

09. 큐, 덱

큐?

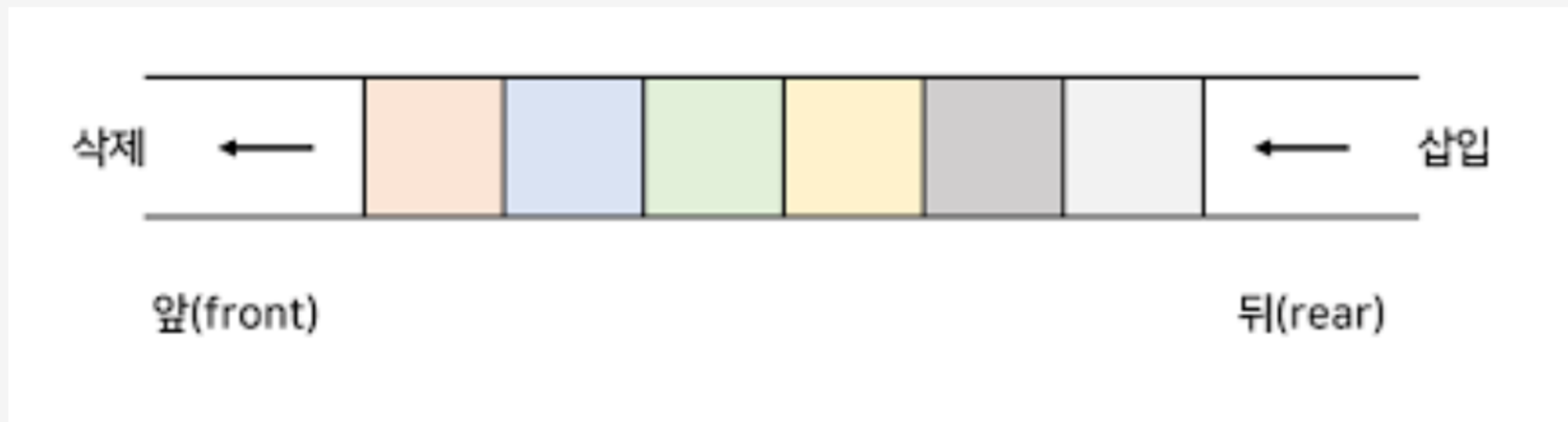
queue : 표를 사러 일렬로 늘어선 사람들로 이루어진 줄



큐?

선입 선출의 자료구조(First In First Out; FIFO)

먼저 들어오는 데이터가 먼저 나 가게 된다.



