

2021 신촌 연합 여름캠프 초급반 3회차

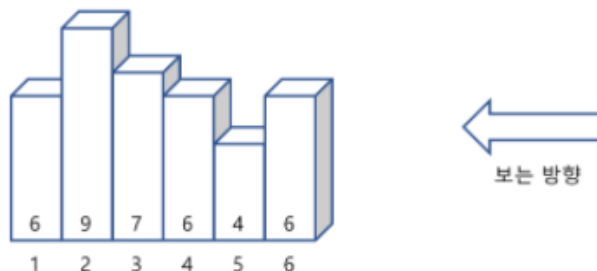
Stack, Queue, Deque

서강대학교 박재형

BOJ 17608 (막대기)

문제

아래 그림처럼 높이만 다르고 (같은 높이의 막대기가 있을 수 있음) 모양이 같은 막대기를 일렬로 세운 후, 왼쪽부터 차례로 번호를 붙인다. 각 막대기의 높이는 그림에서 보인 것처럼 순서대로 6, 9, 7, 6, 4, 6 이다. 일렬로 세워진 막대기를 오른쪽에서 보면 보이는 막대기가 있고 보이지 않는 막대기가 있다. 즉, 지금 보이는 막대기보다 뒤에 있고 높이가 높은 것이 보이게 된다. 예를 들어, 그림과 같은 경우엔 3개(6번, 3번, 2번)의 막대기가 보인다.



N개의 막대기에 대한 높이 정보가 주어질 때, 오른쪽에서 보아서 몇 개가 보이는지를 알아내는 프로그램을 작성하려고 한다.

입력

첫 번째 줄에는 막대기의 개수를 나타내는 정수 N ($2 \leq N \leq 100,000$)이 주어지고 이어지는 N줄 각각에는 막대기의 높이를 나타내는 정수 h ($1 \leq h \leq 100,000$)가 주어진다.

BOJ 17608 (막대기)

Sol 1) Naive하게 해결하기

- ✓ i 번째 막대기가 오른쪽에서 보이는가?
- ✓ $N = 100,000$
- ✓ $O(N^2) \rightarrow$ 시간 초과

BOJ 17608 (막대기)

Sol 2) 뒤에서부터 확인하기

- ✓ 현재 막대기보다 크거나 같은 막대기가 오른쪽에 존재하는가?
- ✓ $O(N)$

```
int right_max = 0;
int ans = 0;
for (int i = n - 1; i >= 0; i--) {
    if (a[i] > right_max) {
        ans++;
        right_max = a[i];
    }
}
```

BOJ 17608 (막대기)

Sol 3) 앞에서부터 확인하기

- ✓ 가려지지 않는 막대기가 될 가능성
- ✓ 현재 막대기로 가려지는 왼쪽의 막대기를 제거

ex) 10 8 6 9 4 7

→ Stack

What?

[Stack, Queue, Deque]

- 특정 위치에서만 원소를 넣고 뺄 수 있는 제한된 '선형 자료구조'
- C++ STL 존재

스택(Stack)

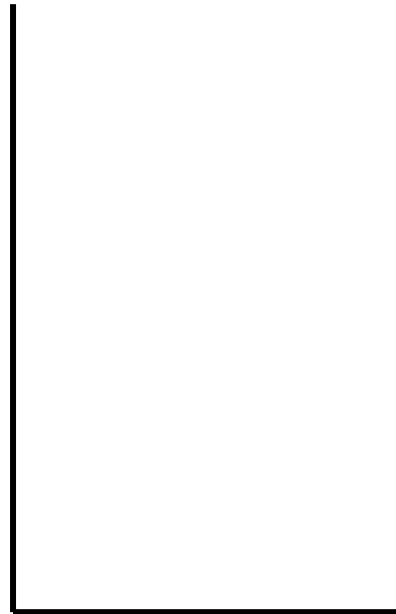
- 한 쪽 끝에서만 원소를 넣고 뺄 수 있는 자료구조
- **LIFO**(Last In First Out)
- push / pop
- 컨테이너 어댑터(Container Adaptor)의 일종
- `#include<stack>`

스택(Stack)

ex) 10 8 6 9 4 7 (막대기 문제)



Size = 0



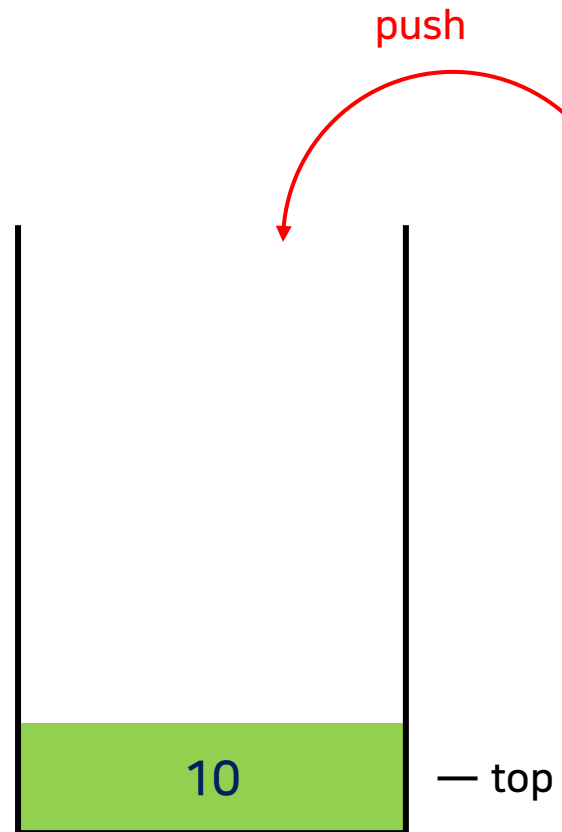
10

스택(Stack)

ex) 10 8 6 9 4 7



Size = 1

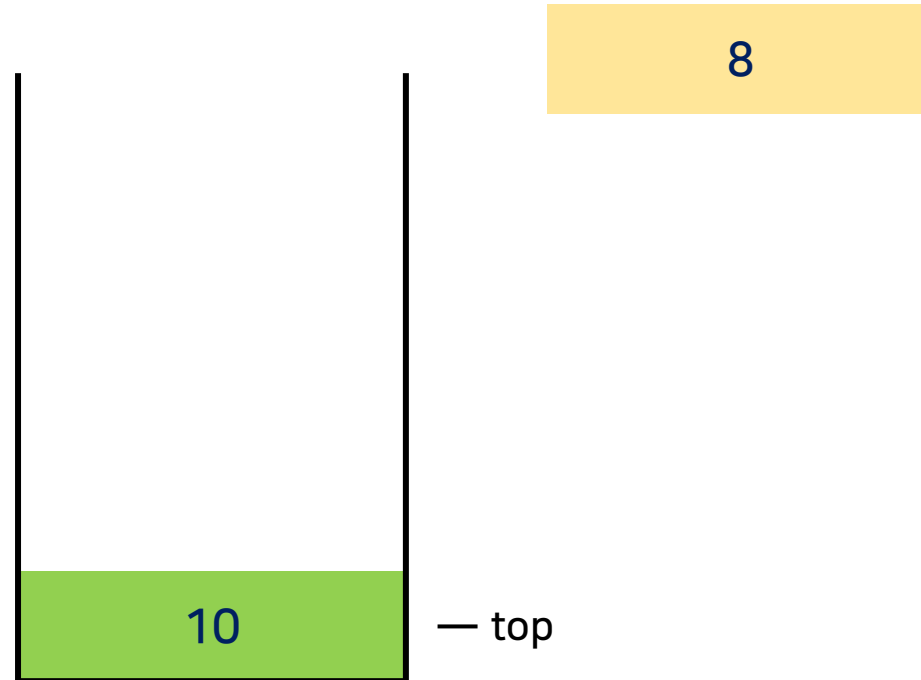


스택(Stack)

