

1 | 연결 큐의 이해와 구현

1 | 연결 큐의 이해와 구현

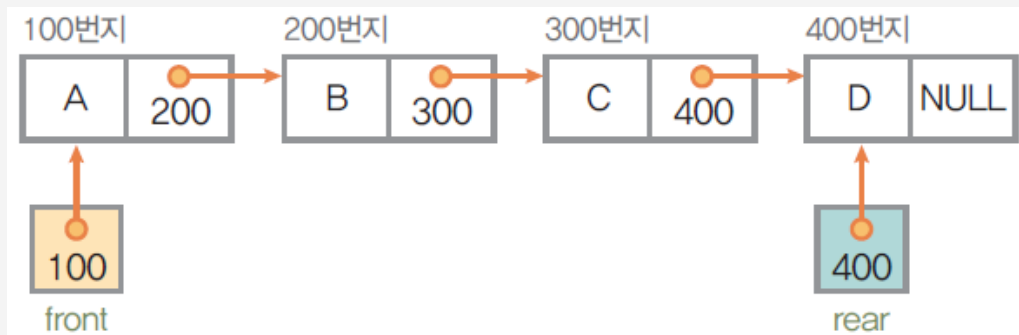
1 단순 연결 리스트를 이용한 큐

- ▶ 큐의 원소 : 단순 연결 리스트의 노드
- ▶ 큐의 원소의 순서 : 노드의 링크 포인터로 연결
- ▶ 변수 front : 첫 번째 노드를 가리키는 포인터 변수
- ▶ 변수 rear : 마지막 노드를 가리키는 포인터 변수
- ▶ 상태 표현
 - 초기 상태와 공백 상태 : front = rear = **null**

1 | 연결 큐의 이해와 구현

1 단순 연결 리스트를 이용한 큐

▶ 연결 큐의 구조



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

1 | 연결 큐의 이해와 구현

2 공백 연결 큐 생성 알고리즘

▶ 초기화 : front = rear = **null**

알고리즘 6-12 공백 연결 큐 생성

```
createLinkedList()  
    front ← NULL;  
    rear ← NULL;  
end createLinkedList()
```

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

1 | 연결 큐의 이해와 구현

3 연결 큐의 공백 상태 검사 알고리즘

▶ 공백 상태 : front = rear = **null**

알고리즘 6-13 연결 큐의 공백 상태 검사

```
isEmpty(LQ)
  if (front = NULL) then return true;
  else return false;
end isEmpty()
```

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 연결 큐의 삽입 알고리즘

알고리즘 6-14 연결 큐의 원소 삽입

```
enqueue(LQ, item)
{
  ① new ← getNode();
    new.data ← item;
    new.link ← NULL;

  ② if (front = NULL) then {
      rear ← new;
      front ← new;
    }

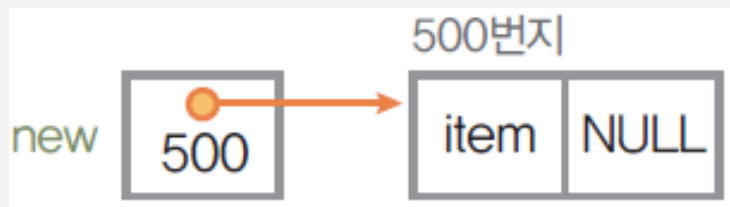
  ③ else {
      rear.link ← new;
      rear ← new;
    }
}
end enqueue()
```

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

1 | 연결 큐의 이해와 구현

4 연결 큐의 삽입 알고리즘

- ① 삽입할 새 노드를 생성하여 데이터 필드에 item을 저장
삽입할 새 노드는 연결 큐의 마지막 노드가 되어야
하므로 링크 필드에 NULL을 저장



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

1 | 연결 큐의 이해와 구현

4 연결 큐의 삽입 알고리즘

- ② 새 노드를 삽입하기 전에 연결 큐가 공백인지 아닌지를 검사
연결 큐가 공백인 경우에는 삽입할 새 노드가 큐의
첫 번째 노드이자 마지막 노드이므로 포인터 front와
rear가 모두 새 노드를 가리키도록 설정

