

# 스택, 큐, 데크, 우선순위 큐

---

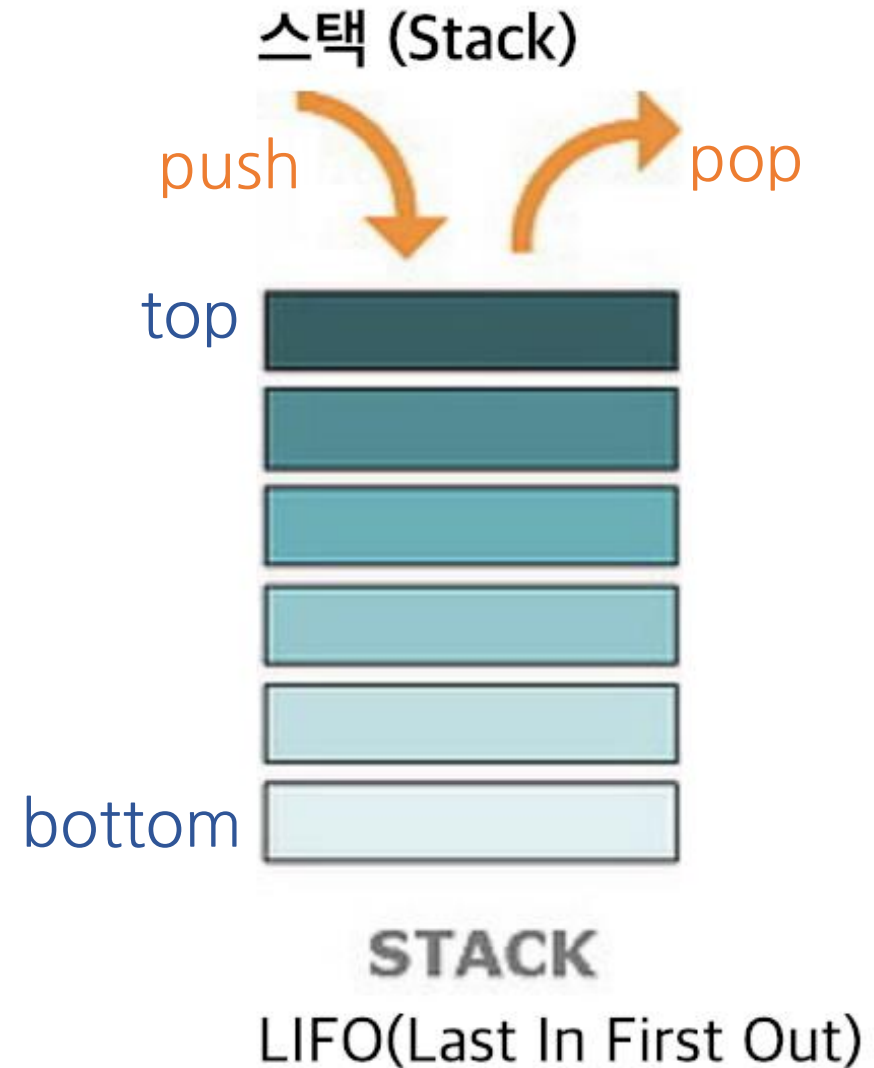
알고리즘 스터디 #2

Author: @hheeseung

# 스택(Stack)

---

- 데이터를 일시적으로 저장하기 위한 자료구조
- **LIFO** 구조 : 가장 나중에 넣은 데이터를 가장 먼저 꺼내는 후입선출 방식