ex) 10 8 6 9 4 7



Size = 1

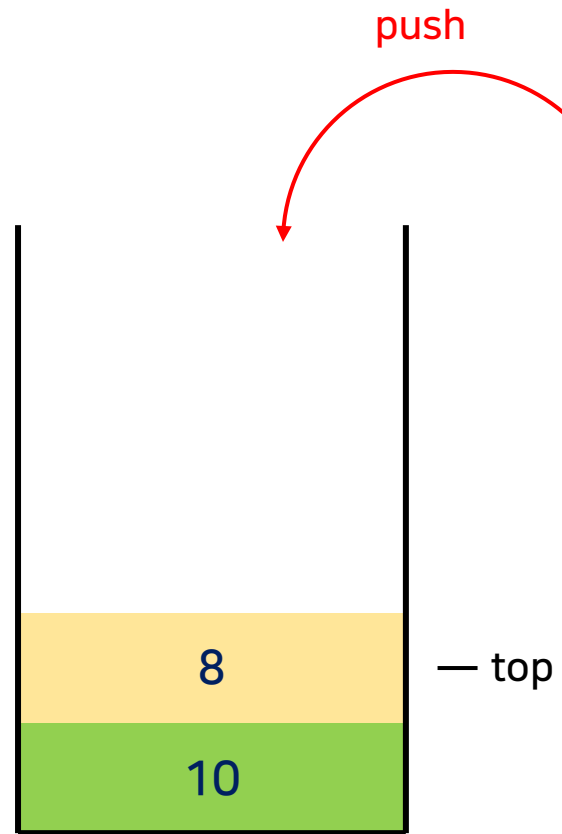


스택(Stack)

ex) 10 8 6 9 4 7



Size = 2

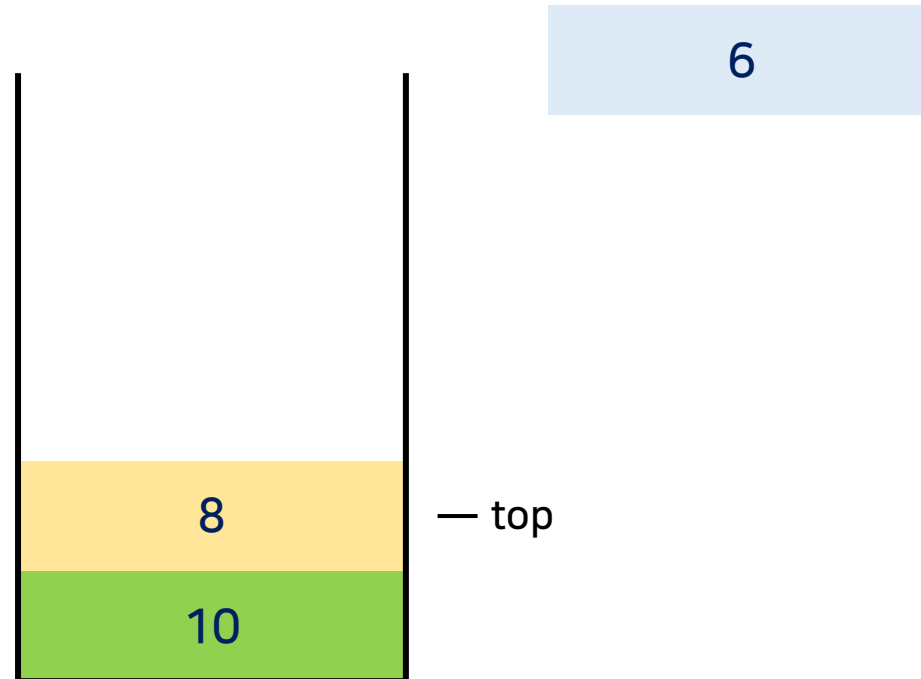


스택(Stack)

ex) 10 8 6 9 4 7



Size = 2

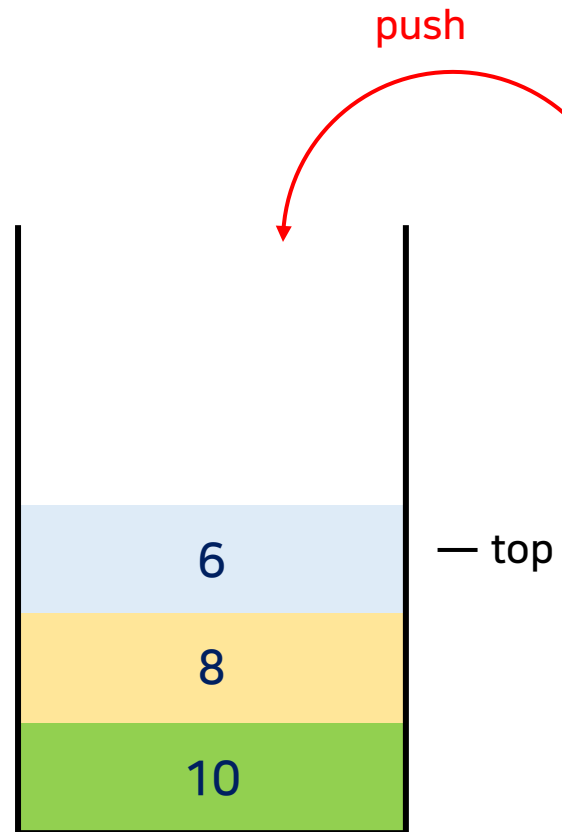


스택(Stack)

