

Week 4- Queue ADT, Circular Queue, List ADT

1. Suppose a circular queue of capacity $(n - 1)$ elements is implemented with an array of n elements. Assume that the insertion and deletion operation are carried out using REAR and FRONT as array index variables, respectively. Initially, $\text{REAR} = \text{FRONT} = 0$. The conditions to detect queue full and queue empty are
 - a) Full: $(\text{REAR}+1) \bmod n == \text{FRONT}$, empty: $\text{REAR} == \text{FRONT}$
 - b) Full: $(\text{REAR}+1) \bmod n == \text{FRONT}$, empty: $(\text{FRONT}+1) \bmod n == \text{REAR}$
 - c) Full: $\text{REAR} == \text{FRONT}$, empty: $(\text{REAR}+1) \bmod n == \text{FRONT}$
 - d) Full: $(\text{FRONT}+1) \bmod n == \text{REAR}$, empty: $\text{REAR} == \text{FRONT}$

Ans: a) Full: $(\text{REAR}+1) \bmod n == \text{FRONT}$, empty: $\text{REAR} == \text{FRONT}$

2. The following postfix expression with single digit operands is evaluated using a stack: $8\ 2\ 3\ ^\wedge\ /\ 2\ 3\ *\ +\ 5\ 1\ *\ -$. Note that $^\wedge$ is the exponentiation operator. The top two elements of the stack after the first $*$ is evaluated are:

Ans: 1 and 6

3. How many queues are needed to implement a stack. Consider the situation where no other data structure like arrays, linked list is available to you. Explain how?

Ans: We will need 2 queues to implement a stack.

If we want to PUSH (equivalent to PUSH operation) all the terms, we can enqueue all terms into QUEUE1. If we want to dequeue (equivalent to POP operation) an element, we can simultaneously dequeue the elements from QUEUE1 and enqueue them into QUEUE2. Now after enqueueing all terms in QUEUE2, they will be in reverse order and when we pop/dequeue from there we get the last-in element like how it should be done for a stack. and elements can be again sent back to QUEUE1.

4. How many stacks are needed to implement a queue. Consider the situation where no other data structure like arrays, linked list is available to you.

Ans: we need 2 stacks to implement a queue.

5. What operation is performed by the above function f ?

Suppose you are given an implementation of a queue of integers. The operations that can be performed on the queue are:

- i. isEmpty (Q) — returns true if the queue is empty, false otherwise.
- ii. delete (Q) — deletes the element at the front of the queue and returns its value.
- iii. insert (Q, i) — inserts the integer i at the rear of the queue.

Consider the following function:

```
void f (queue Q) {  
    int i ;  
    if (! isEmpty(Q)) {  
        i = delete(Q);  
        f(Q);  
        insert(Q, i);  
    }  
}
```

Ans:

When the queue is empty ,function f terminates.

When queue is not empty ,

the function f will delete all elements in the queue from front to rear and then adds them back in the same order. Finally the queue will be in reversed order.

6. What is the status of queue content after the following sequence of steps are performed on the circular queue. Show the content of the circular queue.

The initial configuration of circular queue is as follows:



Enqueue(a)

Dequeue()

Enqueue(b)

Dequeue()

Enqueue(c)

Dequeue()

Ans:

-	a	b	c	-
---	---	---	---	---