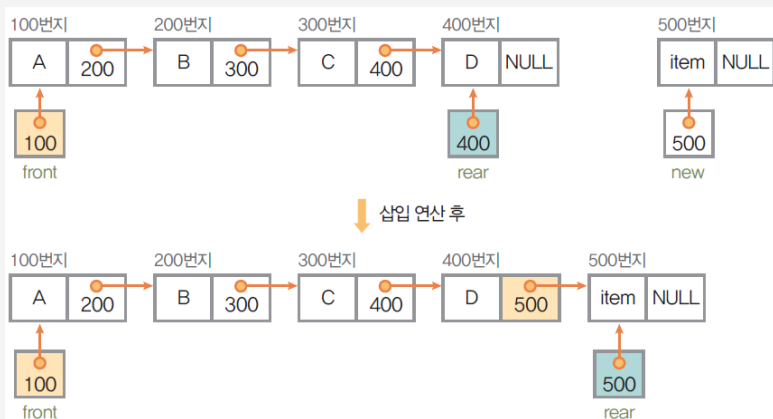


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

1 | 연결 큐의 이해와 구현

4 연결 큐의 삽입 알고리즘

- ③ 큐가 공백이 아닌 경우, 즉 노드가 있는 경우에는 현재 큐의 마지막 노드의 뒤에 새 노드를 삽입하고 마지막 노드를 가리키는 rear가 삽입한 새 노드를 가리키도록 설정



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

1 | 연결 큐의 이해와 구현

5 연결 큐의 원소 삭제 알고리즘

알고리즘 6-15 연결 큐의 원소 삭제

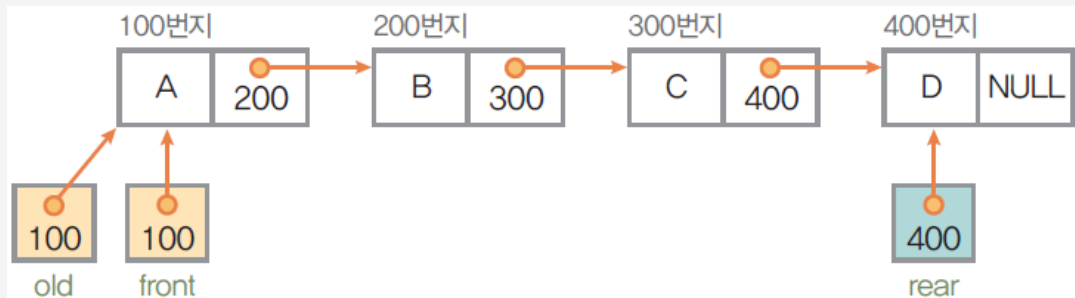
```
deQueue(LQ)
  if (isEmpty(LQ)) then Queue_Empty();
  else {
    ❶ old ← front;
      item ← front.data;
    ❷ front ← front.link;
    ❸ if (isEmpty(LQ)) then rear ← NULL;
    ❹ returnNode(old);
      return item;
  }
end deQueue()
```

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

1 | 연결 큐의 이해와 구현

5 연결 큐의 원소 삭제 알고리즘

- ① 삭제 연산에서 삭제할 노드는 큐의 첫 번째 노드로,
포인터 front가 가리키고 있는 노드
 - Front가 가리키는 노드를 포인터 old가 가리키게 하여
삭제할 노드로 지정

