선수 지식 – 자료구조
덱

덱 | 다양한 알고리즘의 기본이 되는 자료구조 이해하기

강사 나동빈

선수 지식 - 자료구조

덱

덱(Deque)

- 덱은 **스택(stack)**과 **큐(queue)**의 기능을 모두 가지고 있다.
- 포인터 변수가 더 많이 필요하기 때문에, 메모리는 상대적으로 더 많이 필요하다.
- Python에서는 **큐(queue)의 기능**이 필요할 때 간단히 덱(deque)을 사용한다.
- 데이터의 삭제와 삽입 모두에서 $O(1)$의 시간 복잡도가 소요된다.

덱(Deque)

- 덱에 여러 개의 데이터를 삽입하고 삭제하는 예시를 확인해 보자.

[전체 연산]

- 좌측 삽입 4 – 좌측 삽입 3 – 좌측 삽입 2 – 좌측 삽입 1
- 우측 삽입 5 – 우측 삽입 6 – 우측 삽입 7 – 우측 삽입 8
- 우측 삭제 – 좌측 삭제 – 우측 삭제 – 좌측 삭제

덱(Deque)

- 덱에 여러 개의 데이터를 삽입하고 삭제하는 예시를 확인해 보자.

[전체 연산]

- 좌측 삽입 4 – 좌측 삽입 3 – 좌측 삽입 2 – 좌측 삽입 1
- 우측 삽입 5 – 우측 삽입 6 – 우측 삽입 7 – 우측 삽입 8
- 우측 삭제 – 좌측 삭제 – 우측 삭제 – 좌측 삭제

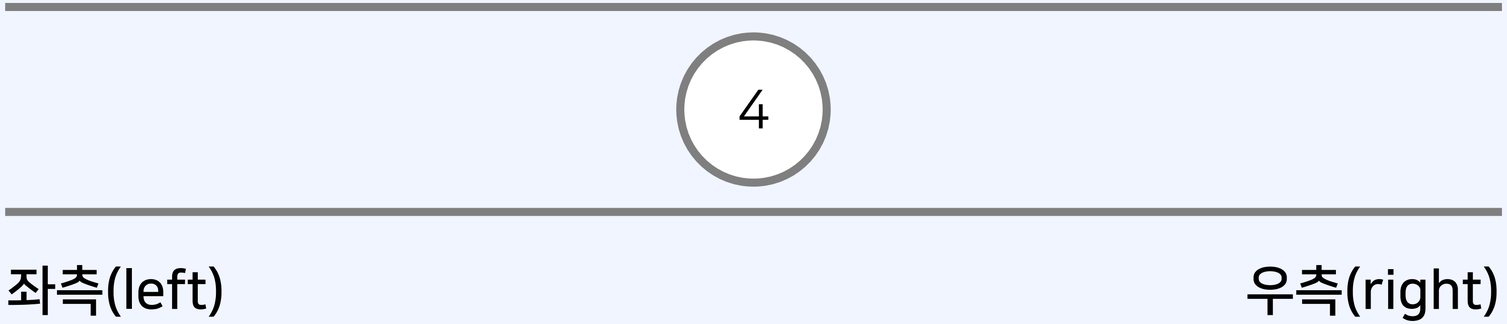
좌측(left) 우측(right)

덱(Deque)

- 덱에 여러 개의 데이터를 삽입하고 삭제하는 예시를 확인해 보자.

[전체 연산]

- 좌측 삽입 4 – 좌측 삽입 3 – 좌측 삽입 2 – 좌측 삽입 1
- 우측 삽입 5 – 우측 삽입 6 – 우측 삽입 7 – 우측 삽입 8
- 우측 삭제 – 좌측 삭제 – 우측 삭제 – 좌측 삭제

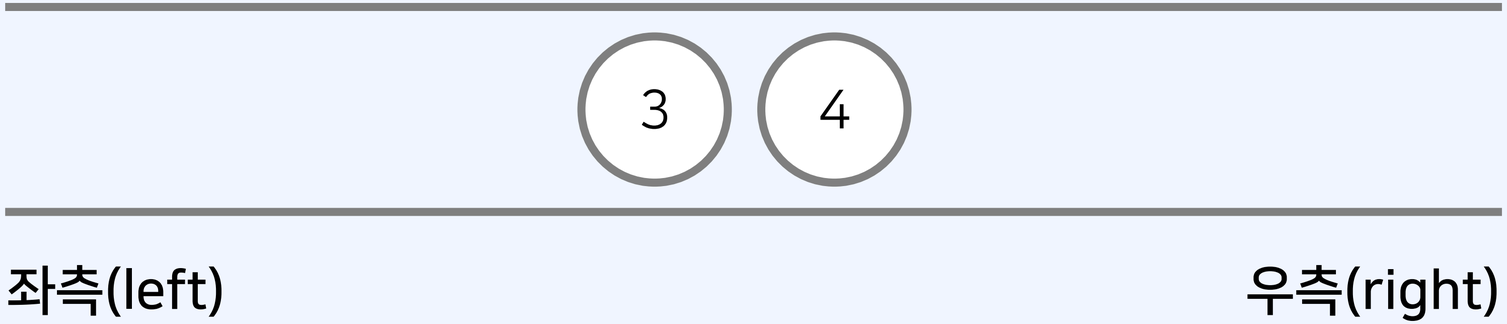


덱(Deque)

- 덱에 여러 개의 데이터를 삽입하고 삭제하는 예시를 확인해 보자.

