

Stacks-Queues

Anh Huynh

1.

Stack:

Start

[_ _ _ _ _]

PUSH(S,4)

[4 _ _ _ _]

PUSH(S,1)

[4 1 _ _ _]

PUSH(S,3)

[4 1 3 _ _]

POP(S) → returns 3

[4 1 _ _ _]

PUSH(S,8)

[4 1 8 _ _]

POP(S) → returns 8

[4 1 _ _ _]

2.

Queue:

Start:

[_ _ _ _ _]

ENQUEUE(Q,4)

[4 _ _ _ _]

ENQUEUE(Q,1)

[4 1 _ _ _]

ENQUEUE(Q,3)

[4 1 3 _ _]

DEQUEUE(Q) → returns 4

[_ 1 3 _ _]

ENQUEUE(Q,8)

[_ 1 3 8 _]

DEQUEUE(Q) → returns 1

[_ _ 3 8 _]

3.

```
ENQUEUE with overflow detection
// check if the queue is already full:
// case 1: head is at 1 and tail is at the very end
```

```

// case 2: tail is just behind head (circular wrap)
if (Q.head == 1 and Q.tail == Q.length) or (Q.tail + 1 == Q.head)
    error "Queue overflow"    // can't add more stuff
else
    Q[Q.tail] = x             // put new element at the tail
    if Q.tail == Q.length     // if tail is at the very end
        Q.tail = 1           // wrap around back to the start
    else
        Q.tail = Q.tail + 1  // otherwise just move tail forward

DEQUEUE with underflow detection
// check if queue is empty: head == tail means no elements
if Q.head == Q.tail
    error "Queue underflow"  // nothing to remove
else
    x = Q[Q.head]            // grab the element at the head
    if Q.head == Q.length    // if head is at the very end
        Q.head = 1          // wrap around back to the start
    else
        Q.head = Q.head + 1  // otherwise just move head forward
    return x                  // return the removed element

```

4.

Deque $O(1)$ Operations

```

// Insertion at Front
if deque is full
    error "Deque Overflow"    // can't add, no room left
else
    if D.head == 1            // if head is already at the first slot
        D.head = D.length    // wrap it around to the last slot
    else
        D.head = D.head - 1   // otherwise, just move head back one slot
    D[D.head] = x             // place new element at the new head position

// Insertion at Rear
if deque is full
    error "Deque Overflow"    // can't add, no room left
else
    D[D.tail] = x             // put new element at tail
    if D.tail == D.length     // if tail is at the very end
        D.tail = 1           // wrap around to the beginning
    else
        D.tail = D.tail + 1   // otherwise, move tail forward

// Delete from Front
if deque is empty
    error "Deque underflow"   // nothing to remove

```

```

else
    x = D[D.head]                // grab the element at the head
    if D.head == D.length        // if head is at the very end
        D.head = 1                // wrap around back to the beginning
    else
        D.head = D.head + 1        // otherwise, move head forward
    return x                    // return the removed element

// Delete from Rear
if deque is empty
    error "Deque underflow"        // nothing to remove
else
    if D.tail == 1                // if tail is at the very first slot
        D.tail = D.length        // wrap it around to the last slot
    else
        D.tail = D.tail - 1        // otherwise, move tail backward
    x = D[D.tail]                // grab the element at the new tail
    return x                    // return the removed element

```

Video Link: <https://www.youtube.com/watch?v=Kpccwu4t8ow>

Error: I meant queueing and dequeuing a queue (0:22 timestamp)