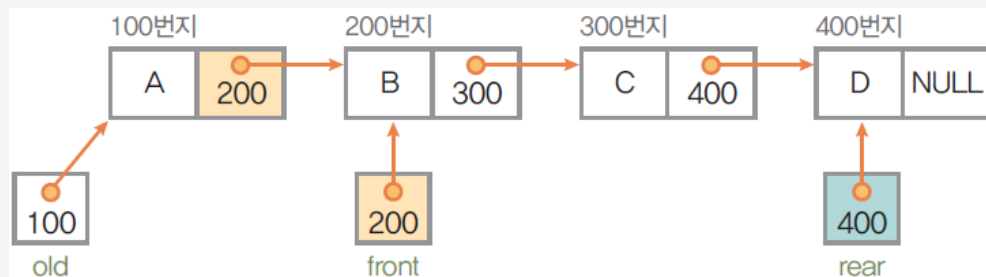


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

1 | 연결 큐의 이해와 구현

5 연결 큐의 원소 삭제 알고리즘

- ② 삭제 연산 후에는 현재 front 노드 다음 노드(front.link)가 front 노드가 되어야 하므로 포인터 front를 재설정

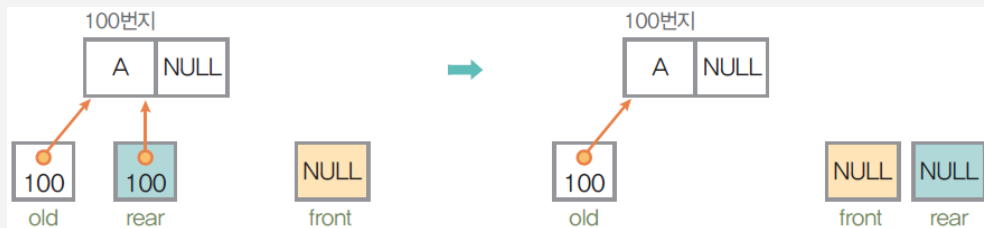


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

1 | 연결 큐의 이해와 구현

5 연결 큐의 원소 삭제 알고리즘

- ③ 현재 큐에 노드가 하나뿐이어서 재설정된 front가 NULL이 되는 경우에는 삭제 연산 후에 공백 큐가 되므로 포인터 rear를 NULL로 설정

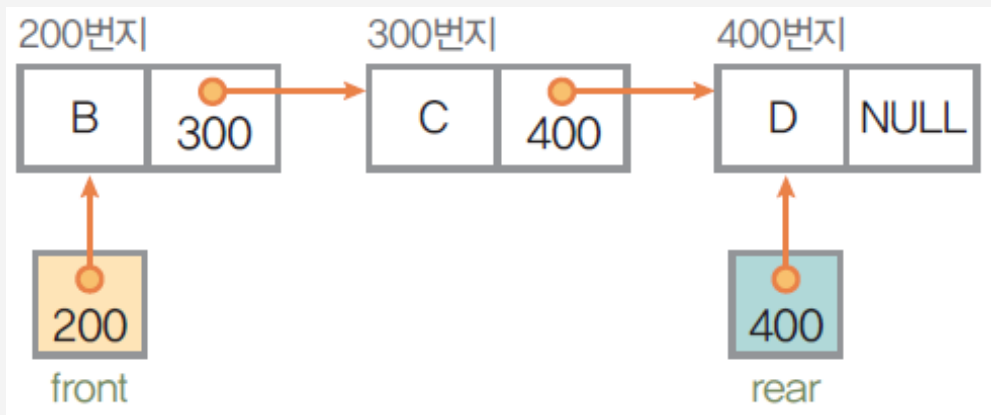


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

1 | 연결 큐의 이해와 구현

5 연결 큐의 원소 삭제 알고리즘

- ④ 포인터 old가 가리키고 있는 노드를 삭제하여 메모리 공간을 시스템에 반환(returnNode())



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

1 | 연결 큐의 이해와 구현

6 연결 큐의 원소 검색 알고리즘

▶ 연결 큐의 첫 번째 노드,
즉 front 노드의 데이터 필드 값을 반환

알고리즘 6-16 연결 큐의 원소 검색

```
peek(LQ)
  if (isEmpty(LQ)) then Queue_Empty()
  else return (front.data);
end peek()
```

