DSC 20
Discussion Section 10

ARDA CANKAT BATI

Today's Plan

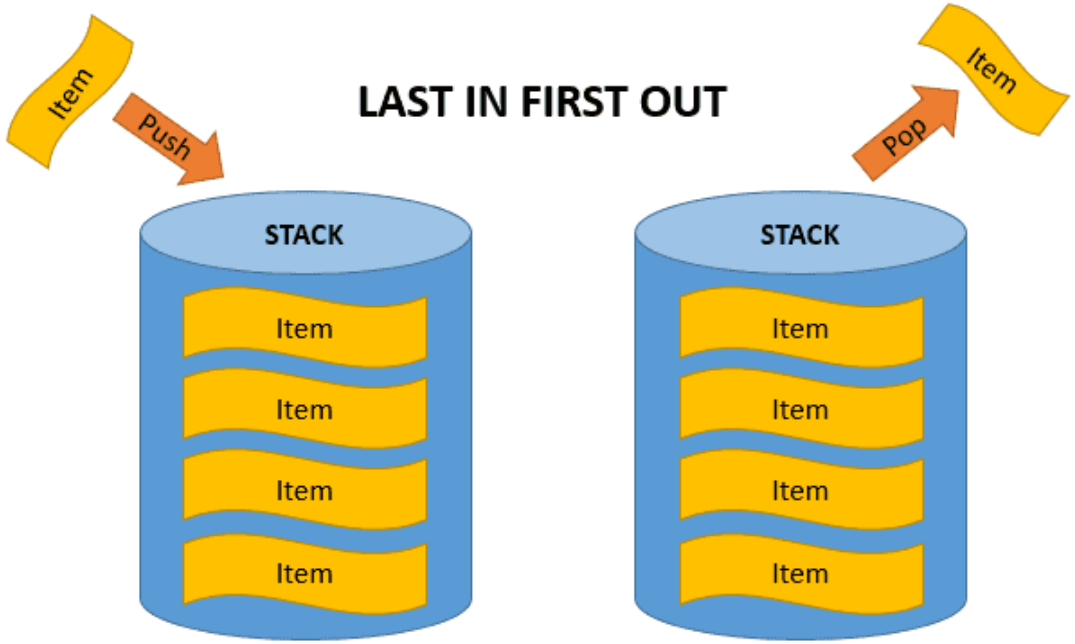
1. Topics Reviews:
   - Stacks & Queues
   - Circular arrays & queues
2. Resizing circular arrays
3. Talking about HW10 question