큐 연산

push : 큐에 자료를 넣는 연산

pop : 큐에서 자료를 빼는 연산

front : 큐의 가장 앞에 있는 자료를 참조하는 연산

back : 큐의 가장 뒤에 있는 자료를 참조하는 연산

empty : 스택이 비어있는지 확인하는 연산

size : 스택에 저장되어 있는 자료의 개수를 참조하는 연산

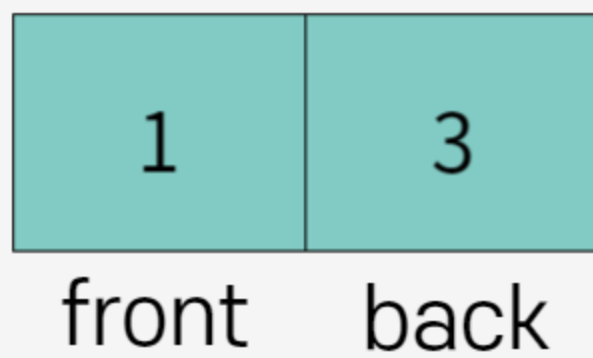
큐 연산

push : 큐에 자료를 넣는 연산

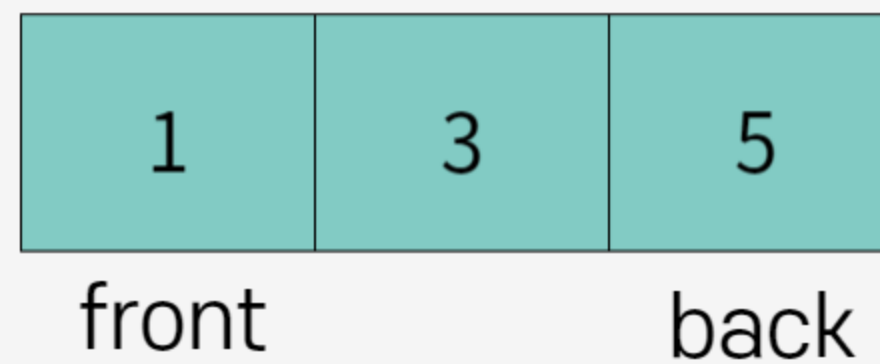


큐 연산

push : 큐에 자료를 넣는 연산

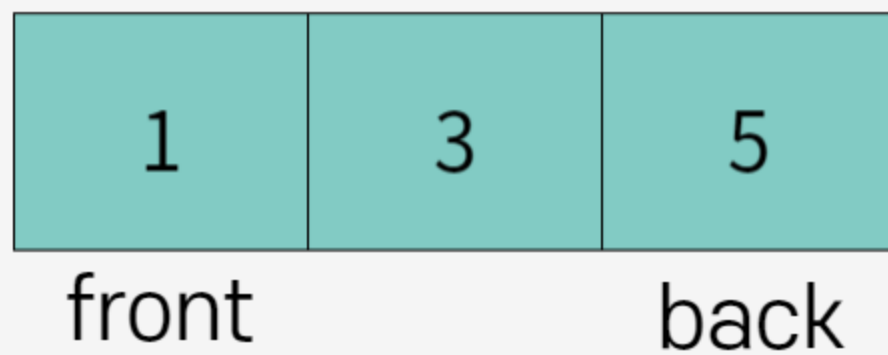


push 5

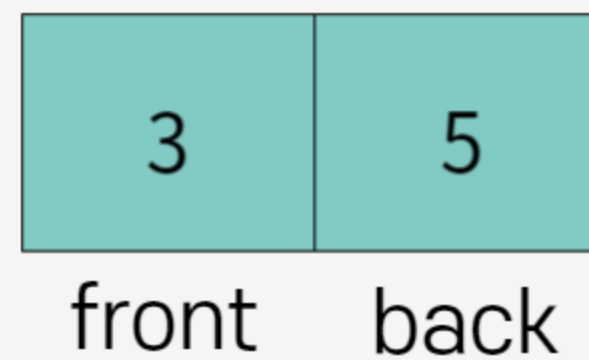


큐 연산

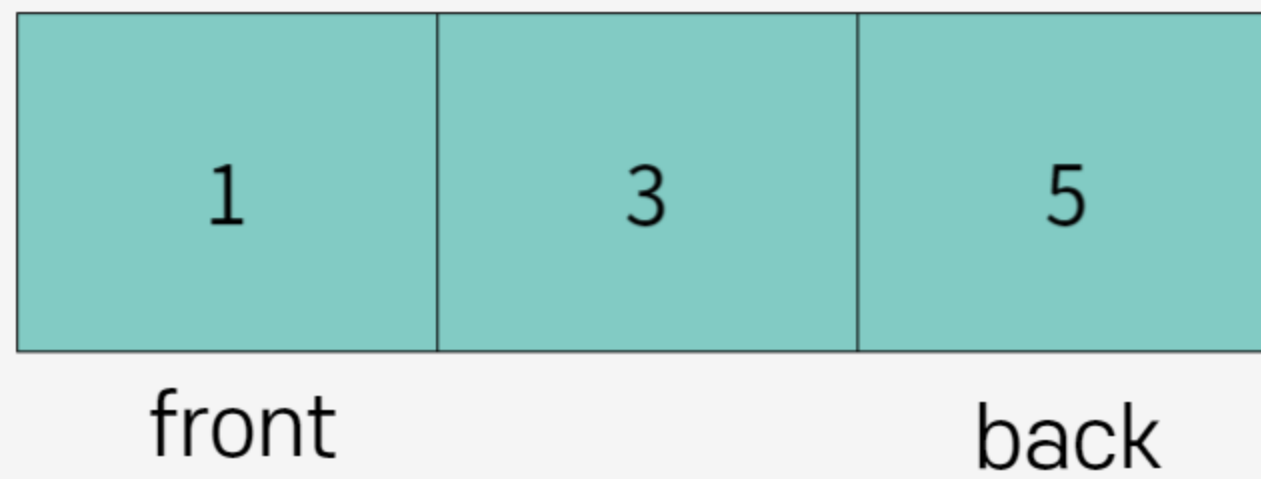
pop : 큐에서 자료를 빼는 연산



pop



큐 연산



큐 구현

기본 선언

```
struct Queue {  
    int data[10010]; //데이터를 저장할 공간  
    int begin, end;   // 큐의 맨 앞, 맨 뒤를 가리킴  
};
```