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

1 | 연결 큐의 이해와 구현

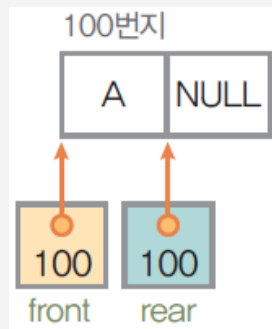
7 연결 큐에서의 연산 과정

▶ 연산 순서 : A삽입, B삽입, A삭제, C 삽입, D 삽입, E 삽입

① 공백 원형 큐 생성 : `createLinkedQueue();`



② 원소 A 삽입 : `enqueue(LQ, A);`

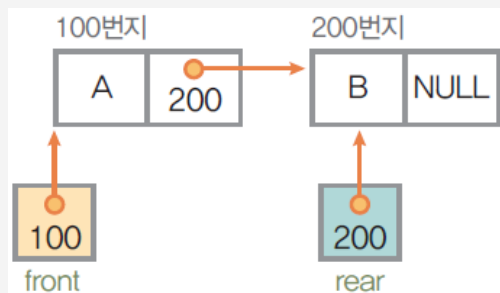


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

1 | 연결 큐의 이해와 구현

7 연결 큐에서의 연산 과정

③ 원소 B 삽입 : `enqueue(LQ, B);`

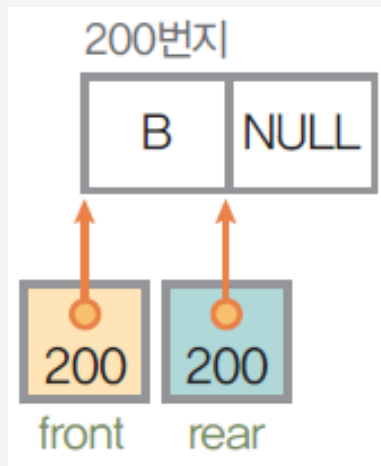


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

1 | 연결 큐의 이해와 구현

7 연결 큐에서의 연산 과정

④ 원소 삭제 : deQueue(LQ);