[전체 연산]

- 좌측 삽입 4 – 좌측 삽입 3 – 좌측 삽입 2 – 좌측 삽입 1
- 우측 삽입 5 – 우측 삽입 6 – 우측 삽입 7 – 우측 삽입 8
- 우측 삭제 – 좌측 삭제 – 우측 삭제 – 좌측 삭제

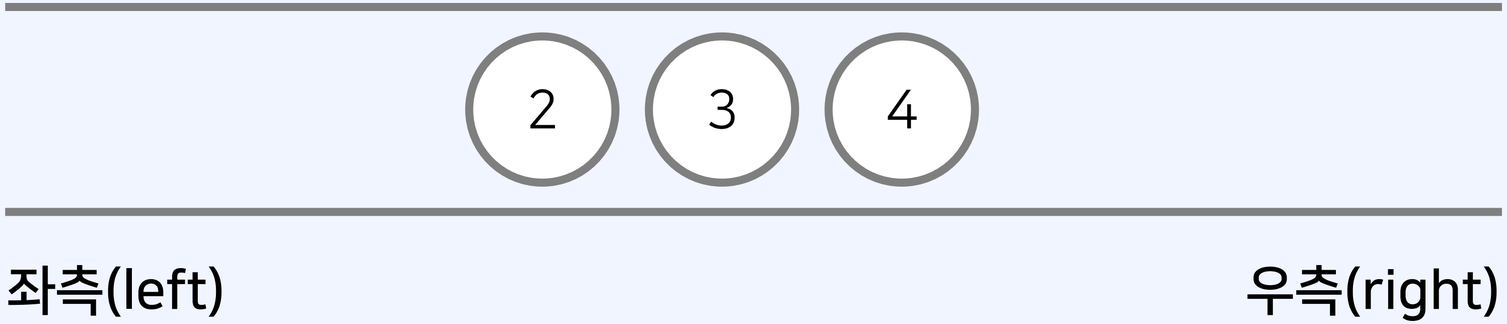


덱(Deque)

- 덱에 여러 개의 데이터를 삽입하고 삭제하는 예시를 확인해 보자.

[전체 연산]

- 좌측 삽입 4 – 좌측 삽입 3 – 좌측 삽입 2 – 좌측 삽입 1
- 우측 삽입 5 – 우측 삽입 6 – 우측 삽입 7 – 우측 삽입 8
- 우측 삭제 – 좌측 삭제 – 우측 삭제 – 좌측 삭제

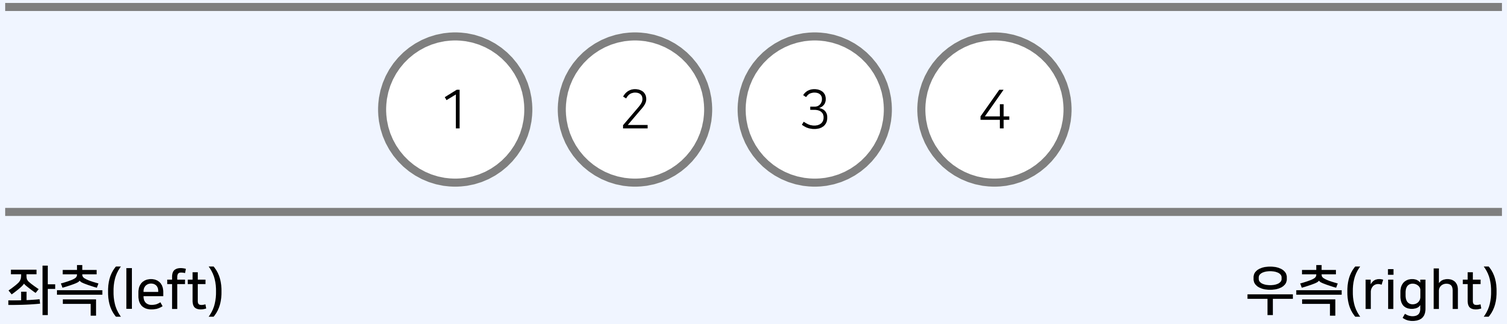


덱(Deque)

- 덱에 여러 개의 데이터를 삽입하고 삭제하는 예시를 확인해 보자.