FIFO vs LIFO

Tunnel with both ends open

FIRST IN **FIRST OUT**

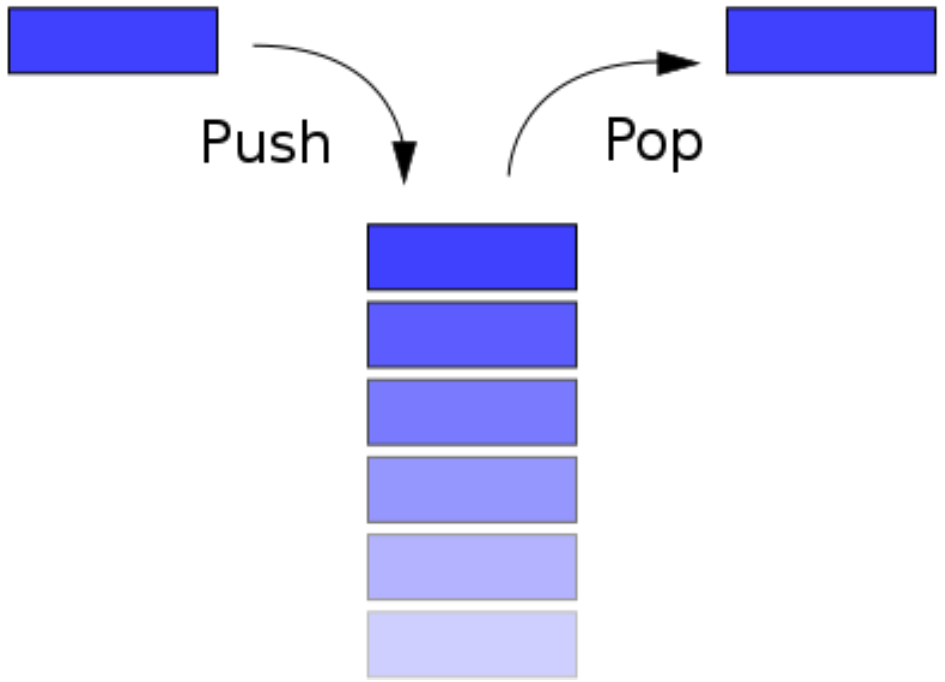
Stack of dishes, can only access top



Stacks (LIFO)

Stack: a data structure that allows adding & removing elements in a particular order. Every time an element is added, it goes on the **top** of the stack.

Only the top element can be removed from the stack, just like a pile of objects.



- **Pop:** Pop element at the top of the stack, removing it.
- **Push:** Pop element to the top of the stack.
- **Peek:** Get element at the top of the stack, **not** removing it.
- **Access:** Given an index, access the element in that index.
- **Search:** Search for a given element in the stack.

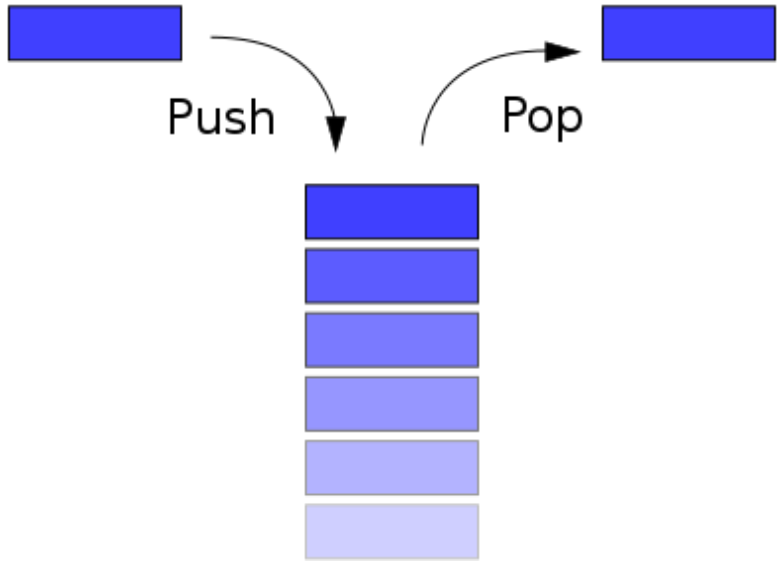
Further info: https://introcs.cs.princeton.edu/java/43stack/

Stacks (LIFO)

Operation	Worst Case Complexity
Push	?
Pop	?
Peek	?
Access element with known index	?
Search certain element	?

iClicker

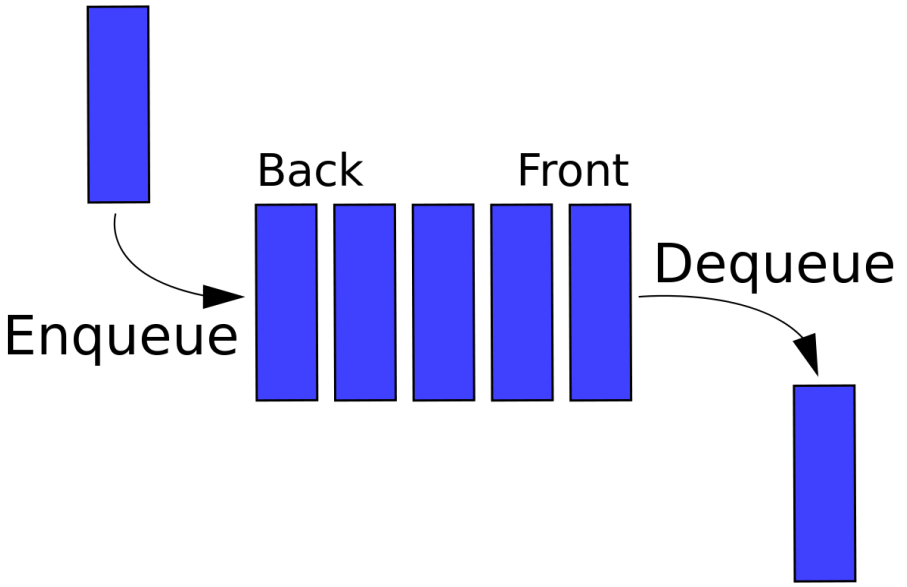
A) O(1)
B) O(n)
C) O(logn)
D) Something Else