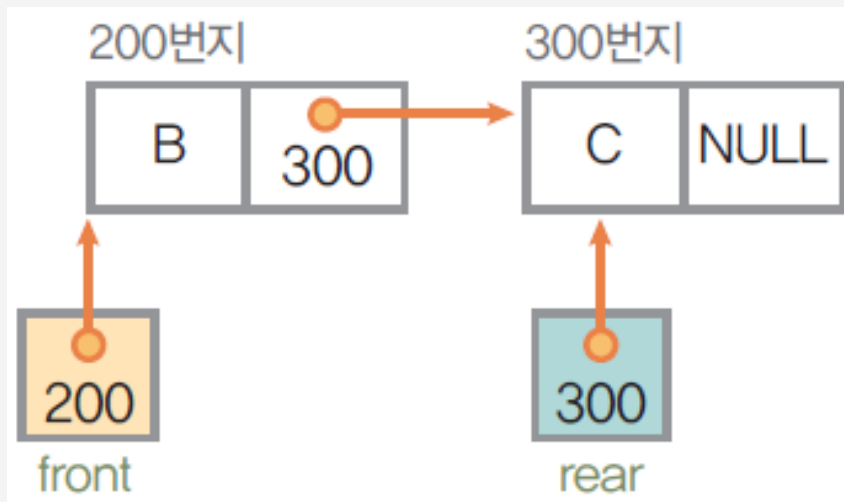


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

1 | 연결 큐의 이해와 구현

7 연결 큐에서의 연산 과정

⑤ 원소 C 삽입 : `enqueue(LQ, C);`

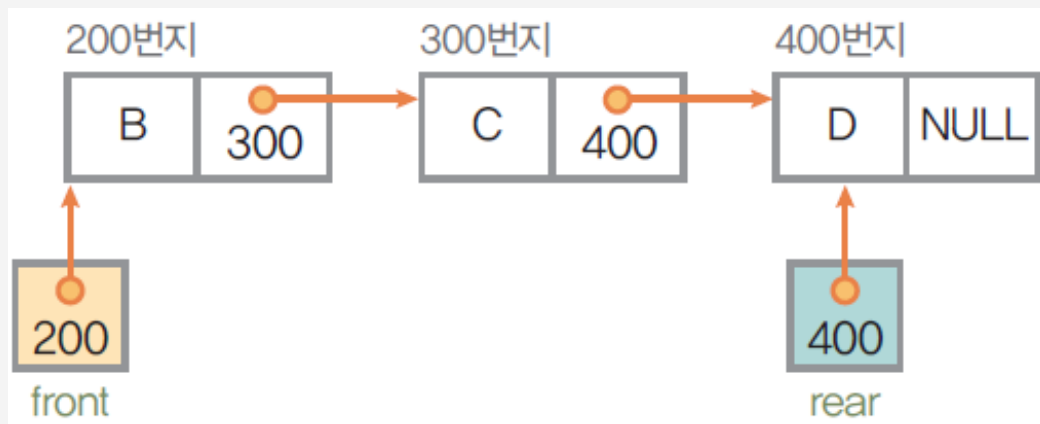


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

1 | 연결 큐의 이해와 구현

7 연결 큐에서의 연산 과정

⑥ 원소 D 삽입 : `enqueue(LQ, D);`

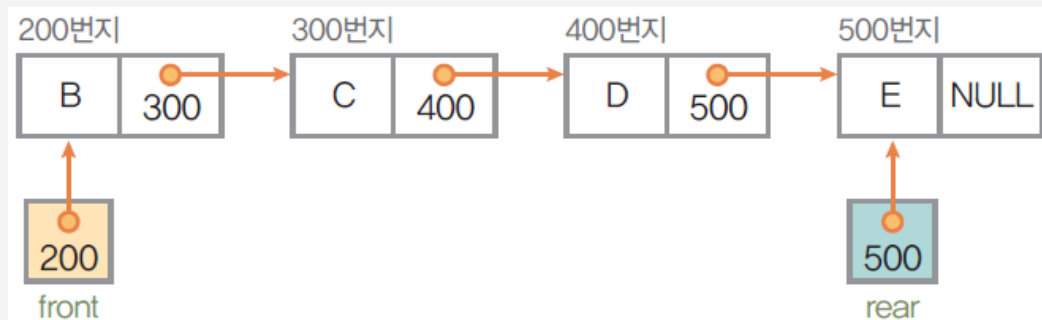


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

1 | 연결 큐의 이해와 구현

7 연결 큐에서의 연산 과정

⑦ 원소 E 삽입 : enqueue(LQ, E);

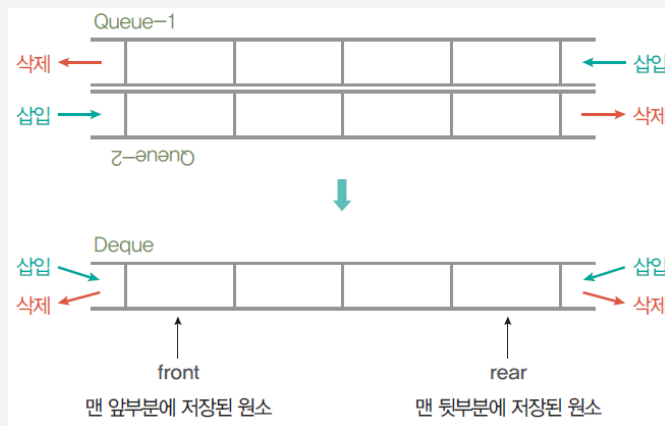


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

2 | 데크의 개념과 구현

1 데크(Deque : double-ended queue)

- ▶ 큐 두 개 중 하나를 좌우로 뒤집어서 붙인 구조,
큐의 양쪽 끝에서 삽입 연산과 삭제 연산을 수행할 수
있도록 확장한 자료구조



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

1 데크(Deque : double-ended queue)

ADT 6-2 데크의 추상 자료형

ADT deque

데이터 : 0개 이상의 원소를 가진 유한 순서 리스트

연산 :