[전체 연산]

- 좌측 삽입 4 – 좌측 삽입 3 – 좌측 삽입 2 – 좌측 삽입 1
- 우측 삽입 5 – 우측 삽입 6 – 우측 삽입 7 – 우측 삽입 8
- 우측 삭제 – 좌측 삭제 – 우측 삭제 – 좌측 삭제

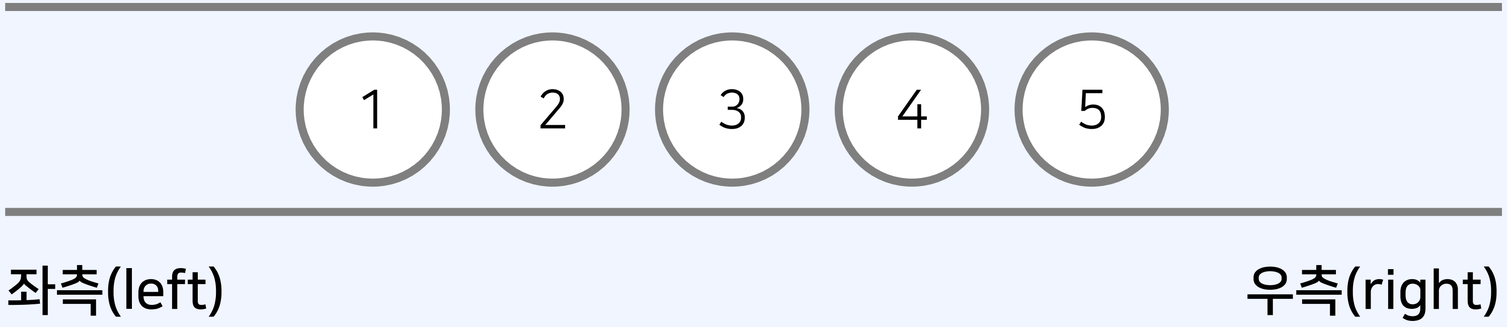


덱(Deque)

- 덱에 여러 개의 데이터를 삽입하고 삭제하는 예시를 확인해 보자.

[전체 연산]

- 좌측 삽입 4 – 좌측 삽입 3 – 좌측 삽입 2 – 좌측 삽입 1
- 우측 삽입 5 – 우측 삽입 6 – 우측 삽입 7 – 우측 삽입 8
- 우측 삭제 – 좌측 삭제 – 우측 삭제 – 좌측 삭제

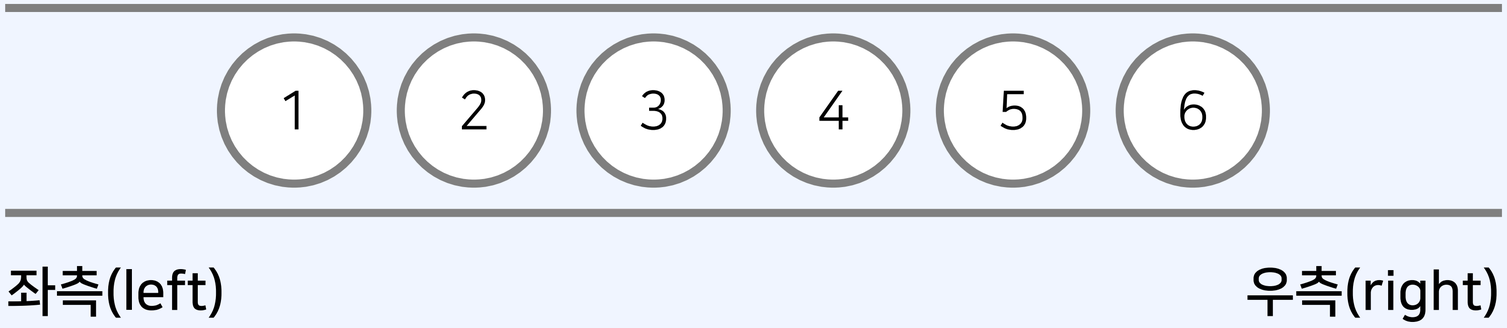


덱(Deque)

- 덱에 여러 개의 데이터를 삽입하고 삭제하는 예시를 확인해 보자.

[전체 연산]

- 좌측 삽입 4 – 좌측 삽입 3 – 좌측 삽입 2 – 좌측 삽입 1
- 우측 삽입 5 – 우측 삽입 6 – 우측 삽입 7 – 우측 삽입 8
- 우측 삭제 – 좌측 삭제 – 우측 삭제 – 좌측 삭제