ex) 10 8 6 9 4 7



Size = 3

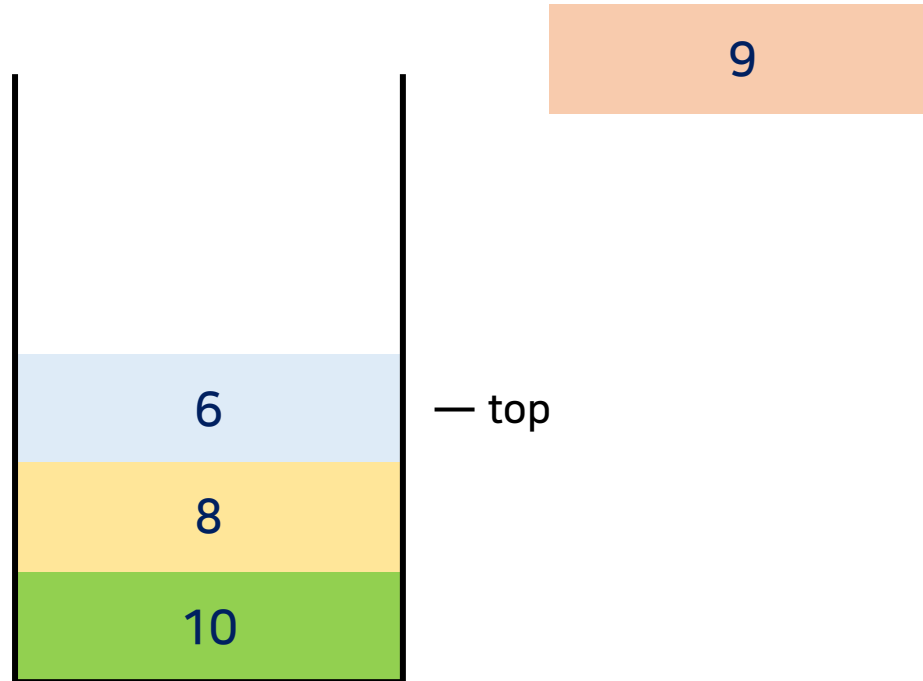


스택(Stack)

ex) 10 8 6 9 4 7



Size = 3

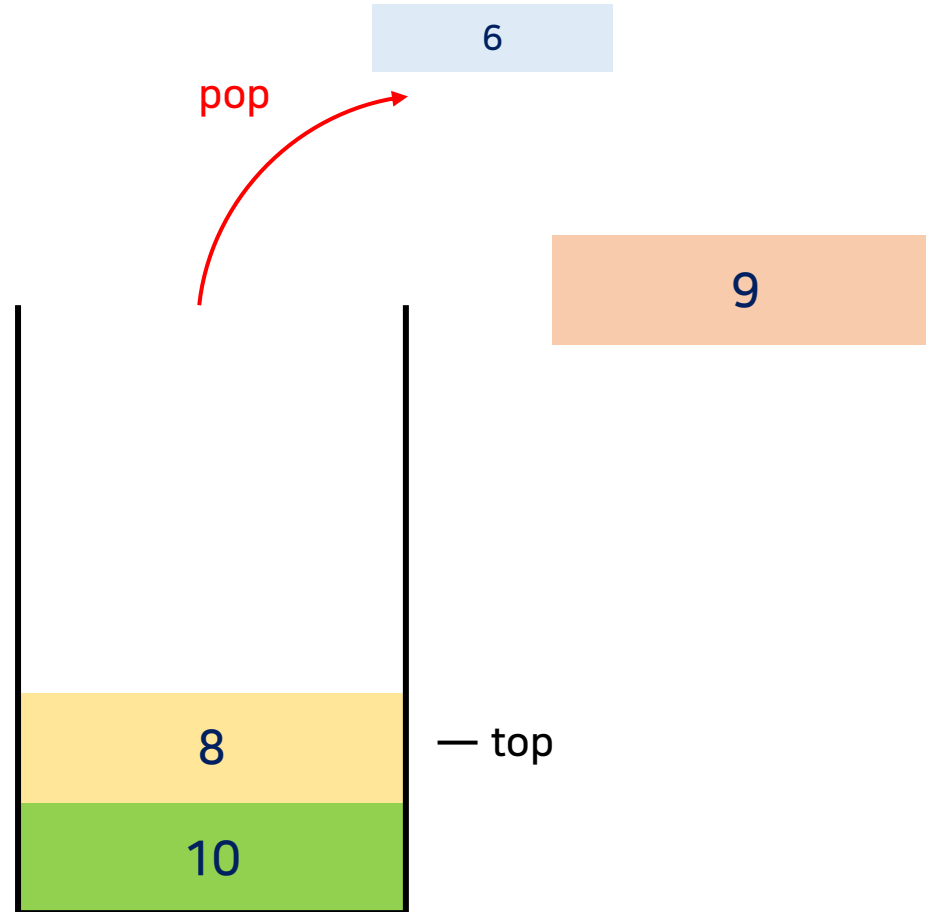


스택(Stack)

ex) 10 8 6 9 4 7



Size = 2

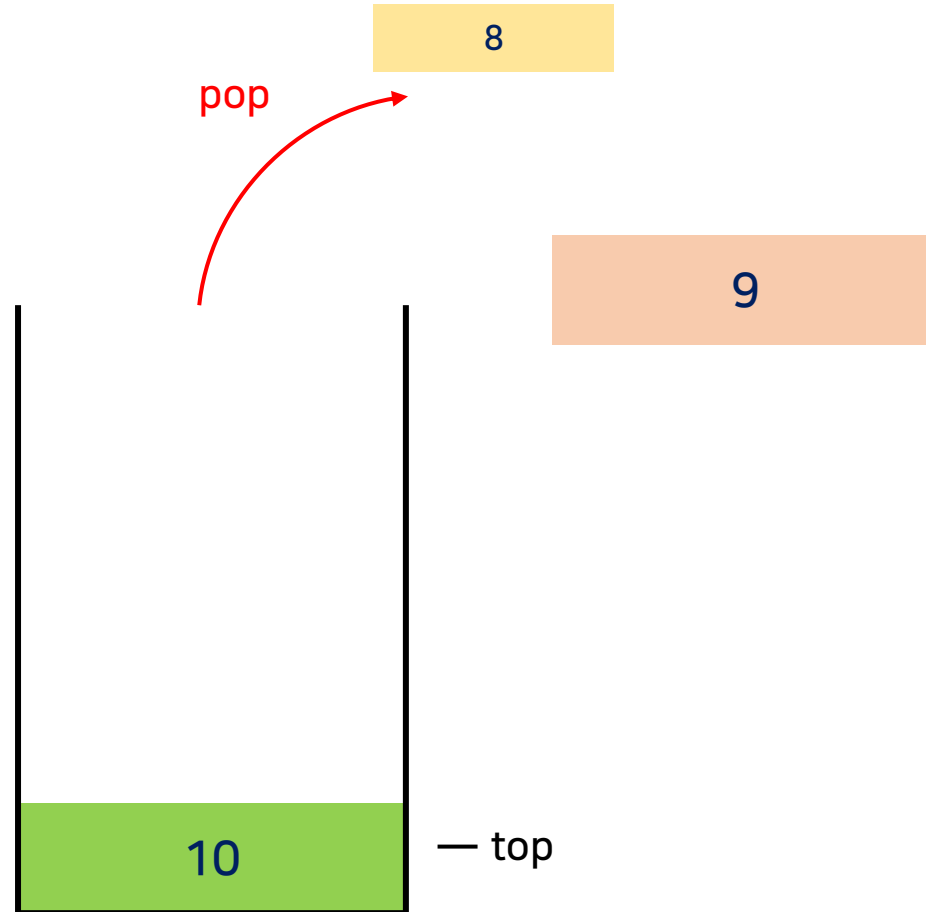


스택(Stack)