Queues (FIFO)

Queue is used when things don't have to be processed immediately but processed in **F**irst **I**n **F**irst **O**ut order.

- **Enqueue:** Used to put elements at the back of the queue
- **Dequeue:** Used to get elements from front of the queue, removing them.
- **Peek:** Similar to peek() of stack, used to access the element at the front, without removing it from queue.
- **Access:** Given an index, access the element in
- **Search:** Search for a given element in the queue at index.



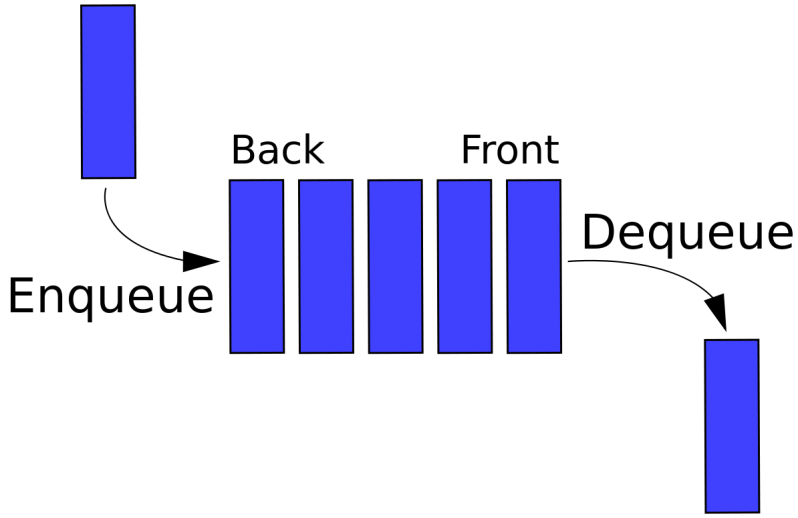
Further info: https://introcs.cs.princeton.edu/java/43stack/

Queues (FIFO)

Operation	Worst Case Complexity
Enqueue	?
Dequeue	?
Peek	?
Access element with known index	?
Search certain element	?

iClicker

A) O(1)
B) O(n)
C) O(logn)
D) Something Else



Example Problems Stack or Queue?

1) Imagine you are implementing an undo mechanism for a photo editing program. Which data structure you would use, stack or queue?

2) Consider we have a data transfer application (ie. Sending streaming packets). However, data is transferred asynchronously (data not necessarily received at same rate as sent) between the sender and receiver. Which data structure you would use, stack or queue?

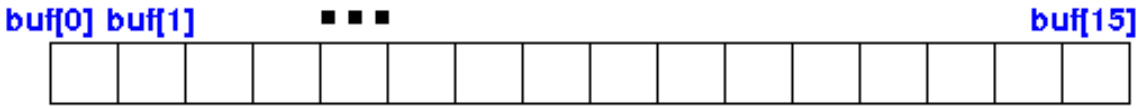
Example Problems Stack or Queue?

