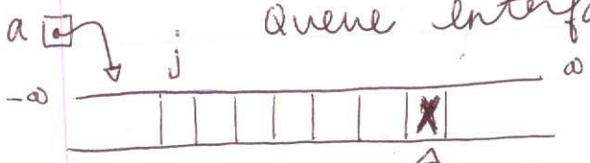
Read:
Section 2.3 & 2.4

Sept 22

Queue interface



n [7]
j [-2]

```
add(x)
  a[j+n] = x
  n++
```
} O(1)

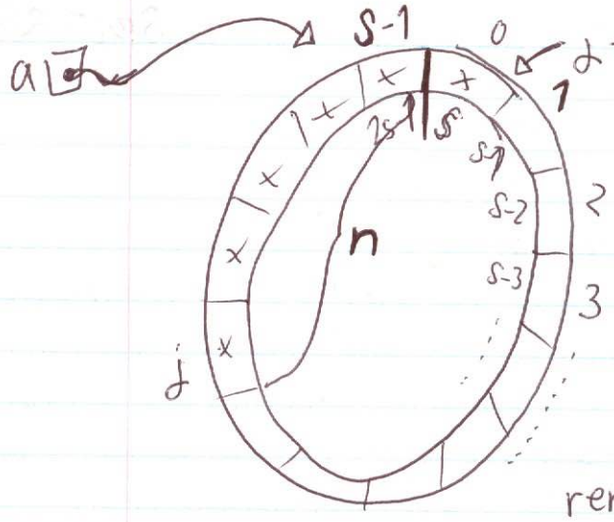
```
remove()
  Ty = a[j]
  a[j] = null
  j++
  n--
  return y
```
} O(1)

Deque interface

```
addFront(x)
  j--
  a[j] = x
  n++
```
} O(1)

O(1)

```
removeBack()
  Ty = a[j+n-1]
  a[j+n-1] = null
  n--
  return y
```



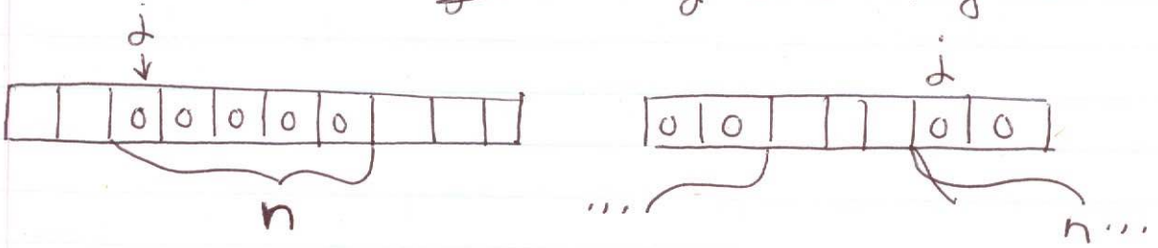
$j+n-1 \% a.length$

$2s-1 \% s = s-1$

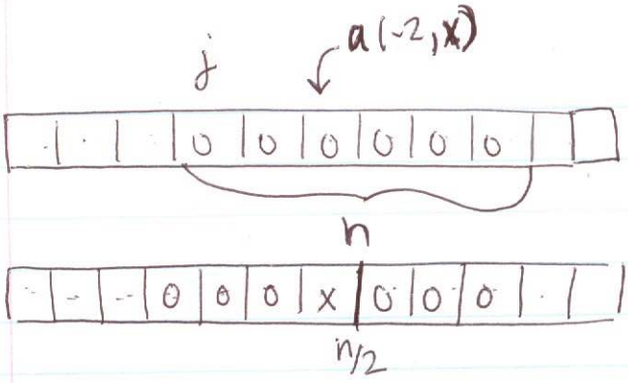
```
add(x)
  a[j+n % a.length] = x
  n ++
```
} O(1)

```
remove()
  Ty = a[j]
  a[j] = null
  (j+1) % a.length
  n--
  return y
```

```
resize()

T[] b = newArray(an)
  for (int i=0; i<n; i++)
      b[i] = ~~a[j+i]~~ a[j+i % a.length]
```

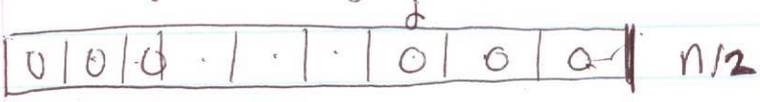


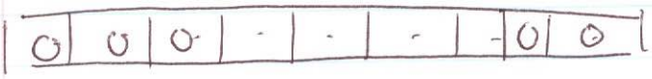

```
get(i)
  return a[(j+i) % a.length]

set(i,x)
  a[(j+i) % a.length] = x
```



List item

```
add(i,x)
  if (n == length) resize()
  if (i < n/2)
    shift j... i to the left      } O(i)
  else
    shift i... n-1 to the right   } O(n-i)
  n++                                          } O(1+min: {i,n-i})
  a[(j+i) % a.length] = x
```





```
remove(i)
  if (i < n/2)
     shift j... i right
     a[j] = null
     j++
  else
     shift i... n-1 left
     a[(j+i) % length] = null
  n--
  resize()?
```