ex) 10 8 6 9 4 7



Size = 1

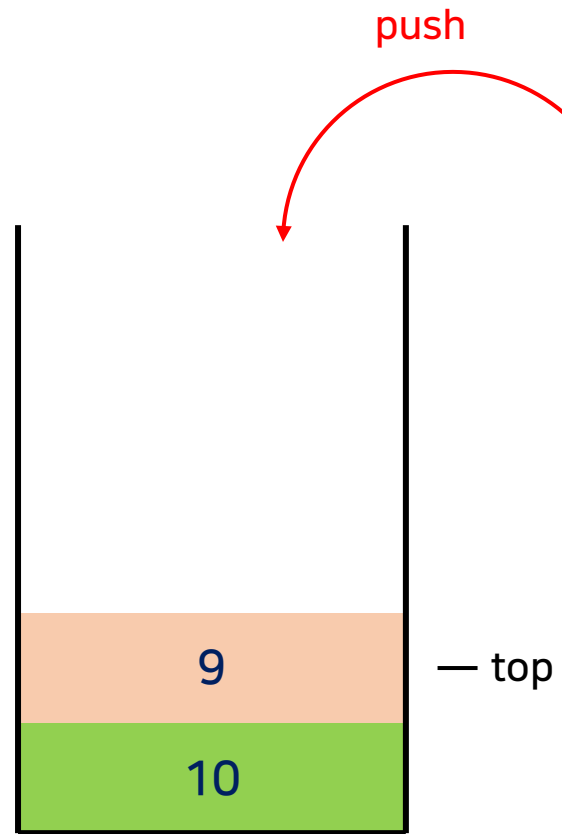


스택(Stack)

ex) 10 8 6 9 4 7



Size = 2

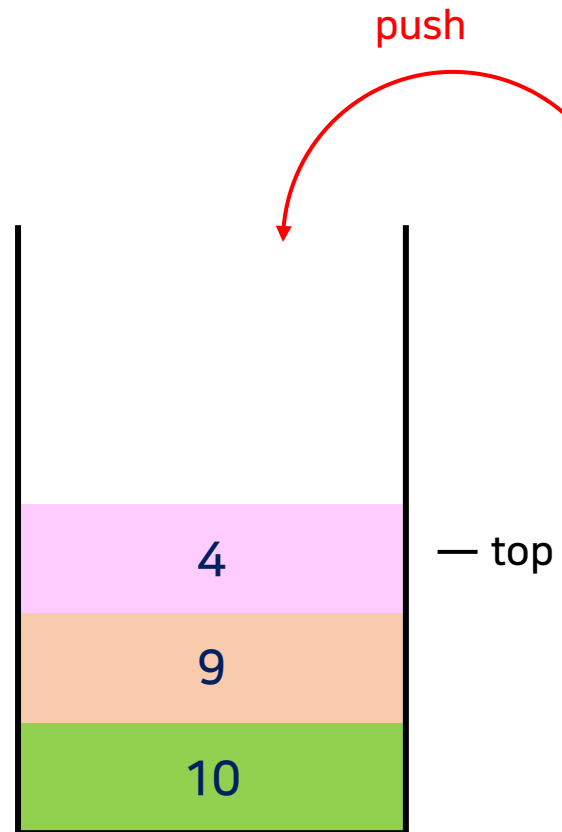


스택(Stack)

ex) 10 8 6 9 4 7



Size = 3

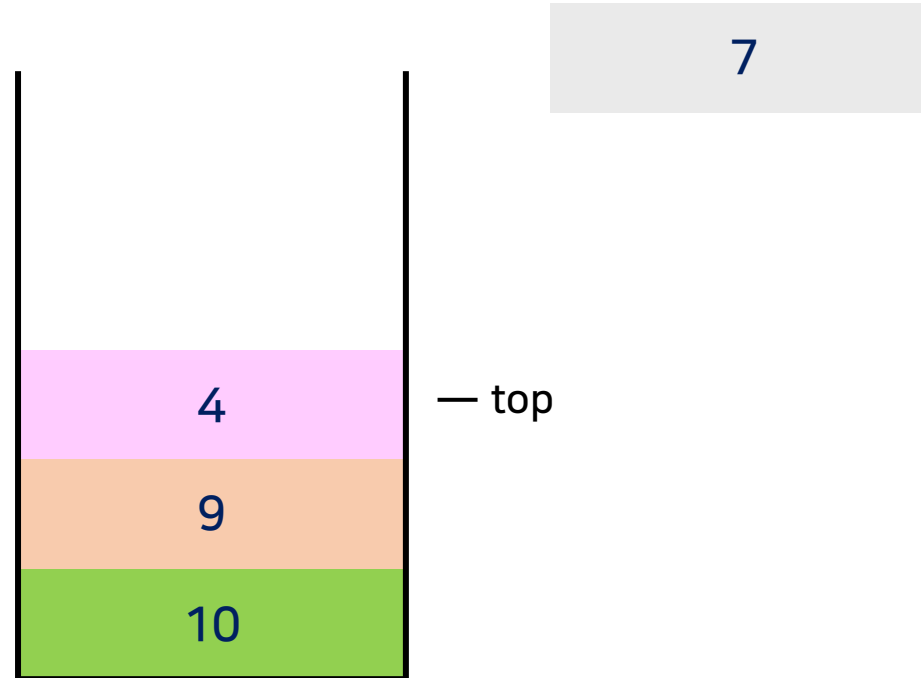


스택(Stack)

ex) 10 8 6 9 4 7



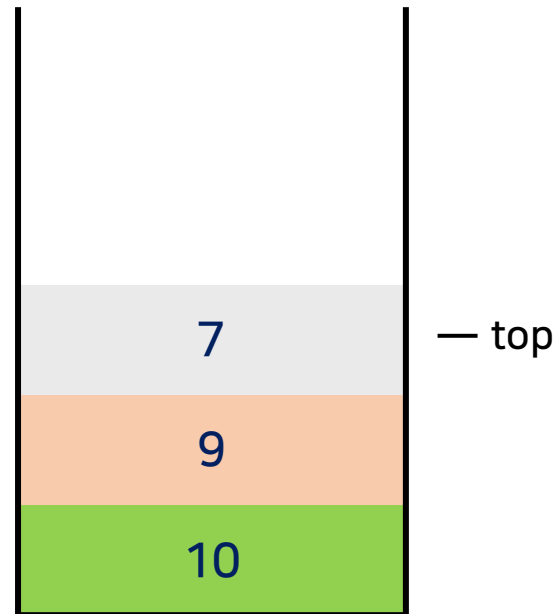
Size = 3



스택(Stack)

ex) 10 8 6 9 4 7
 ↑

Size = 3



Stack in C++

```
template<class T, class Container = deque<T>> class stack;
```

- push(k) : stack에 k를 추가
- pop() : stack에서 제일 마지막에 넣은 원소를 제거
- top() : stack에서 제일 마지막에 넣은 원소를 반환
- size() : stack에 들어있는 원소의 개수 반환
- empty() : stack이 비어있으면 true, 아니면 false 반환

