Queue

Queue

- Definition
  - FIFO (first-in first-out, 선입선출) 로 element 를 삽입/삭제하는 dynamic set

스택 큐

[그림 7-1] 스택과 큐의 구조 예

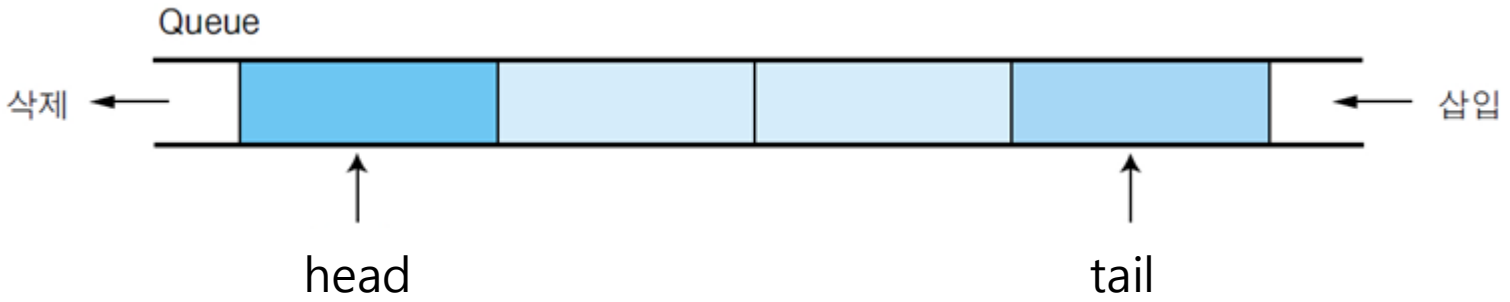
Queue

- Data Member
  - Q [1 … n] : array
  - head[*Q*] : 처음 원소의 index
  - tail[Q] : 새로이 삽입될 원소의 index

head

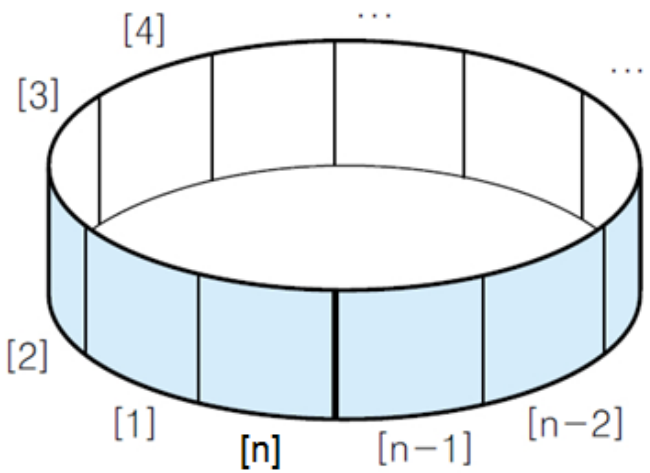
tail

큐에 저장된 원소는
tail 을 통하여 삽입,
head 를 통하여 삭제 가능



Circular Queue

- 원형 큐
  - 1차원 배열을 사용하면서 논리적으로 배열의 처음과 끝이 연결되어 있다고 가정하고 사용



initial state: head[Q] = tail[Q] = 1
queue empty: head[Q] = tail[Q]
queue full: head[Q] = (tail[Q]+1) % n

Q[1...n] 중 n-1 개의 원소만 사용

Circular Queue

- Operations
  - QUEUE-EMPTY(S)
    : queue 가 비어있는 지 여부 체크
  - QUEUE-FULL(S)
    : queue 가 꽉차있는 지 여부 체크
  - ENQUEUE(Q, x)
    : queue Q 에 원소 x 삽입
    : O(1)
  - DEQUEUE(S)
    : queue Q 에서 원소 삭제하고 값 리턴
    : O(1)

✓ Underflow: queue 가 이 empty 인데 DEQUEUE 수행
✓ Overflow: queue 가 full 인데 DEQUEUE 수행

QUEUE-EMPTY(Q)
```
if head[Q] = tail[Q]
    then return TRUE
    else  return FALSE
```

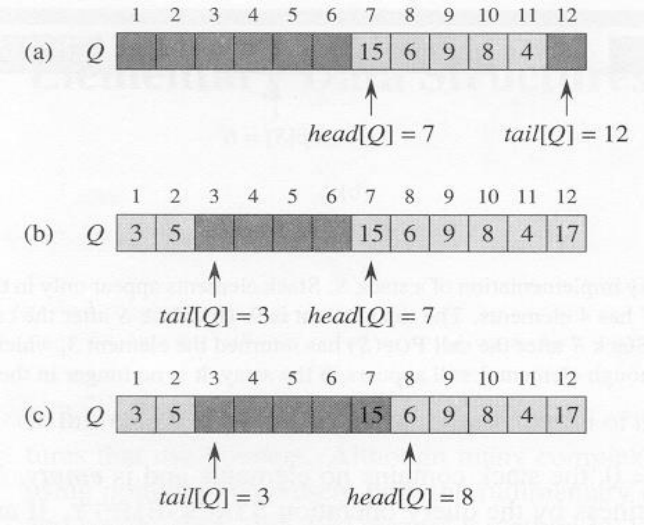
QUEUE-FULL(Q)
```
if head[Q] = (tail[Q]+1) % n
    then return TRUE
    else  return FALSE
```

ENQUEUE(Q, x)
```
if  QUEUE-FULL(Q)
     then error "overflow"
     else Q[tail[Q]] ← x
          tail[Q] = (tail[Q]+1)%n
```

DEQUEUE(Q)
```
if  QUEUE-EMPTY(QS)
     then error "underflow"
     else x ← Q[head[Q]]
          head[Q] = (head[Q]+1)%n
          return x
```

Circular Queue

- Operations



ENQUEUE(Q,17)
ENQUEUE(Q,3)
ENQUEUE(Q,5)

DEQUEUE(Q)

Circular Queue

(Q) Q[4],

ENQUEUE(Q,1), ENQUEUE(Q,2), DEQUEUE(Q), ENQUEUE(Q,3),
DEQUEUE(Q), ENQUEUE(Q,4), ENQUEUE(Q,5)

⇒ Q, head, tail = ?

Applications

- 운영체제의 작업 큐
  - 프린터 버퍼 큐
    - CPU에서 프린터로 보낸 데이터 순서대로(선입선출) 프린터에서 출력하기 위해서 선입선출 구조의 큐 사용
  - 키보드 버퍼 큐
  - 스케줄링 큐 / 이벤트 큐
    - CPU 사용을 요청한 프로세서들의 순서를 스케줄링하기 위해서 큐를 사용

- First In - First Service 응용 프로그램
  - 은행 창구
  - 선착순 대기열