# 스택(Stack): Push, Pop, Peek

## 삽입(Push)

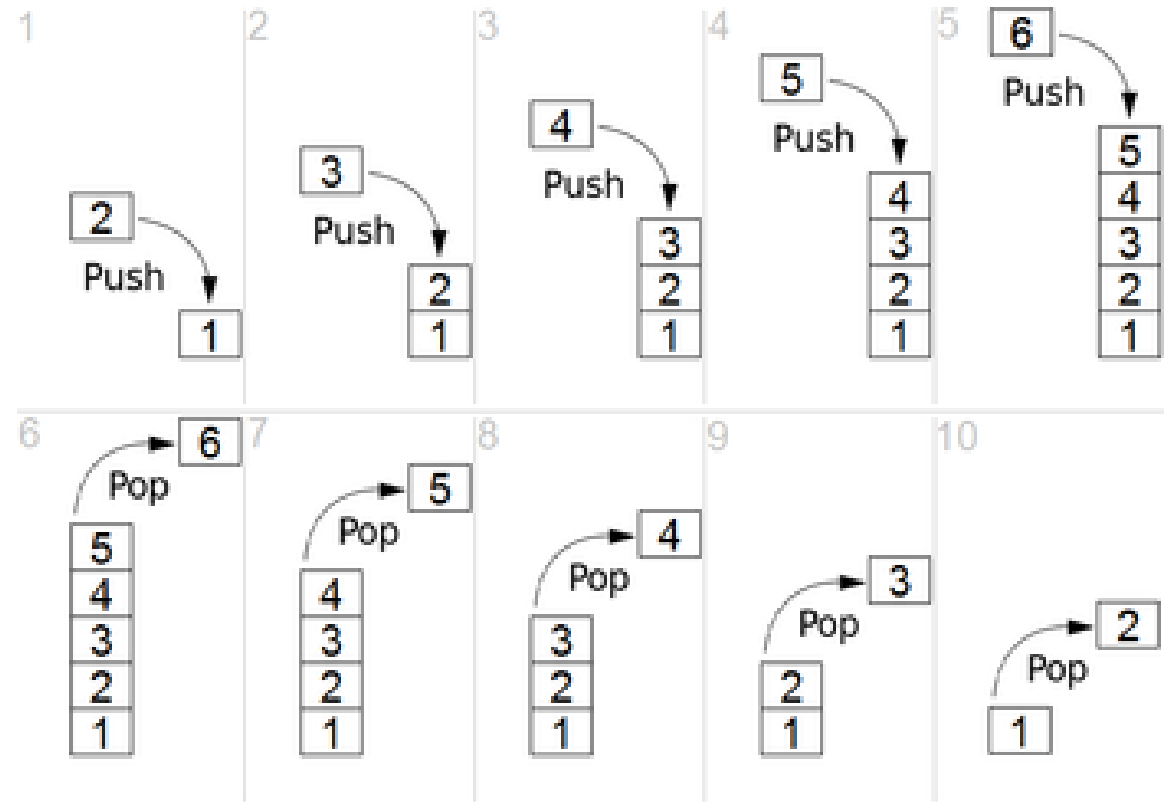
- 데이터를 넣는 작업
- 삽입 시 마지막 데이터 위치: Top

## 삭제(Pop)

- 데이터를 꺼내는 작업
- 삭제 시 마지막 데이터 위치: Top - 1

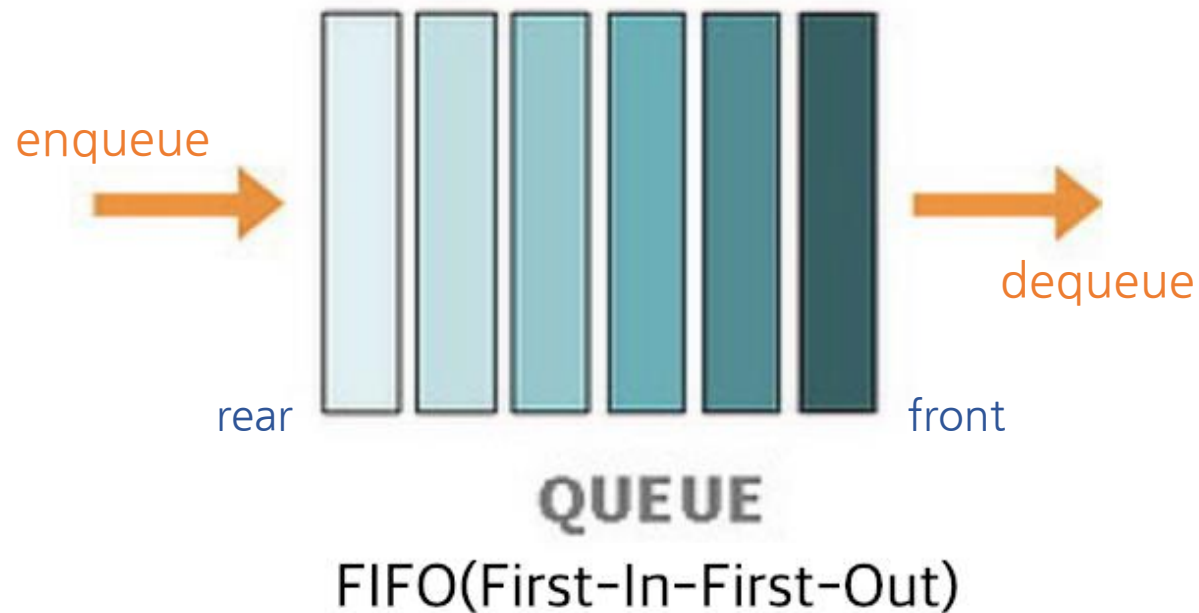
## 읽기(Peek)

