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
if (Q.head == 1 and Q.tail == Q.length) or (Q.tail + 1 == Q.head)
    error "Queue overflow"
```

```
else
    Q[Q.tail] = x
    if Q.tail == Q.length
        Q.tail = 1
    else
        Q.tail = Q.tail + 1

DEQUEUE with underflow detection
if Q.head == Q.tail
    error "Queue underflow"
else
    x = Q[Q.head]
    if Q.head == Q.length
        Q.head = 1
    else
        Q.head = Q.head + 1
    return x
```

4.

```
Deque O(1) Operations

Insertion at Front
if deque is full
  error "Deque Overflow"
else
  if D.head == 1
    D.head = D.length
  else
    D.head = D.head - 1
  D[D.head] = x

Insertion at Rear
if deque is full
  error "Deque Overflow"
else
  D[D.tail] = x
  if D.tail == D.length
    D.tail = 1
  else
    D.tail = D.tail + 1

Delete from Front
if deque is empty
  error "Deque underflow"
else
  x = D[D.head]
  if D.head == D.length
    D.head = 1
  else
    D.head = D.head + 1
```

```
  return x

Delete from Rear
if deque is empty
  error "Deque underflow"
else
  if D.tail == 1
    D.tail = D.length
  else
    D.tail = D.tail - 1
  x = D[D.tail]
  return x
```