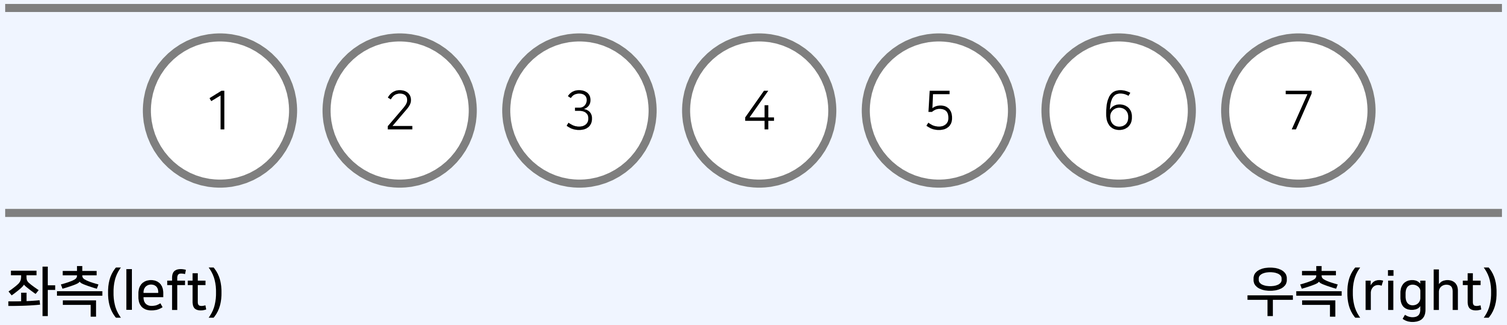


덱(Deque)

- 덱에 여러 개의 데이터를 삽입하고 삭제하는 예시를 확인해 보자.

[전체 연산]

- 좌측 삽입 4 – 좌측 삽입 3 – 좌측 삽입 2 – 좌측 삽입 1
- 우측 삽입 5 – 우측 삽입 6 – 우측 삽입 7 – 우측 삽입 8
- 우측 삭제 – 좌측 삭제 – 우측 삭제 – 좌측 삭제

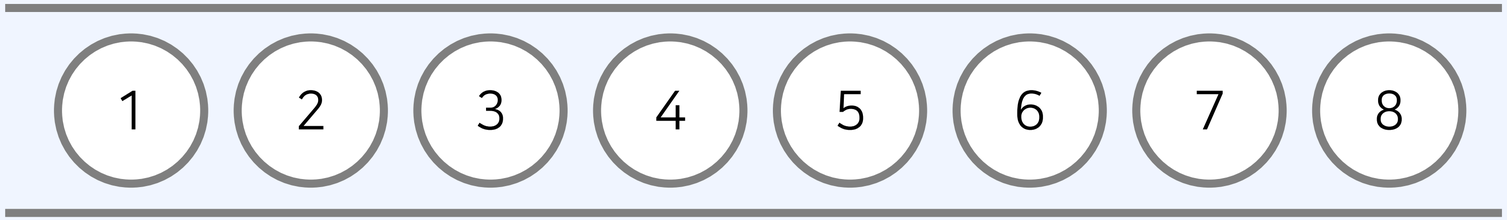


덱(Deque)

- 덱에 여러 개의 데이터를 삽입하고 삭제하는 예시를 확인해 보자.

[전체 연산]

- 좌측 삽입 4 – 좌측 삽입 3 – 좌측 삽입 2 – 좌측 삽입 1
- 우측 삽입 5 – 우측 삽입 6 – 우측 삽입 7 – 우측 삽입 8
- 우측 삭제 – 좌측 삭제 – 우측 삭제 – 좌측 삭제



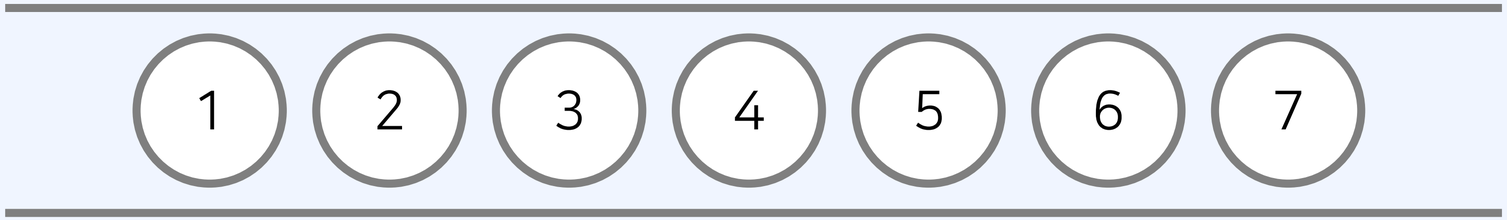
좌측(left) 우측(right)

덱(Deque)

- 덱에 여러 개의 데이터를 삽입하고 삭제하는 예시를 확인해 보자.

[전체 연산]

- 좌측 삽입 4 – 좌측 삽입 3 – 좌측 삽입 2 – 좌측 삽입 1
- 우측 삽입 5 – 우측 삽입 6 – 우측 삽입 7 – 우측 삽입 8
- 우측 삭제 – 좌측 삭제 – 우측 삭제 – 좌측 삭제



