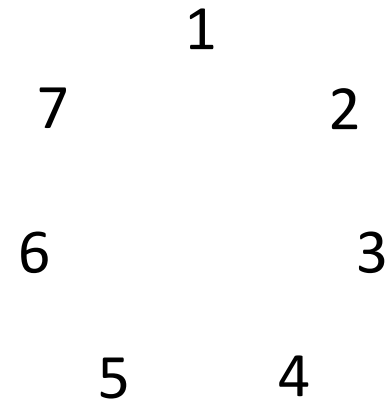
```
6
```

```
1
```

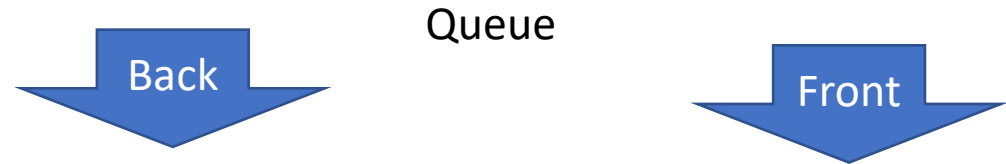
```
5
```

```
3
```

```
7
```



# Josephus



	1	7	6	5	4	3
--	---	---	---	---	---	---

```
>_ ~/workspace/exercise2
```

```
$ java Josephus 7 2
```

```
2
```

```
4
```

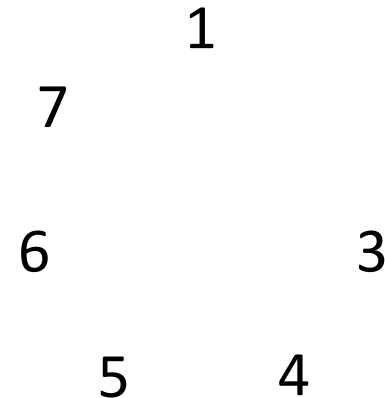
```
6
```

```
1
```

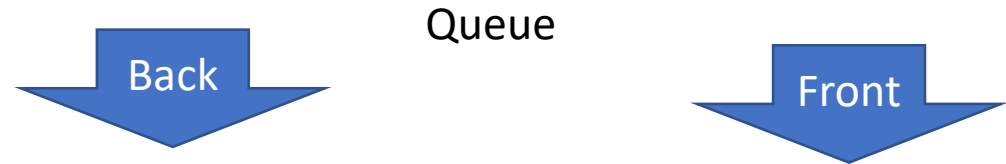
```
5
```

```
3
```

```
7
```



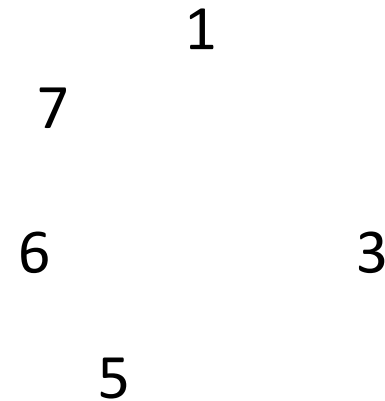
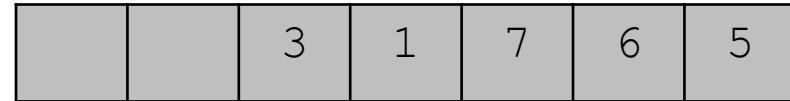
# Josephus



```
>_ ~/workspace/exercise2
```

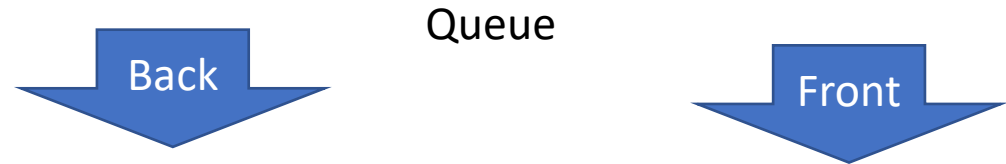
```
$ java Josephus 7 2
```

```
2
4
6
1
5
3
7
```



2 4

# Josephus



			5	3	1	7
--	--	--	---	---	---	---

```
>_ ~/workspace/exercise2
```

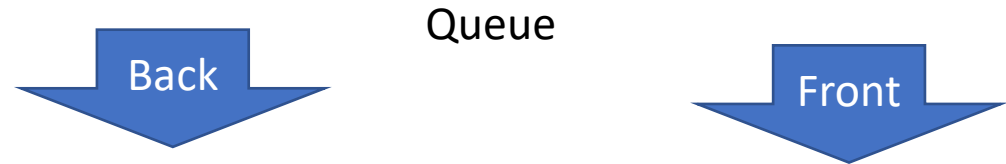
```
$ java Josephus 7 2
```

```
2  
4  
6  
1  
5  
3  
7
```

1  
7  
3  
5

2 4 6

# Josephus



```
>_ ~/workspace/exercise2
```

```
$ java Josephus 7 2
```

```
2  
4  
6  
1  
5  
3  
7
```



7

3

5

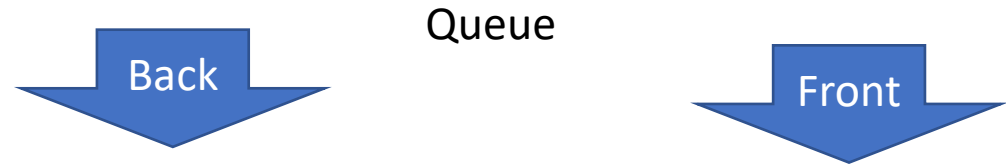
2

4

6

1

# Josephus



```
>_ ~/workspace/exercise2
```

```
$ java Josephus 7 2
```

```
2  
4  
6  
1  
5  
3  
7
```



7

3

2

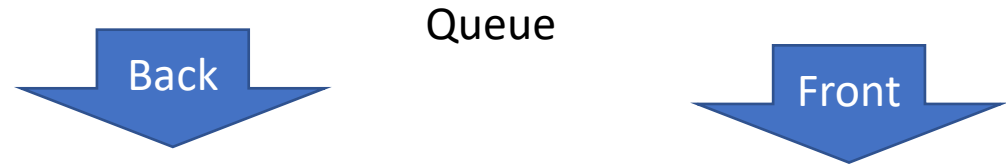
4

6

1

5

# Josephus



```
>_ ~/workspace/exercise2
```

```
$ java Josephus 7 2
```

```
2  
4  
6  
1  
5  
3  
7
```

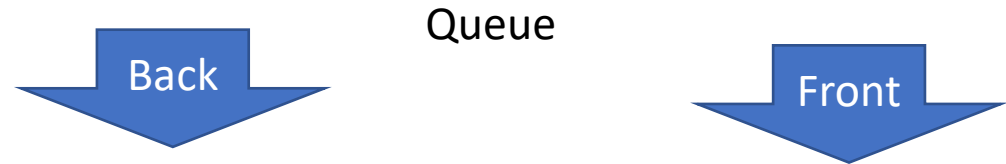


7

2 4 6 1 5 3



# Josephus



```
>_ ~/workspace/exercise2
```

```
$ java Josephus 7 2
```

```
2  
4  
6  
1  
5  
3  
7
```



2      4      6      1      5      3      7

# Questions?