3) A resource is shared among multiple consumers. For example, consider we have a dataset. Some processes are adding more data to the dataset one by one. Also, some other processes are removing and processing data points one by one. We want to process every datapoint within a certain time frame. (An example is running Random Forests algorithm on big data, using many worker computers).

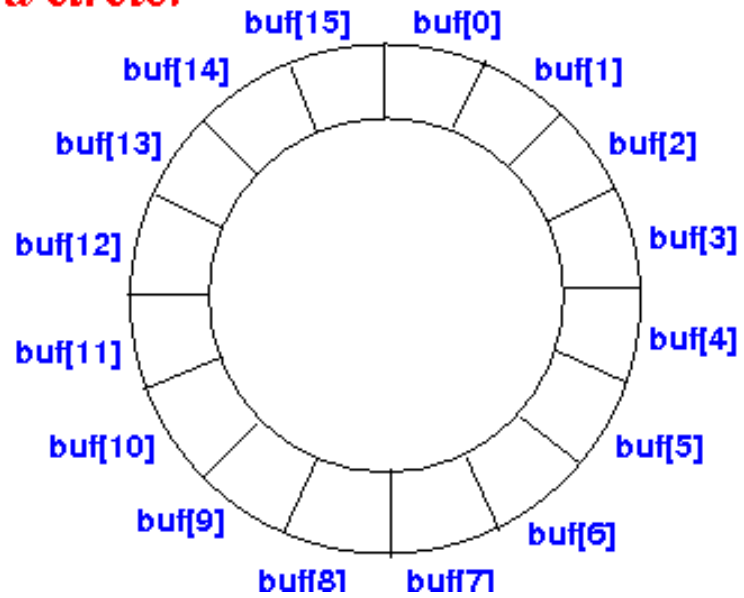
4) Consider you are given a single word in string form. You are asked to detect if this word is a palindrome (ie. "racecar"). We already did this with recursion. Which data structure you would us to do it without recursion?

Circular Arrays

Array:



Pretend array is a circle:

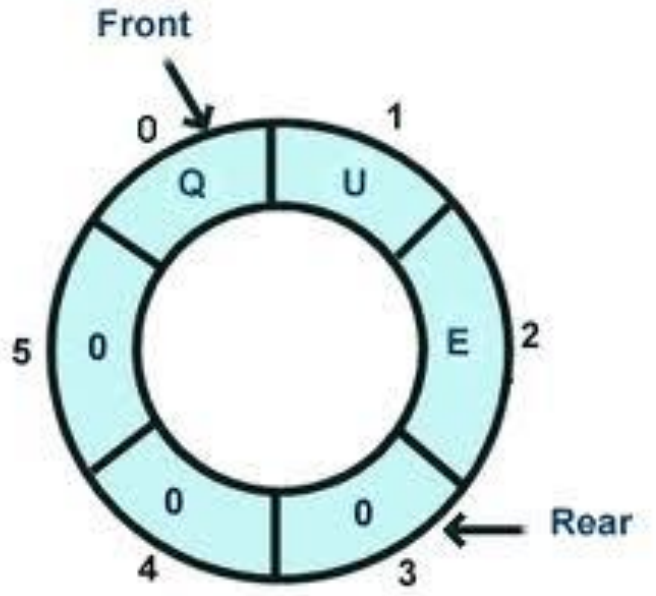


Circular Queues

We call the following:

```
queue1.enqueue("A")
queue1.enqueue("B")
queue1.dequeue()
```

What is the content of the queue, front and rear?