Stack in C++

<BOJ 17608 (막대기) 예시 코드>

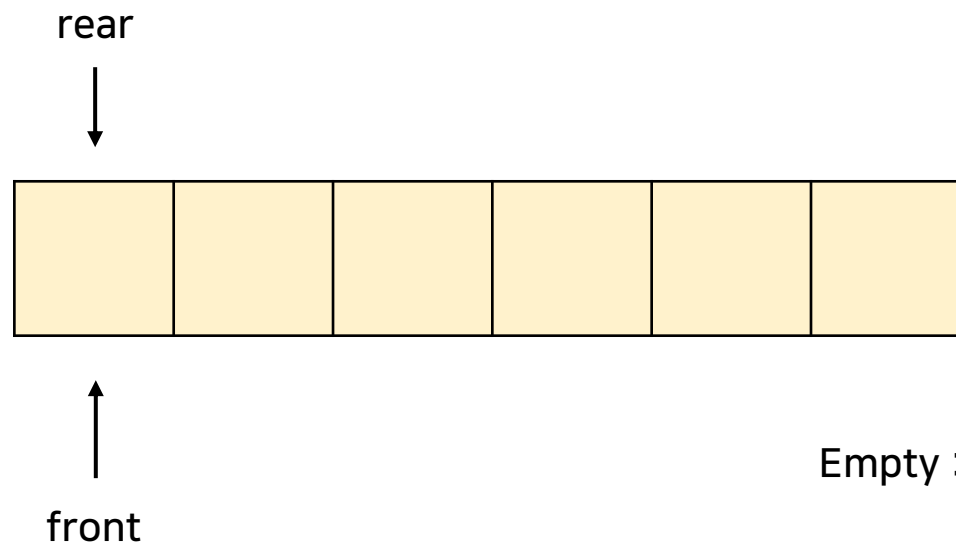
```
stack<int> s;
int n; cin >> n;
for (int i = 1; i <= n; i++) {
    int h; cin >> h;
    if (s.empty()) s.push(h);
    else {
        while (!s.empty() && s.top() <= h) s.pop();
        s.push(h);
    }
}
cout << s.size();
```

큐(Queue)

- 한 쪽에서는 원소를 넣고 다른 쪽에서는 뺄 수만 있는 자료구조
- **FIFO**(First In First Out)
- push / pop
- 컨테이너 어댑터(Container Adaptor)의 일종
- `#include<queue>`

큐(Queue)

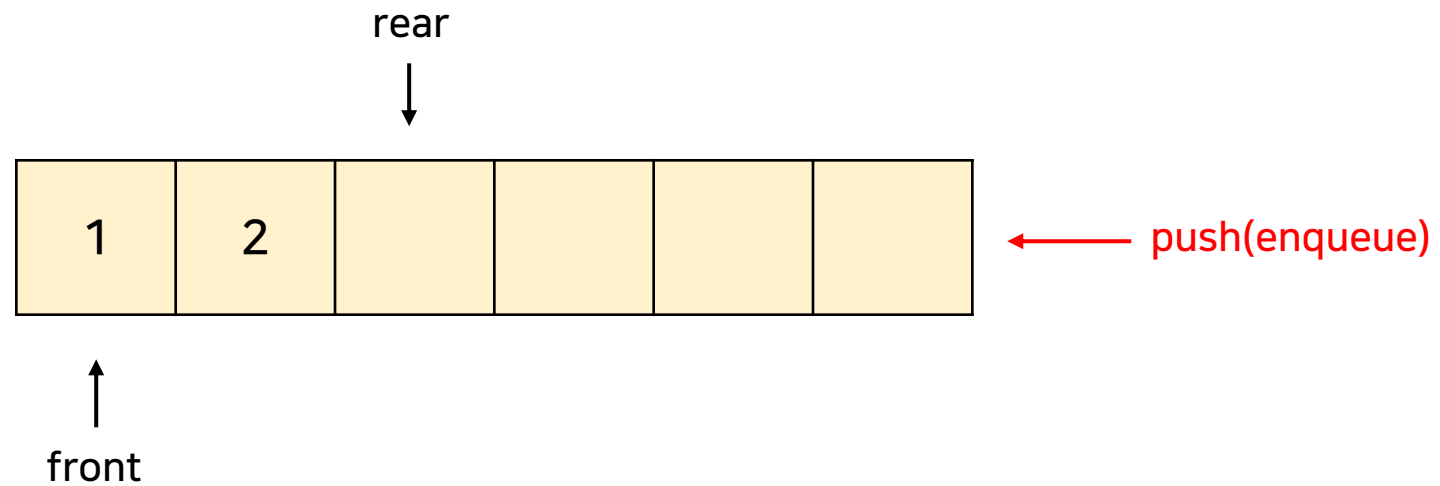
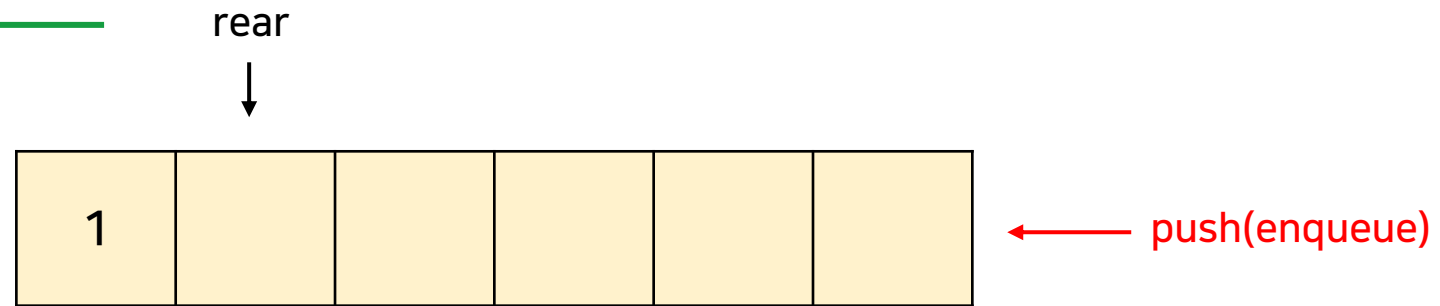
- 선형 큐