마지막 위치(Top)에 해당하는 데이터를 읽음



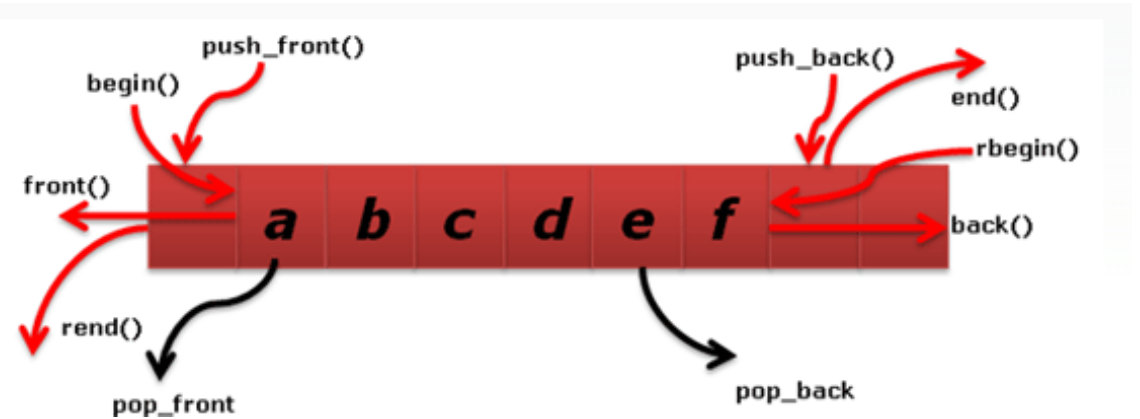
# 큐(Queue)

- 데이터를 일시적으로 저장하기 위한 자료구조
- FIFO 구조 : 가장 먼저 넣은 데이터를 가장 먼저 꺼내는 선입선출 방식
- 인큐(enqueue) : 큐에 데이터를 넣는 작업
- 디큐(dequeue): 데이터를 꺼내는 작업
- 프런트(front): 데이터를 꺼내는 쪽
- 리어(rear): 데이터를 넣는 쪽



# 데크(Deque: Double Ended Queue)

- 큐 두 개 중 하나를 좌우로 뒤집어서 붙인 구조
- 큐의 양쪽 끝에서 삽입, 삭제 연산을 수행할 수 있도록 확장한 자료구조
- 스크롤(scroll): 입력이 한쪽 끝으로만 가능하도록 제한 (입력제한)
- 셸프(shelf): 출력이 한쪽 끝으로만 가능하도록 제한 (출력제한)