iClicker

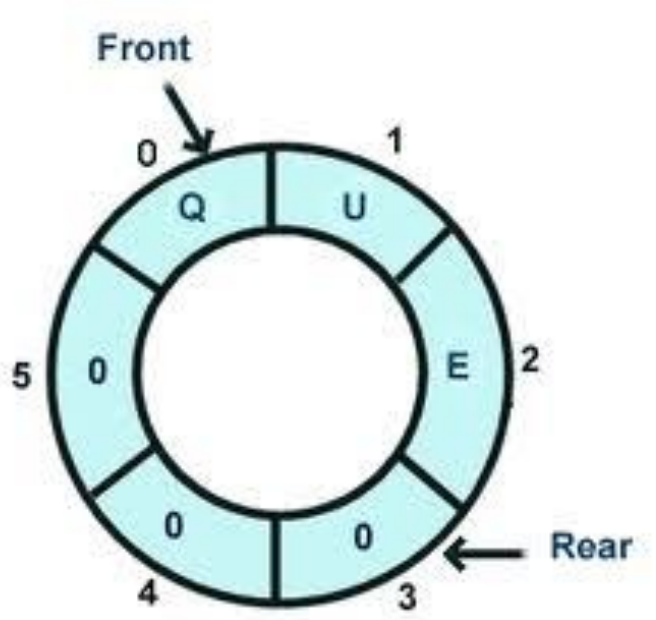
1. Queue: [E, A, B], front = 1, rear 4
2. Queue: [U, E, A, B], front = 1, rear 5
3. Queue: [U, E, A, B], front = 1, rear 4
4. Queue: [A, B], front = 1, rear 5

Circular Queues

Assume we don't do anything when front == rear. We call the following:

```
queue1.dequeue()
queue1.enqueue("A")
queue1.enqueue("B")
queue1.enqueue("C")
queue1.enqueue("D")
queue1.enqueue("E")
```

What are the final values of front and rear?



iClicker

1. front = 1, rear 0
2. front = 0, rear 1
3. front = 0, rear 3
4. front = 1, rear 2

Circular Queues

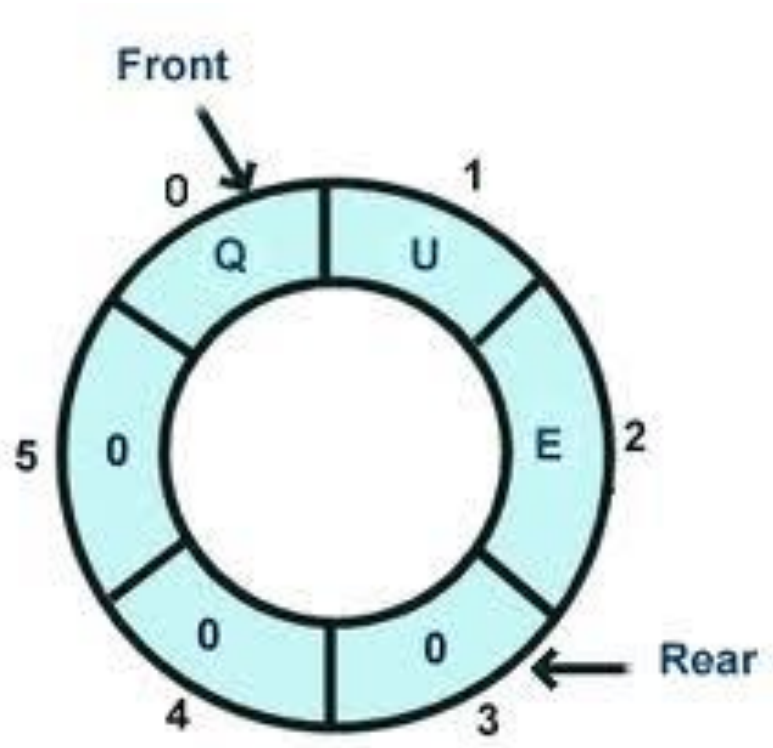
Assume we don't do anything when front == rear. We call the following:

```
queue1.enqueue("A")
queue1.enqueue("B")
queue1.enqueue("C")
queue1.enqueue("D")
queue1.enqueue("E")
```

Do we overwrite any elements without dequeue'ing them first (which means we lost the data by overwriting)

iClicker

A. No overwrites
B. Q overwritten
C. Q, U overwritten
D. Q, U, E all overwritten



HW10

Let's talk about homework 10 and resizing circular arrays.