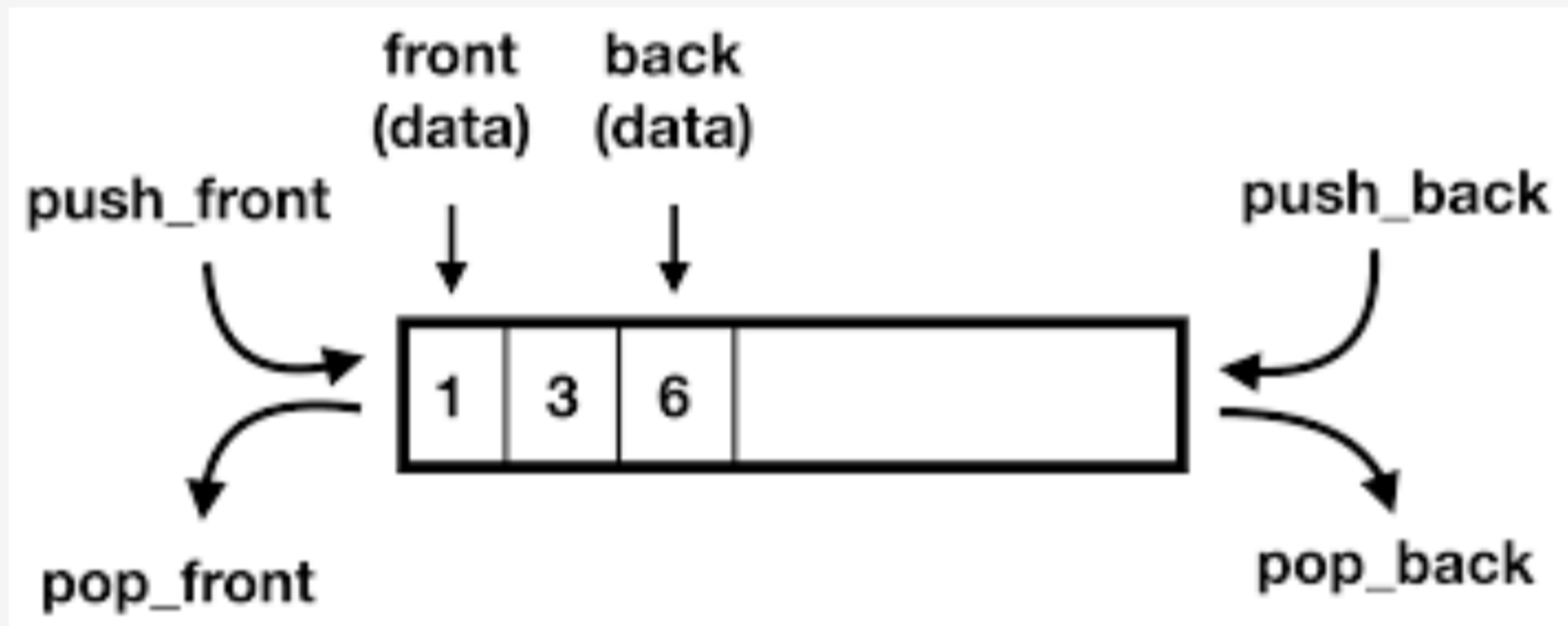
큐 구현

문제

BOJ 10845번 큐

덱?

