

※ 공통점 : 배열 인덱스로 특정 위치의 자료에 접근가능! only 앞 or 뒤에서만!

Stack (스택)

- LIFO (Last In First Out)
- 시간복잡도 : $O(1)$ \hookrightarrow 맨끝에서만 연산이 일어나므로
- 연산 : push & pop, top, empty, size
- empty일때 $top = -1$, 자료가 매일때 $top = 0$, 자료가 2개일때 $top = 1 \dots$
 \hookrightarrow top : 배열의 index 처럼 생각!
- `std::stack` \hookrightarrow `#include <stack>`
 \hookrightarrow `stack<자료형> 이름;`
 \hookrightarrow 함수사용시 '이름.pop()' 과 같이 사용하면 된다.

Queue (큐)

- FIFO (First In First Out)
- 시간복잡도 : $O(1)$ \hookrightarrow 왼쪽 끝에서 삭제, 오른쪽 끝에서 삽입 (끝에서 연산)하므로
- 연산 : push & pop, front & back (중복), empty, size
- 배열로 원형큐를 구현한다고 했을 때, 배열 0번은 비워둬야 한다.
front는 맨 처음 데이터 위치의 바로 앞을 가리키며, rear은 맨 마지막 데이터 위치를 가리킨다.
- `std::queue` \hookrightarrow `#include <queue>`
 \hookrightarrow `queue<자료형> 이름;`
 \hookrightarrow 함수 사용시 '이름.front()' 와 같이 사용하면 된다.

Deque (덱)

- Stack + Queue : 자료의 양끝에서 연산이 이루어짐
- 시간복잡도 : $O(1)$ \Rightarrow 양 끝에서 연산이 이루어지므로
- 연산 : push-front, push-back & pop-front, pop-back, front, back, empty, size
- `<std::deque>` \Rightarrow `#include <deque>`
 - \Rightarrow `deque<자료형>` 이름;
 - \Rightarrow 함수 사용시 '이름.front()' 와 같이 사용하면 된다.