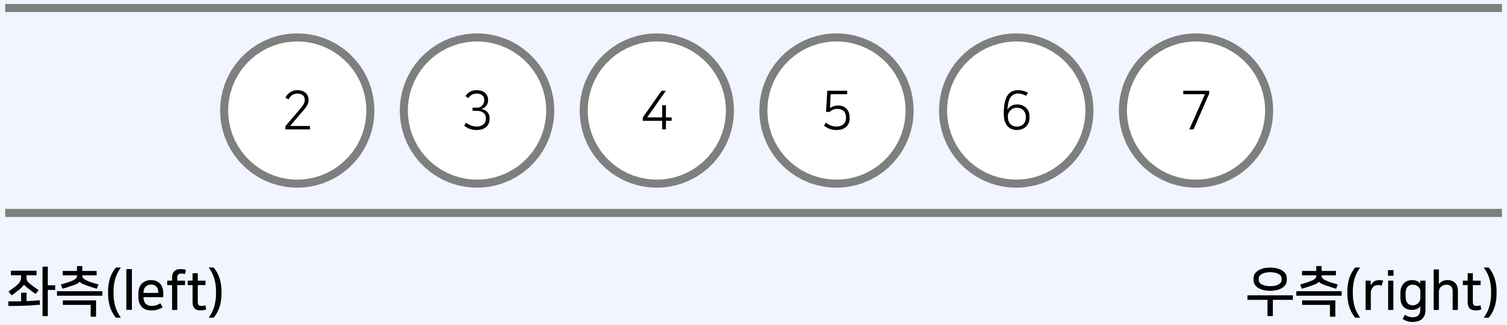
좌측(left) 우측(right)

덱(Deque)

- 덱에 여러 개의 데이터를 삽입하고 삭제하는 예시를 확인해 보자.

[전체 연산]

- 좌측 삽입 4 – 좌측 삽입 3 – 좌측 삽입 2 – 좌측 삽입 1
- 우측 삽입 5 – 우측 삽입 6 – 우측 삽입 7 – 우측 삽입 8
- 우측 삭제 – 좌측 삭제 – 우측 삭제 – 좌측 삭제

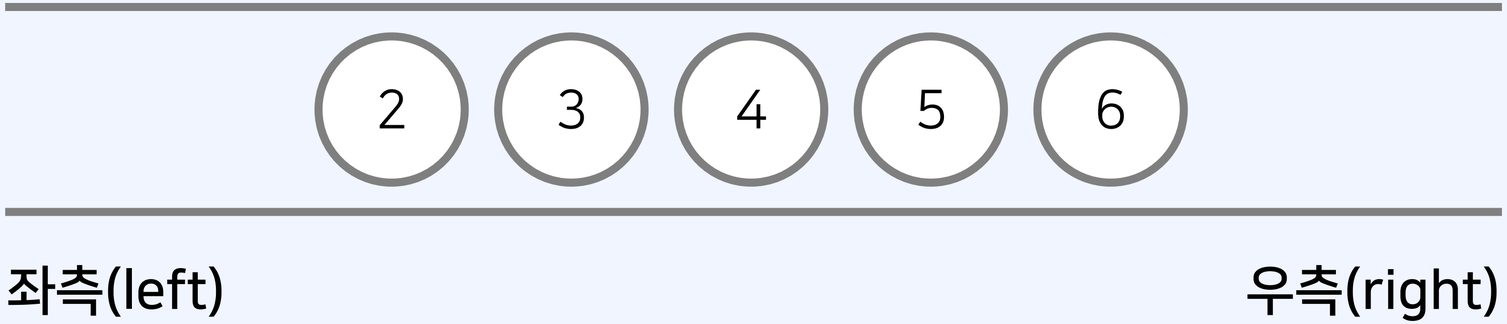


덱(Deque)

- 덱에 여러 개의 데이터를 삽입하고 삭제하는 예시를 확인해 보자.

[전체 연산]

- 좌측 삽입 4 – 좌측 삽입 3 – 좌측 삽입 2 – 좌측 삽입 1
- 우측 삽입 5 – 우측 삽입 6 – 우측 삽입 7 – 우측 삽입 8
- 우측 삭제 – 좌측 삭제 – 우측 삭제 – 좌측 삭제

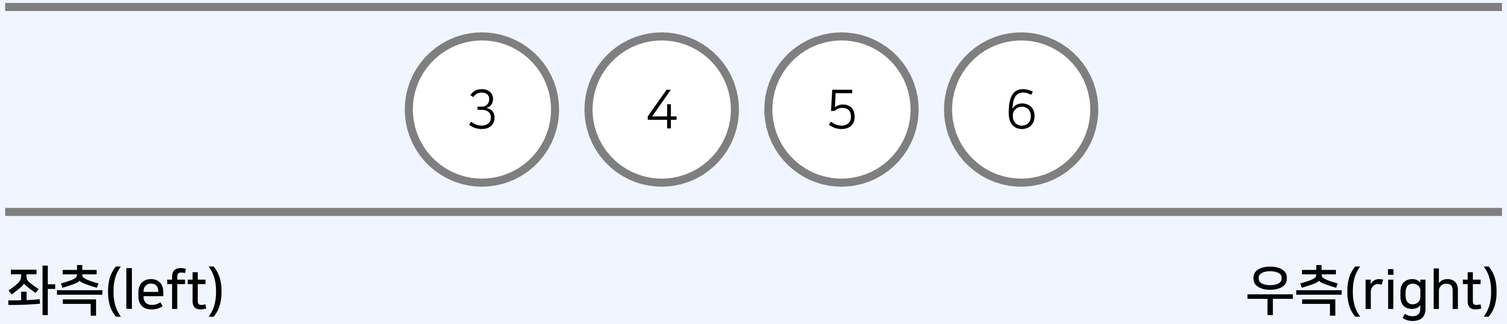


덱(Deque)

- 덱에 여러 개의 데이터를 삽입하고 삭제하는 예시를 확인해 보자.

