

12311410 Pg. 14 DSAA (H) Lab 08.

Q 8.1. 1. S: empty

⇒ 7

⇒ 7 4

⇒ 7 4 5

⇒ 7 4

⇒ 7 4 8

⇒ 7 4

⇒ 7

2. Q: empty

⇒ 7

⇒ 7 4

⇒ 7 4 5

⇒ 4 5

⇒ 4 5 8

⇒ 5 8

⇒ 8

3. 1: empty

⇒ 7

⇒ 4-7

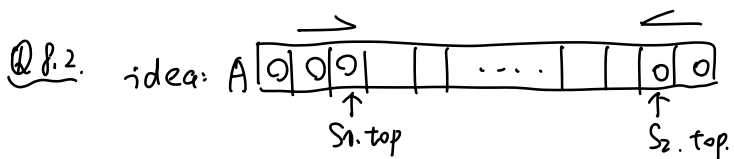
⇒ 5-4-7

⇒ 5-7

⇒ 8-5-7

⇒ 8-5

⇒ 5



Push  $S_1(A, x)$

1. if  $S_1.top == S_2.top - 1$
2. error "overflow".
3. else
4.  $S_1.top = S_1.top + 1$
5.  $A[S_1.top] = x.$

Push  $S_2(A, x)$

1. if  $S_2.top == S_1.top + 1$
2. error "overflow".
3. else
4.  $S_2.top = S_2.top + 1$
5.  $A[S_2.top] = x.$

Pop  $S_1(A)$

1. if  $S_1.top == 0$
2. error "underflow".
3. else
4.  $S_1.top = S_1.top - 1$
5. return  $A[S_1.top + 1].$

Pop  $S_2(A)$

1. if  $S_2.top == A.size + 1$
2. error "underflow".
3. else
4.  $S_2.top = S_2.top - 1$
5. return  $A[S_2.top + 1].$

Q8.3 Enqueue( $Q, x$ ).

1. if  $Q.tail + 1 == Q.head$
2. error "overflow".
3.  $Q[Q.tail] = x.$
4. if  $Q.tail == Q.size$
5.  $Q.tail = 1.$
6. else
7.  $Q.tail = Q.tail + 1.$

Dequeue( $Q$ ).

1. if  $Q.tail == Q.head$
2. error "underflow".
3. if  $Q.head == Q.size$
4.  $Q.head = 1$
5. else
6.  $Q.head = Q.head + 1.$

Q.8.V. idea: Construct a object "queue"  $Q$  which  
uses  $S_1$  to push and  $S_2$  to pop.

Enqueue ( $Q, x$ ).

1. Push ( $Q.S_1, x$ )

$\Rightarrow$  runtime: same as push  
 $\Rightarrow O(1)$ .

Dequeue ( $Q$ ).

1. if Is\_Empty ( $Q.S_2$ )

2. while not Is\_Empty ( $Q.S_1$ )

3. Push ( $Q.S_2, \text{Pop}(Q.S_1)$ ).

4. return Pop ( $Q.S_2$ )

$\Rightarrow$  runtime: worst: need to take the whole  $S_1$  to  
 $S_2$ .

$\Rightarrow O(n)$ .

amortized: cost per element: mainly  
 $\text{Pop}(Q.S_1), \text{Push}(Q.S_2), \text{Pop}(Q.S_2)$   
 $\Rightarrow$  all  $O(1) \Rightarrow O(1)$ .