$DQ \in \text{deque}; \text{item} \in \text{Element};$

// 공백 데크를 생성하는 연산

`createDeque() ::= create an empty DQ;`

// 데크가 공백 상태인지 검사하는 연산

`isEmpty(DQ) ::= if (DQ is empty) then return true
 else return false;`

// 데크의 front 앞에 item(원소)을 삽입하는 연산

`insertFront(DQ, item) ::= insert item at the front of DQ;`

// 데크의 rear 뒤에 item(원소)을 삽입하는 연산

`insertRear(DQ, item) ::= insert item at the rear of DQ;`

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

1 데크(Deque : double-ended queue)

```
// 데크의 front에 있는 item(원소)을 삭제하는 연산
deleteFront(DQ) ::= if (isEmpty(DQ)) then return NULL
                    else { delete and return the front item of DQ };

// 데크의 rear에 있는 item(원소)을 삭제하는 연산
deleteRear(DQ) ::= if (isEmpty(DQ)) then return NULL
                   else { delete and return the rear item of DQ };

// 데크의 front에 있는 item(원소)을 반환하는 연산
getFront(DQ) ::= if (isEmpty(DQ)) then return NULL
                 else { return the front item of the DQ };

// 데크의 rear에 있는 item(원소)을 반환하는 연산
getRear(DQ) ::= if (isEmpty(DQ)) then return NULL
                else { return the rear item of the DQ };
```

End deque

※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

2 | 데크의 개념과 구현

2 데크의 연산 과정

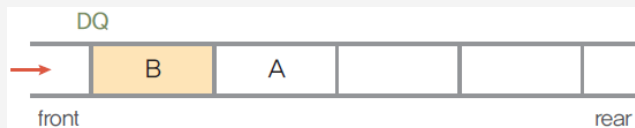
① createDeque();



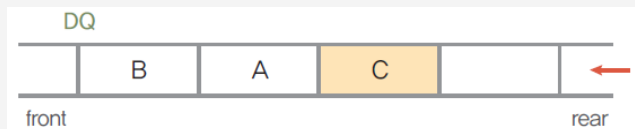
② insertFront(DQ, 'A');



③ insertFront(DQ, 'B');



④ insertRear(DQ, 'C');

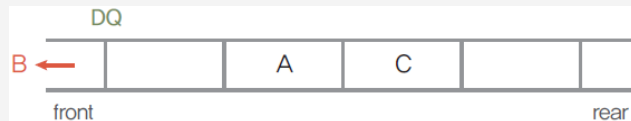


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

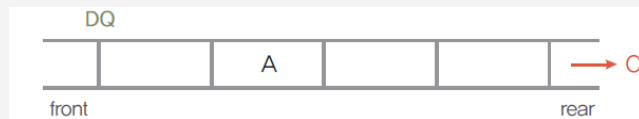
2 | 데크의 개념과 구현

2 데크의 연산 과정

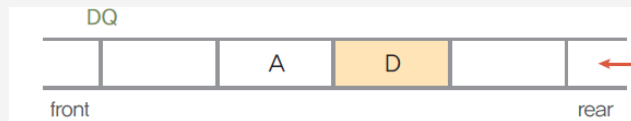
⑤ deleteFront(DQ);



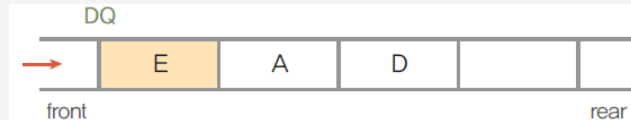
⑥ deleteRear(DQ);



⑦ insertRear(DQ, 'D');



⑧ insertFront(DQ, 'E');

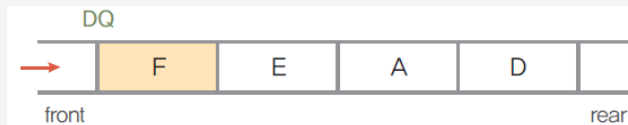


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

2 | 데크의 개념과 구현

2 데크의 연산 과정

⑨ insertFront(DQ, 'F');