[전체 연산]

- 좌측 삽입 4 – 좌측 삽입 3 – 좌측 삽입 2 – 좌측 삽입 1
- 우측 삽입 5 – 우측 삽입 6 – 우측 삽입 7 – 우측 삽입 8
- 우측 삭제 – 좌측 삭제 – 우측 삭제 – 좌측 삭제



덱(Deque)의 시간 복잡도

- 데이터의 삭제와 삽입 모두에서 $O(1)$의 시간 복잡도가 소요된다.

	연산	수행 시간	설명
1	좌측 삽입(Append Left)	$O(1)$	덱의 가장 왼쪽에 새 데이터를 삽입
2	좌측 삭제(Pop Left)	$O(1)$	덱의 가장 왼쪽에서 데이터를 추출
3	우측 삽입(Append Right)	$O(1)$	덱의 가장 오른쪽에 새 데이터를 삽입
4	우측 삽입(Pop Right)	$O(1)$	덱의 가장 오른쪽에서 데이터를 추출

파이썬의 덱(Deque) 라이브러리

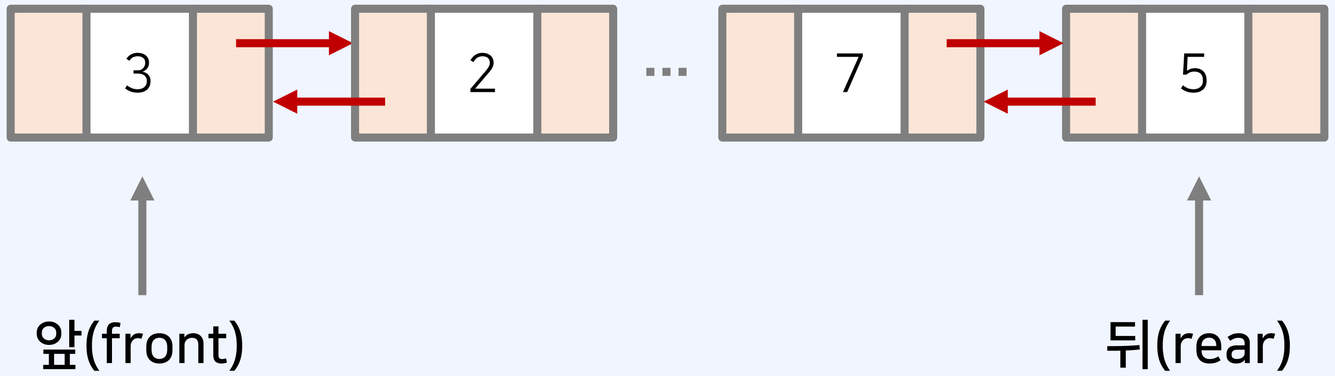
- Python에서는 **덱(deque)** 라이브러리를 사용할 수 있다.
- 아래의 모든 메서드는 최악의 경우 시간 복잡도 $O(1)$을 보장한다.
- **우측 삽입**: $append()$
- **좌측 삽입**: $appendleft()$
- **우측 추출**: $pop()$
- **좌측 추출**: $popleft()$

연결 리스트로 덱 구현하기

- 덱(deque)을 연결 리스트로 구현하면, 삽입과 삭제에 있어서 $O(1)$을 보장할 수 있다.
- 연결 리스트로 구현할 때는 **앞(front)**과 **뒤(rear)** 두 개의 포인터를 가진다.
- **앞(front)**: 가장 좌측에 있는 데이터를 가리키는 포인터
- **뒤(rear)**: 가장 우측에 있는 데이터를 가리키는 포인터