Empty : $\text{front} == \text{rear}$

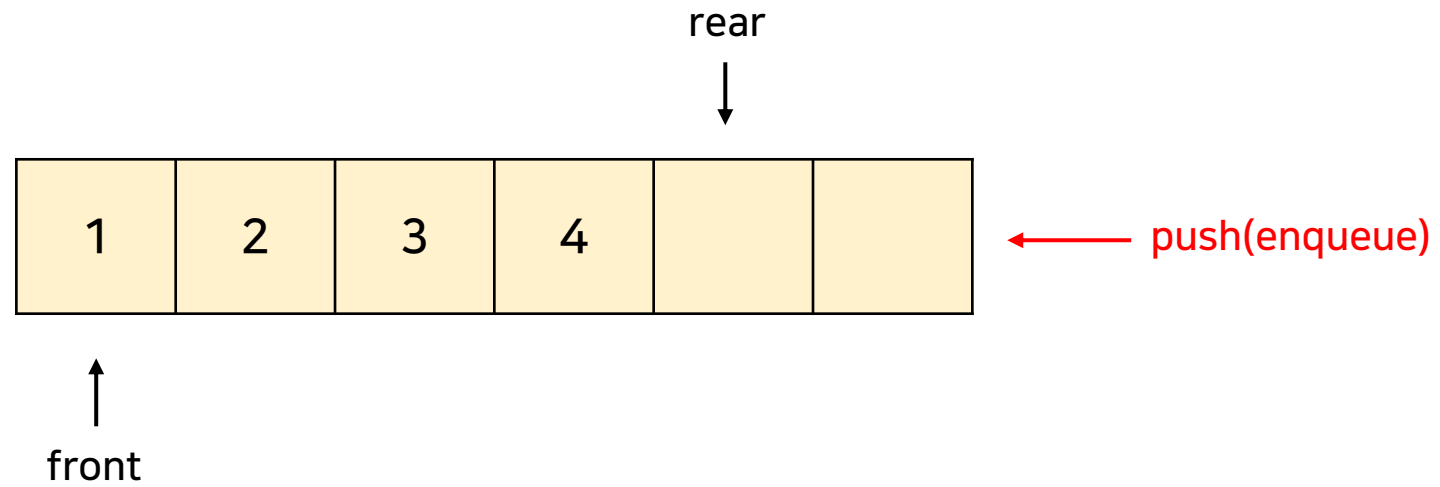
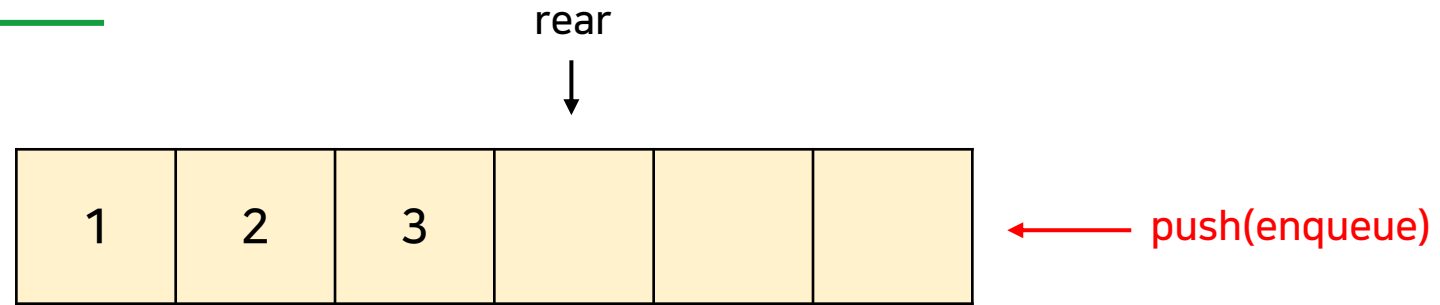
큐(Queue)

- 선형 큐



큐(Queue)

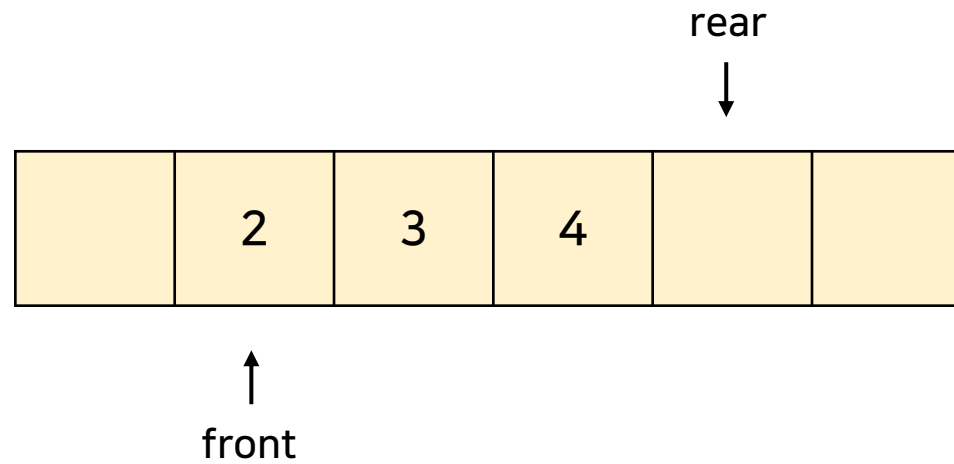
- 선형 큐



큐(Queue)