# Scroll, Shelf 구조

---

- Scroll



- Shelf

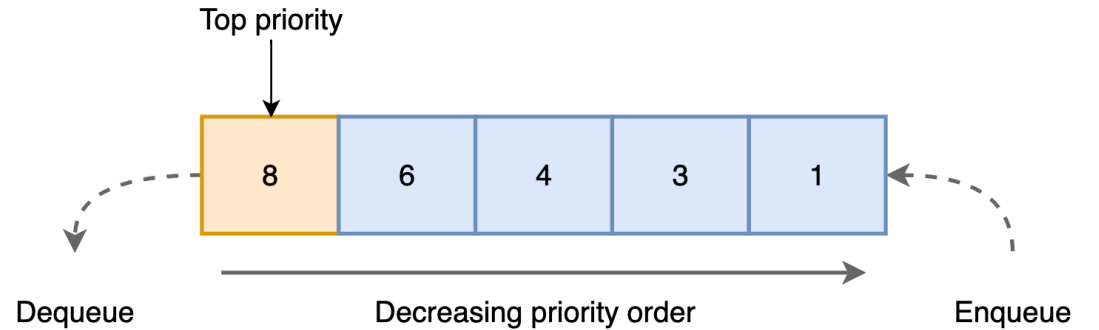


# 우선순위 큐 (Priority Queue)

- 들어간 순서에 상관없이 우선순위가 높은 데이터가 먼저 나오는 것

## <속성>

- 모든 항목에는 우선순위가 존재
- 우선순위가 높은 요소는 우선순위가 낮은 요소보다 먼저 queue에서 제외
- 두 요소의 우선순위가 같으면 queue의 순서에 따라 제공



# example

$4 \rightarrow 8 \rightarrow 2$

순서로 데이터가 들어갔을 때 처리순서  
(높은 값이 높은 우선순위를 갖는다고 가정)

Input:  $4 \rightarrow 8 \rightarrow 2$

Queue:  $4 \rightarrow 8 \rightarrow 2$

Priority queue:  $8 \rightarrow 4 \rightarrow 2$



# 풀어올 문제

- <https://programmers.co.kr/learn/courses/30/lessons/42586>

코딩테스트 연습 > 스택/큐 > 기능개발

도움말 컴파일 옵션

기능개발

문제 설명

프로그래머스 팀에서는 기능 개선 작업을 수행 중입니다. 각 기능은 진도가 100%일 때 서비스에 반영할 수 있습니다.

또, 각 기능의 개발속도는 모두 다르기 때문에 뒤에 있는 기능이 앞에 있는 기능보다 먼저 개발될 수 있고, 이때 뒤에 있는 기능은 앞에 있는 기능이 배포될 때 함께 배포됩니다.

먼저 배포되어야 하는 순서대로 작업의 진도가 적힌 정수 배열 progresses와 각 작업의 개발 속도가 적힌 정수 배열 speeds가 주어질 때 각 배포마다 몇 개의 기능이 배포되는지를 return 하도록 solution 함수를 완성하세요.

제한 사항

- 작업의 개수(progresses, speeds배열의 길이)는 100개 이하입니다.
- 작업 진도는 100 미만의 자연수입니다.
- 작업 속도는 100 이하의 자연수입니다.
- 배포는 하루에 한 번만 할 수 있으며, 하루의 끝에 이루어진다고 가정합니다. 예를 들어 진도율이 95%인 작업의 개발 속도가 하루에 4%라면 배포는 2일 뒤에 이루어집니다.

입출력 예

progresses	speeds	return
[93, 30, 55]	[1, 30, 5]	[2, 1]
[95, 90, 99, 99, 80, 99]	[1, 1, 1, 1, 1, 1]	[1, 3, 2]

입출력 예 설명

입출력 예 #1

첫 번째 기능은 93% 완료되어 있고 하루에 1%씩 작업이 가능하므로 7일간 작업 후 배포가 가능합니다.

두 번째 기능은 30%가 완료되어 있고 하루에 30%씩 작업이 가능하므로 3일간 작업 후 배포가 가능합니다.

질문하기 (104) 테스트 케이스 추가하기

Solution.java

```
1 class Solution {
2     public int[] solution(int[] progresses, int[] speeds) {
3         int[] answer = {};
4         return answer;
5     }
6 }
```

실행 결과

실행 결과가 여기에 표시됩니다.

다른 사람의 풀이 초기화 코드 실행 제출 후 재점하기