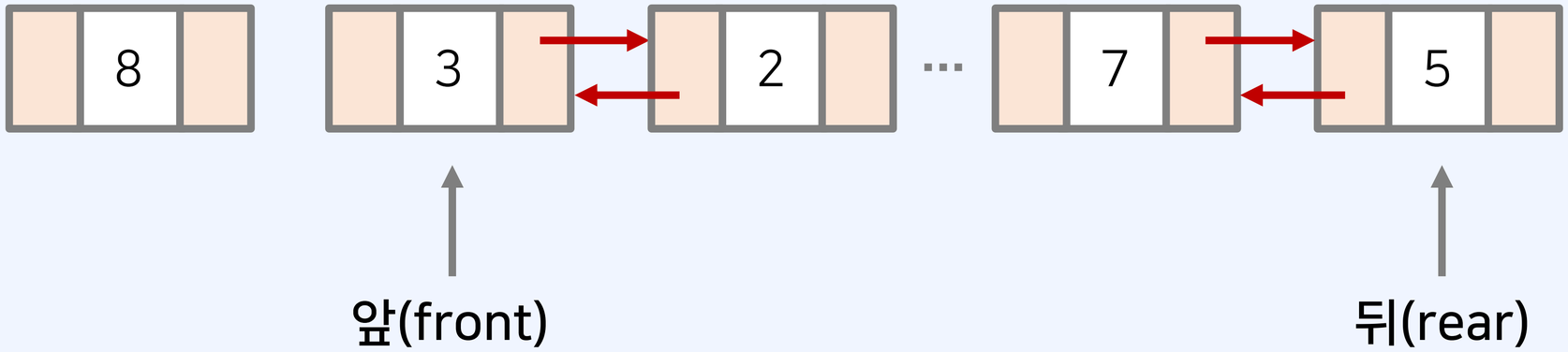
연결 리스트로 덱 구현하기

- **삽입**과 **삭제**의 구현 방법은 스택 및 큐와 유사하다.
- **앞**(front)과 **뒤**(rear)에 대하여 대칭적으로 구현할 수 있다.



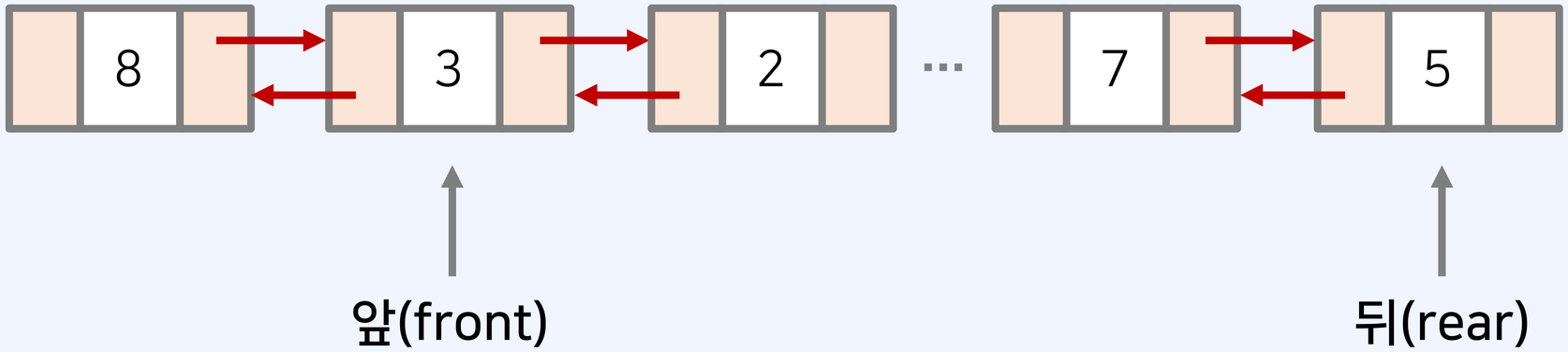
연결 리스트로 덱 구현하기 – 좌측 삽입 연산

- **좌측 삽입**할 때는 앞(front) 위치에 데이터를 넣는다.
- 값으로 8을 갖는 새로운 데이터가 삽입되었을 때 예시)



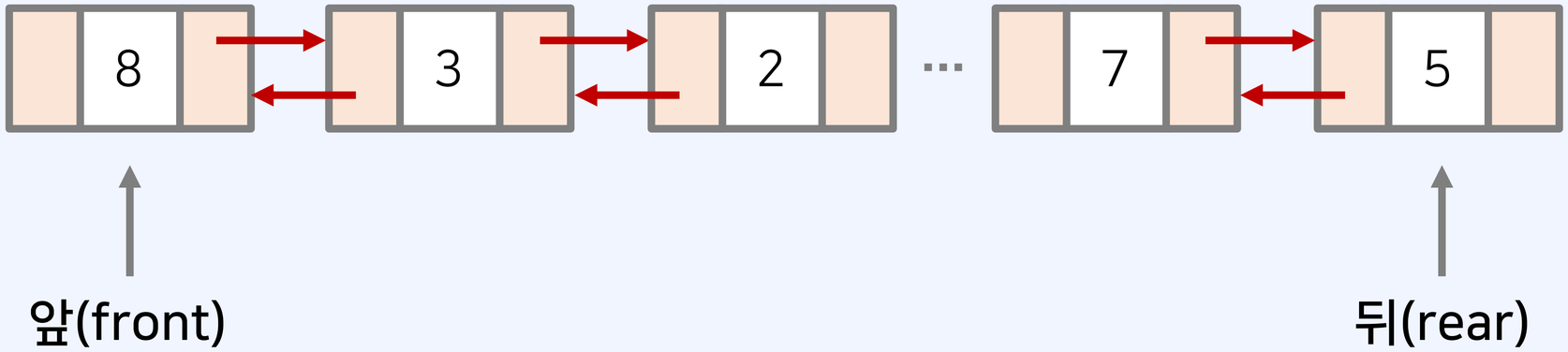
연결 리스트로 덱 구현하기 – 좌측 삽입 연산

- **좌측 삽입**할 때는 앞(front) 위치에 데이터를 넣는다.
- 값으로 8을 갖는 새로운 데이터가 삽입되었을 때 예시)