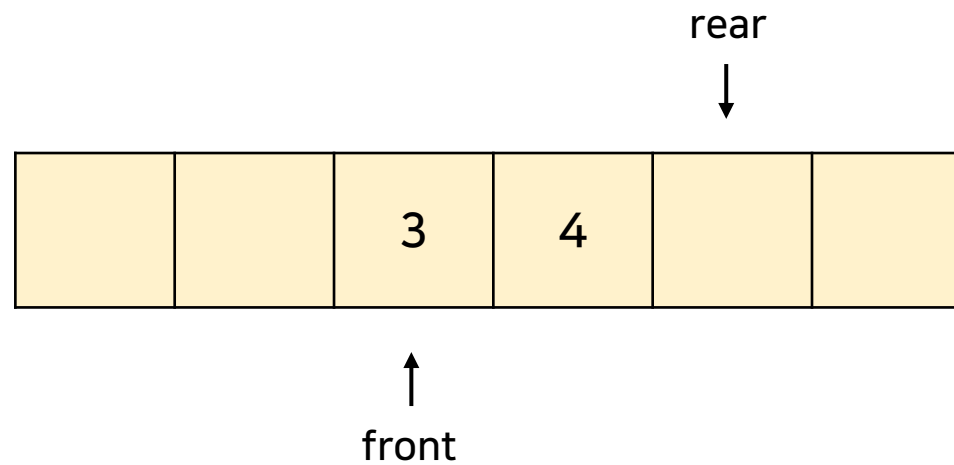
- 선형 큐

pop(dequeue) ←



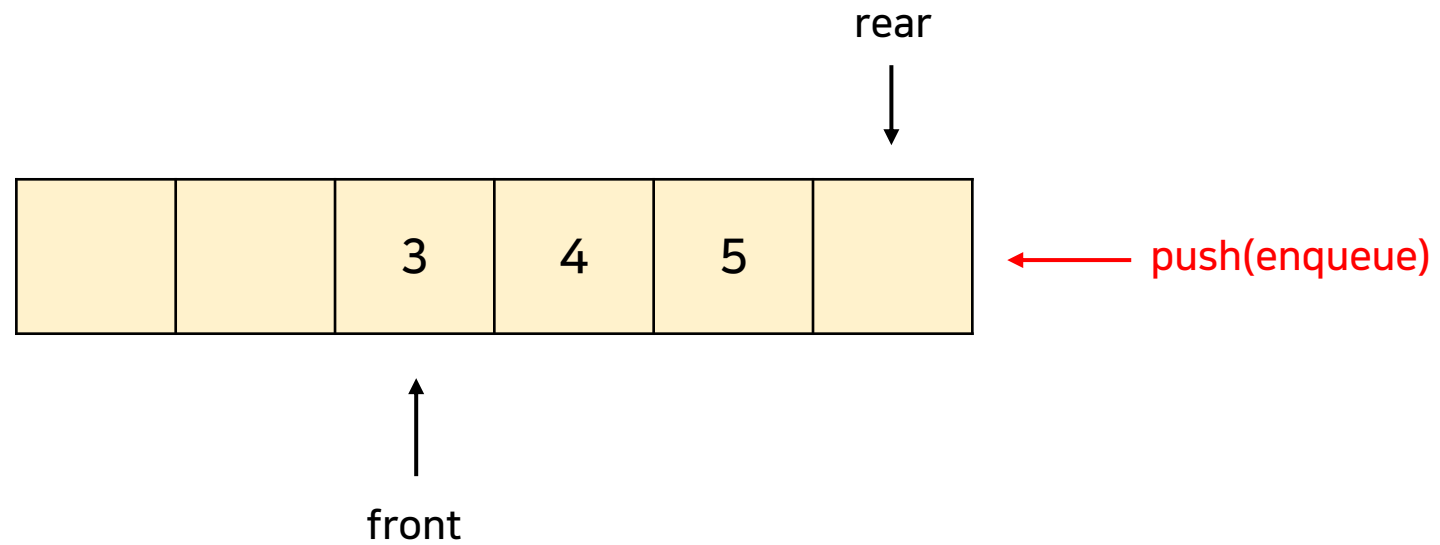
- 단점

pop(dequeue) ←



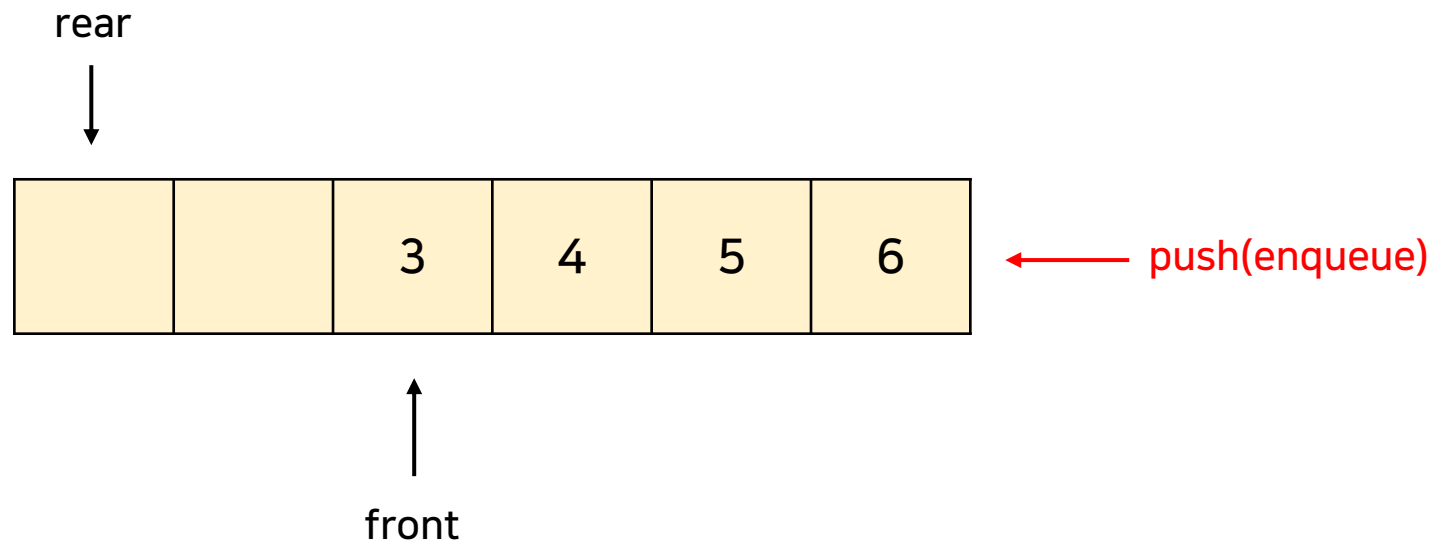
큐(Queue)

- 원형 큐



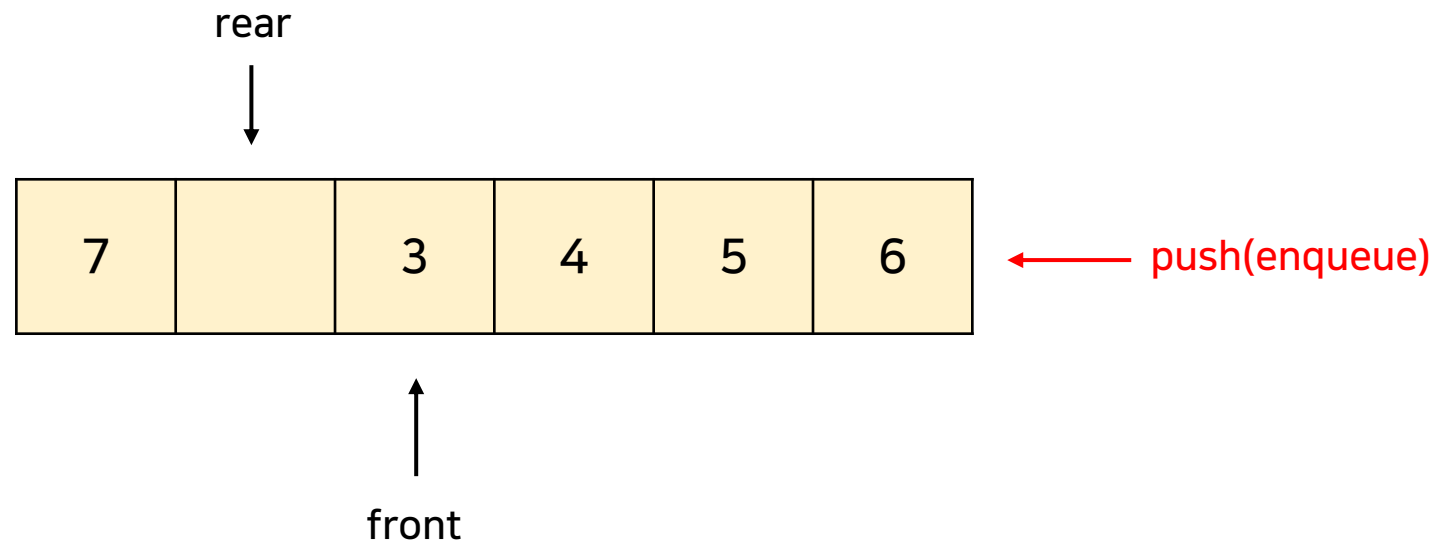
큐(Queue)

- 원형 큐



큐(Queue)

- 원형 큐



Queue in C++

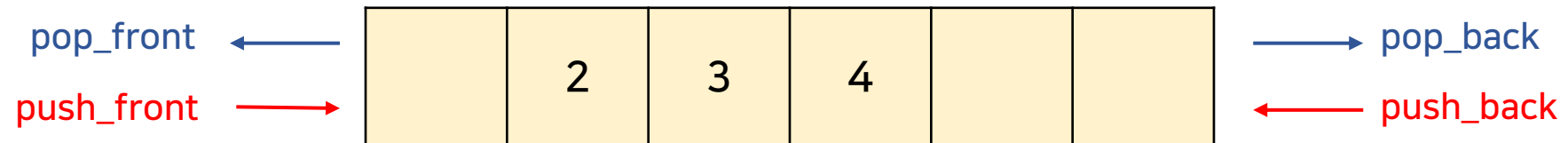
```
template <class T, class Container = deque<T>> class queue;
```