양쪽 끝에서 삽입과 삭제가 모두 가능한 자료구조



덱 연산

push_front : 앞에 원소를 삽입

push_back : 뒤에 원소를 삽입

pop_front : 맨 앞 원소를 삭제

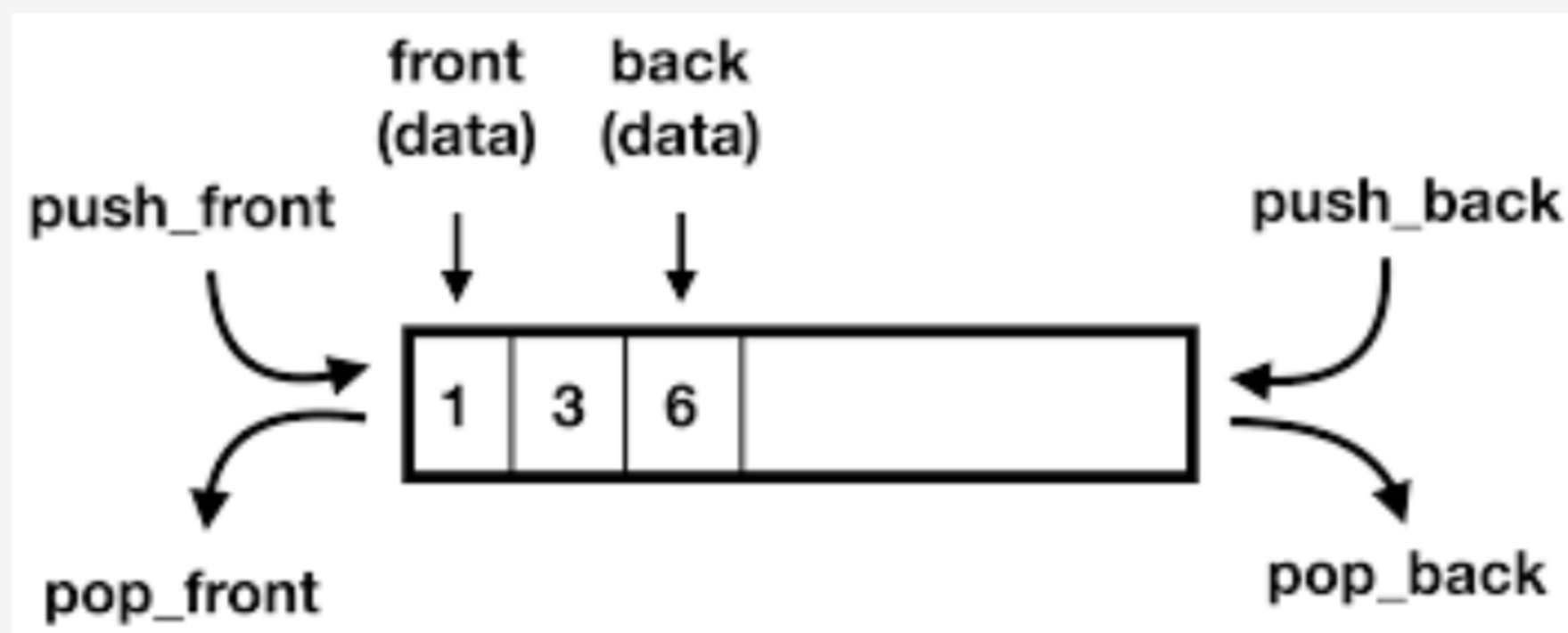
pop_back : 맨 뒤 원소를 삭제

front : 맨 앞 원소 참조

back : 맨 뒤 원소 참조

empty : 비어있는지 확인

size : 원소 개수 리턴



덱 특징

앞, 뒤로 삽입, 삭제가 가능하다.

vector처럼 임의 접근 반복자를 제공한다. ex) deque[5]

확장 시 relocation에 따른 손실이 커서 최대한 적당히 해야 한다!

덱 구현

참조 - <https://rebas.kr/797>

문제

BOJ 10866번 덱

요세푸스 문제

BOJ 1158번

1번 부터 N번 까지 N명의 사람이 원을 이루면서 앉아있다.

순서대로 K번째 사람을 제거할 때 사람들이 제거되는 순서는?

요세푸스 문제

BOJ 1158번

K번째 사람이 제거되면, 남은 사람들 중 "그 다음 K번째 사람"을 제거하는 문제

문제 그대로 큐를 이용해 시뮬레이션하면 된다.

요세푸스 문제

BOJ 1158번

$N = 7, K = 3$

AC

BOJ 5430번

AC는 정수 배열에 연산을 하기 위해 만든 언어이다.

이 언어에는 두 가지 함수 R(뒤집기)과 D(버리기)가 있다.

함수 R은 배열에 있는 숫자의 순서를 뒤집는 함수이고,

D는 첫 번째 숫자를 버리는 함수이다. 배열이 비어있는데 D를 사용한 경우에는 에러가 발생한다.

배열의 초기값과 수행할 함수가 주어졌을 때, 최종 결과를 구하는 문제

AC

BOJ 5430번

문제점

1. 입출력 양식이 깔끔하지 못하다! [1,2,3,4]

해결

1. 우선 문자열(string)로 모두 받는다.
2. 숫자가 나오면 기존 수 $\times 10$ + 나온 숫자
3. ',' or ']' 이 나오면 수 저장

AC

BOJ 5430번

문제점

2. 뒤집기(R)에 대한 시간 복잡도

정수 배열의 앞, 뒤 원소를 차례로 swap 하면 $\rightarrow O(N)$

만약 원소가 10만 개고, 10만 개의 명령이 모두 'R' 이라면? \rightarrow 시간 초과

AC

BOJ 5430번

해결

앞, 뒤로 삽입, 삭제가 가능한 deque를 이용한다.

1. 미리 방향을 저장하는 변수를 하나 만든다. (0 : front, 1 : back)
2. 'R' 이 나오면 방향만 바꿔준다.
3. 'D' 가 나오면 방향에 따라 버릴 원소를 선택한다.
4. 마지막엔 방향에 따라 원소를 순회하며 출력한다. (idx 사용 가능!)

참고

queue는 다음 그래프 탐색 시간에 계속 써야하니 꼭 알아두세요!