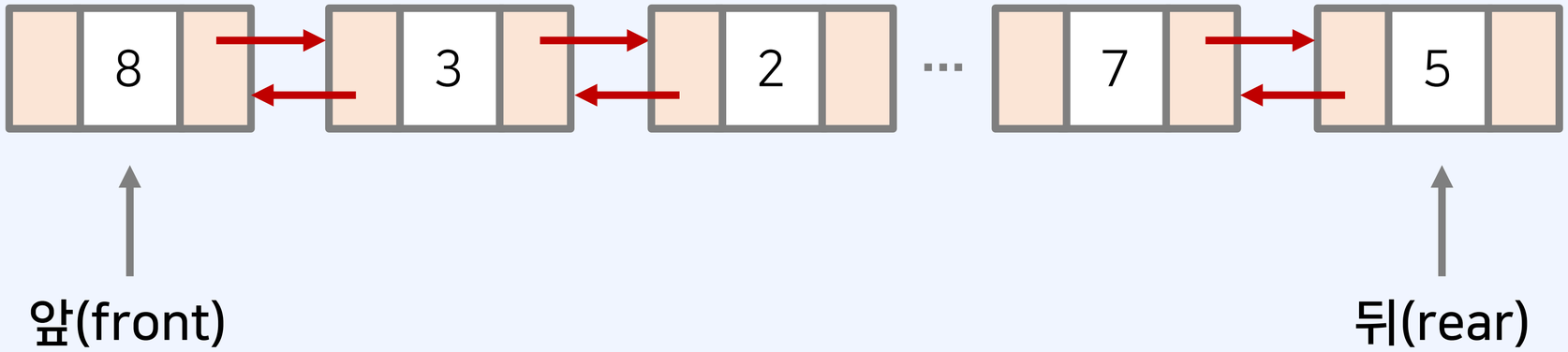
연결 리스트로 덱 구현하기 – 좌측 삽입 연산

- **좌측 삽입**할 때는 앞(front) 위치에 데이터를 넣는다.
- 값으로 8을 갖는 새로운 데이터가 삽입되었을 때 예시)



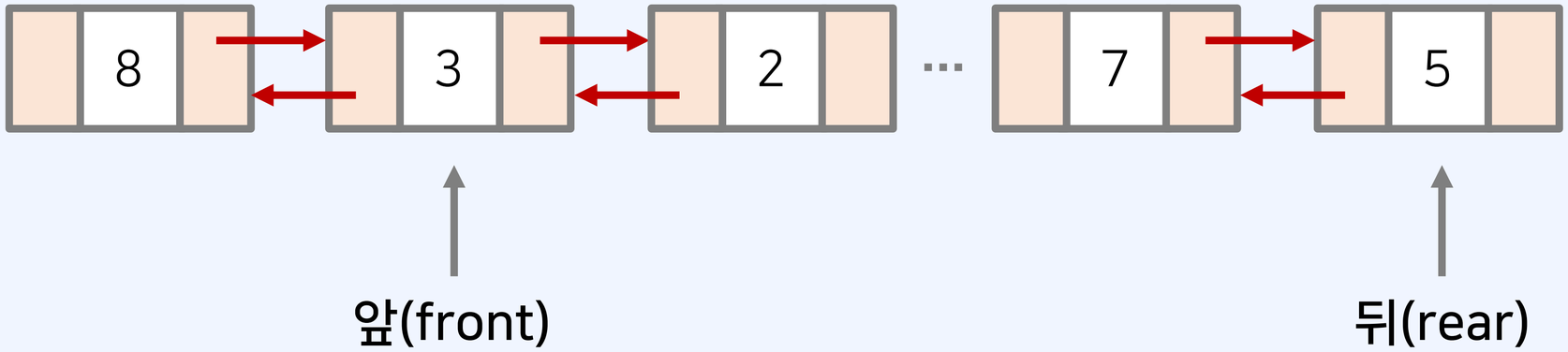
연결 리스트로 덱 구현하기 – 좌측 삭제 연산

- **삭제**할 때는 앞(front) 위치에서 데이터를 꺼낸다.
- 하나의 데이터를 삭제할 때의 예시)



연결 리스트로 덱 구현하기 – 좌측 삭제 연산

- **삭제**할 때는 앞(front) 위치에서 데이터를 꺼낸다.
- 하나의 데이터를 삭제할 때의 예시)



Python에서 덱(Deque)을 사용하는 경우

- 기본적인 Python의 리스트 자료형은 큐(queue)의 기능을 제공하지 않는다.
- 가능하다면 Python에서 제공하는 덱(deque) 라이브러리를 사용한다.
- 큐(queue)의 기능이 필요할 때는 덱 라이브러리를 사용하는 것을 추천한다.
- 삽입과 삭제에 대하여 모두 시간 복잡도 $O(1)$이 요구된다.