- push(k) : queue에 k를 추가
- pop() : queue의 원소 중 제일 먼저 넣은 원소를 제거
- front() : queue의 원소 중 제일 먼저 넣은 원소를 반환
- back() : queue의 원소 중 제일 마지막에 넣은 원소를 반환
- size() : queue에 들어있는 원소의 개수 반환
- empty() : queue가 비어있으면 true, 아니면 false 반환

덱(Deque)

- 스택 + 큐의 형태
- 양쪽 끝에서 삽입과 삭제가 모두 가능한 자료구조
- 순차 컨테이너(Sequence Container)의 일종
- `#include<deque>`

덱(Deque)



Deque in C++

```
template<class T, class Allocator = std::allocator<T>> class deque;
```

- push_front(k) / push_back(k) : deque의 앞/뒤에 k를 추가
- pop_front() / pop_back() : deque의 제일 앞/뒤의 원소 제거
- front() / back() : deque의 원소 중 제일 앞/뒤의 원소를 반환
- size() : deque에 들어있는 원소의 개수 반환
- empty() : deque가 비어있으면 true, 아니면 false 반환
- vector에서 사용되는 기능들 (Random Access, begin/end, clear, insert 등)

예시 문제

BOJ 1158 ([요세푸스 문제](#))

문제

요세푸스 문제는 다음과 같다.

1번부터 N번까지 N명의 사람이 원을 이루면서 앉아있고, 양의 정수 K ($K \leq N$)가 주어진다. 이제 순서대로 K번째 사람을 제거한다. 한 사람이 제거되면 남은 사람들로 이루어진 원을 따라 이 과정을 계속해 나간다. 이 과정은 N명의 사람이 모두 제거될 때까지 계속된다. 원에서 사람들이 제거되는 순서를 (N, K)-요세푸스 순열이라고 한다. 예를 들어 (7, 3)-요세푸스 순열은 $\langle 3, 6, 2, 7, 5, 1, 4 \rangle$ 이다.

N과 K가 주어지면 (N, K)-요세푸스 순열을 구하는 프로그램을 작성하시오.

입력

첫째 줄에 N과 K가 빈 칸을 사이에 두고 순서대로 주어진다. ($1 \leq K \leq N \leq 5,000$)

BOJ 1158 (요세푸스 문제)

- ✓ 사람들이 원을 이루면서 앉아있음
- ✓ k번째 사람은 k+1번째 사람을 기준으로 가장 마지막 사람
- ✓ k번째 사람이 제거되지 않는다면 마지막에 다시 확인해야 함

→ 큐(Queue)

BOJ 9012 (괄호)

문제

괄호 문자열(Parenthesis String, PS)은 두 개의 괄호 기호인 '(' 와 ')' 만으로 구성되어 있는 문자열이다. 그 중에서 괄호의 모양이 바르게 구성된 문자열을 올바른 괄호 문자열(Valid PS, VPS)이라고 부른다. 한 쌍의 괄호 기호로 된 "(")" 문자열은 기본 VPS 이라고 부른다. 만일 x 가 VPS 라면 이것을 하나의 괄호에 넣은 새로운 문자열 "(x)"도 VPS 가 된다. 그리고 두 VPS x 와 y를 접합(concatenation)시킨 새로운 문자열 xy도 VPS 가 된다. 예를 들어 "(()())"와 "((()))" 는 VPS 이지만 "(()(", "(()())()", 그리고 "(()" 는 모두 VPS 가 아닌 문자열이다.

여러분은 입력으로 주어진 괄호 문자열이 VPS 인지 아닌지를 판단해서 그 결과를 YES 와 NO 로 나타내어야 한다.

BOJ 9012 (괄호)

- ✓ 올바른 괄호 문자열이 되기 위한 조건?
- ✓ '(' 의 개수 == ')'의 개수
- ✓ 올바른 괄호 문자열 생략 가능
- ✓ ')'는 앞에 있으면서 가장 뒤에 존재하는 '('와 연결

→ 스택(Stack)

BOJ 3078 (좋은 친구)

[문제 요약]

- 학생의 이름이 성적순으로 주어짐 (2글자 ~ 20글자)
- 좋은 친구의 조건
 - 1) 자신의 등수와 차이가 K 이하
 - 2) 이름의 길이가 같음
- 좋은 친구가 총 몇쌍인가?

BOJ 3078 (좋은 친구)

- ✓ 학생들이 성적순으로 주어짐 → 순서대로 탐색
- ✓ X등 학생과 좋은 친구가 될 수 있는 등수 범위 : $[X-K, X+K]$
- ✓ $X-K$ 등 학생은 $X+1$ 등 학생과 좋은 친구가 될 수 없음 → 제외

→ 큐(Queue)

- ✓ 현재 큐에 있으면서 길이가 같은 학생의 수

BOJ 18115 ([카드 놓기](#))

문제

수현이는 카드 기술을 연습하고 있다. 수현이의 손에 들린 카드를 하나씩 내려놓아 바닥에 쌓으려고 한다. 수현이가 쓸 수 있는 기술은 다음 3가지다.

1. 제일 위의 카드 1장을 바닥에 내려놓는다.
2. 위에서 두 번째 카드를 바닥에 내려놓는다. 카드가 2장 이상일 때만 쓸 수 있다.
3. 제일 밑에 있는 카드를 바닥에 내려놓는다. 카드가 2장 이상일 때만 쓸 수 있다.

수현이는 처음에 카드 N 장을 들고 있다. 카드에는 1부터 N 까지의 정수가 중복되지 않게 적혀 있다. 기술을 N 번 사용하여 카드를 다 내려놓았을 때, 놓여 있는 카드들을 확인했더니 위에서부터 순서대로 1, 2, ..., N 이 적혀 있었다!

놀란 수현이는 처음에 카드가 어떻게 배치되어 있었는지 궁금해졌다. 처음 카드의 상태를 출력하여라.

입력

첫 번째 줄에는 N ($1 \leq N \leq 10^6$)이 주어진다.

두 번째 줄에는 길이가 N 인 수열 A 가 주어진다. A_i 가 x 이면, i 번째로 카드를 내려놓을 때 x 번 기술을 썼다는 뜻이다. A_i 는 1, 2, 3 중 하나이며, A_n 은 항상 1이다.

BOJ 18115 (카드 놓기)

- ✓ 최종적으로 위에서부터 1, 2, ... N이 쌓여 있음
- ✓ N, N-1, ... 2, 1 순서대로 바닥에 내려놔야 함
- ✓ 카드 더미를 거꾸로 구성해보자!

→ 덱(Deque)

필수 / 연습문제

[필수문제]

- [BOJ 2504] 괄호의 값
- [BOJ 2304] 창고 다각형
- [BOJ 10828] 스택
- [BOJ 3078] 좋은 친구
- [BOJ 20923] 숫자 할리갈리 게임
- [BOJ 18115] 카드 놓기

[연습문제]

- [BOJ 2812] 크게 만들기
- [BOJ 1935] 후위 표기식2
- [BOJ 9012] 괄호
- [BOJ 10799] 쇠막대기
- [BOJ 20301] 반전 요세푸스
- [BOJ 5430] AC
- [BOJ 1966] 프린터 큐
- [BOJ 3190] 뱀