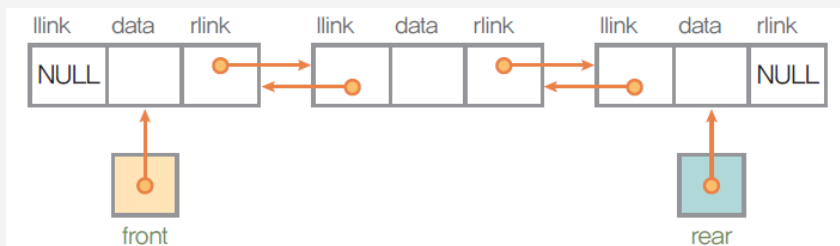
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

2 | 데크의 개념과 구현

3 데크의 구현

- ▶ 양쪽 끝에서 삽입/삭제 연산을 수행하면서 크기 변화와 저장된 원소의 순서 변화가 많으므로 순차 자료구조는 비효율적임
- ▶ 양방향으로 연산이 가능한 이중 연결 리스트를 사용

[데크의 이중 연결 리스트 구조]



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

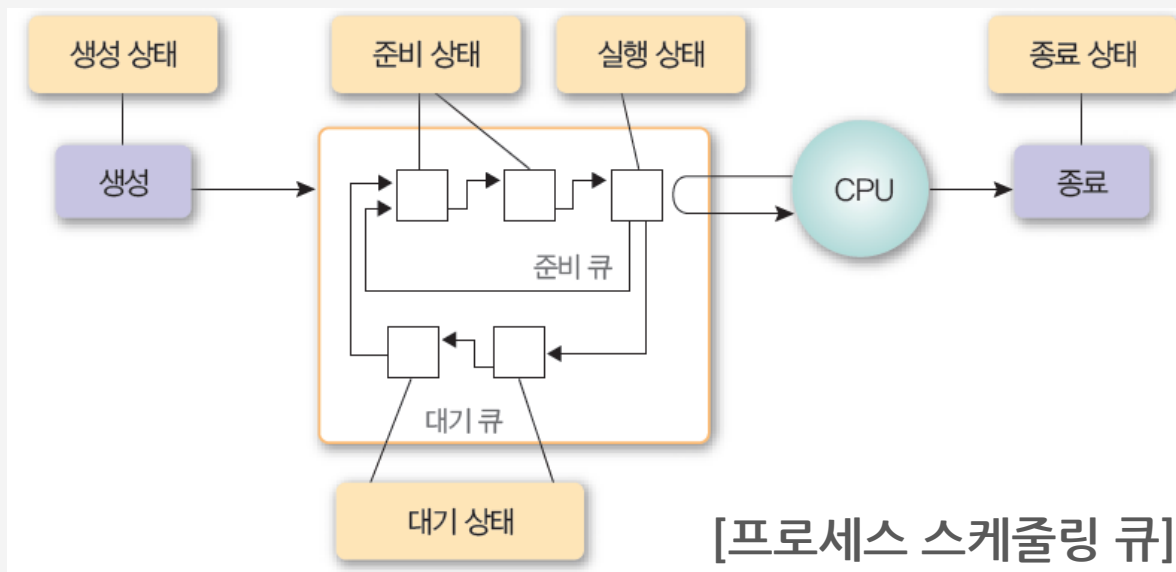
3 큐의 응용

1 운영체제의 작업 큐

- ▶ 프린터 버퍼 큐(Printer Buffer Queue)
 - CPU에서 프린터로 보낸 데이터 순서대로(선입선출) 프린터에서 출력하기 위해서 선입선출 구조의 큐 사용
- ▶ 스케줄링 큐(Scheduling Queue)
 - CPU 사용을 요청한 프로세서들의 순서를 스케줄링 하기 위해서 큐를 사용

3 | 큐의 응용

1 운영체제의 작업 큐



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어



2 시뮬레이션에서의 큐잉 시스템

- ▶ 시뮬레이션을 위한 수학적 모델링에서 대기행렬과 대기시간 등을 모델링 하기 위해서 큐의 이론(Queue theory) 사용