

스택

STACK

# 01

## 스택이란?

스택이란?  
스택의 특징

# 02

## 2.스택 구현 방법

스택의 연산

# 03

## 3.스택 문제

[https://www.acmicpc.net/  
problem/10828](https://www.acmicpc.net/problem/10828)

# 01

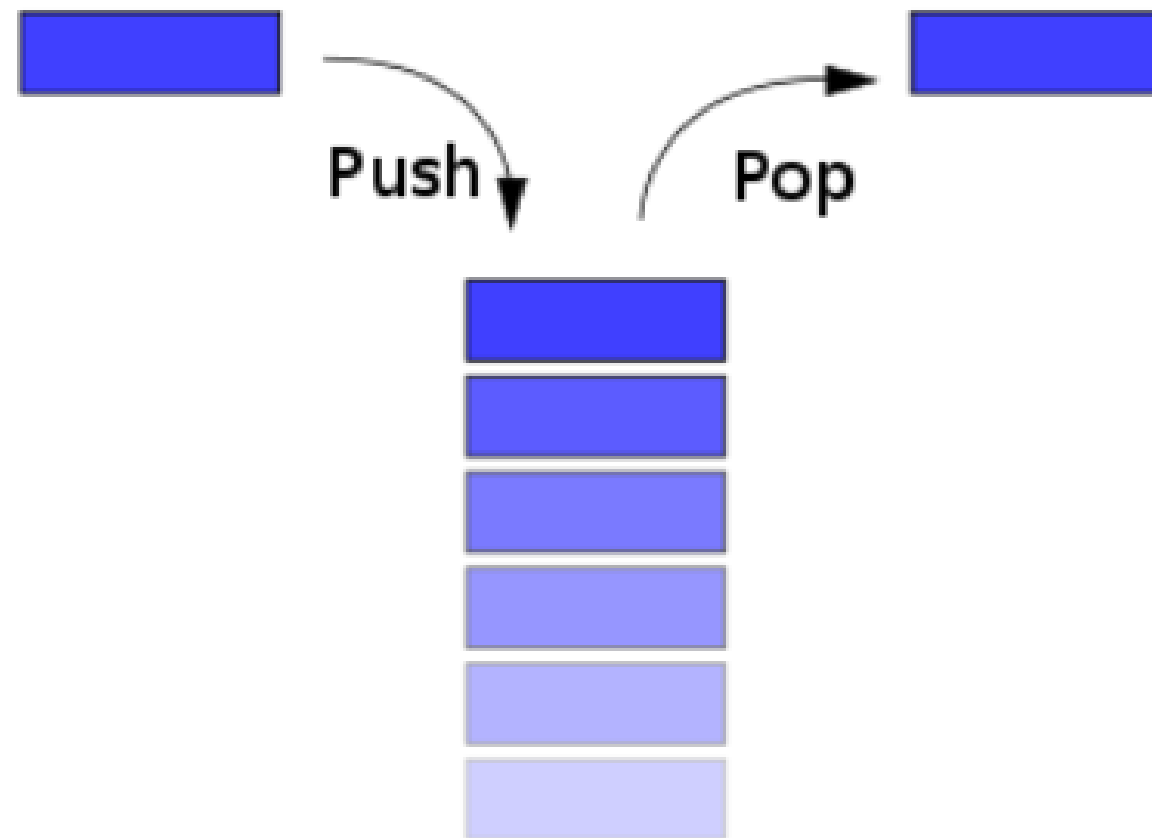
스택이란?

스택이란?  
스택의 특징

---

# 스택이란?

스택이란 모든 원소의 삽입과 삭제가 한곳에서 (Top)에서만 이루어지도록 제한되어 있는 리스트이다!



스택은 실행이 되는 특정한 순서를 따르는 선형적 데이터 구조입니다.

한 쪽 끝에서만 자료를 넣고 뺄 수 있는 LIFO(Last In First Out) 형식의 자료 구조

# 스택의 특징

정해진 방향으로만 쌓을 수 있으며 TOP으로 정한 곳을 통해서만 접근 가능

구조가 단순해서 구현이 쉽다.  
데이터 저장/읽기 속도가 빠르다

데이터 최대 갯수를 미리 정해야 한다.  
-> 파이썬의 경우 재귀 함수는 1000번까지만 호출이 가능함  
저장 공간의 낭비가 발생할 수 있다.  
-> 미리 최대 갯수만큼 저장 공간 확보해야 함

웹 브라우저의 방문 기록  
실행 취소(undo)  
역순 문자열 만들기  
후위 표기법 계산

# 스택의 특징

시간 복잡도 (Time complexity)

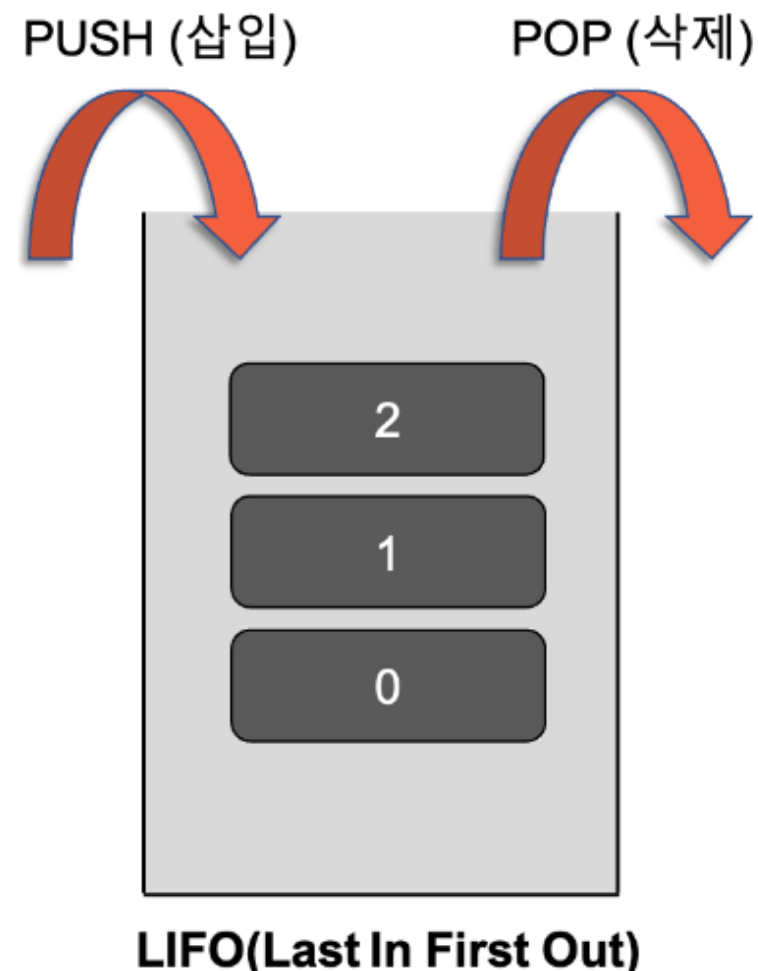
Operation	Average	Worst
Access	$O(n)$	$O(n)$
Search	$O(n)$	$O(n)$
Insert (push)	$O(1)$	$O(1)$
Delete (pop)	$O(1)$	$O(1)$

# 02

## 스택 구현 방법

스택의 연산

# 스택의 연산



## 스택 생성/삽입

```
#include <stack>
stack<int>s1; //int형 스택 선언
stack<char>s2; //char형 스택 선언
s1.push(1); // stack s1에 값 1 추가
s1.push(2); // stack s1에 값 2 추가
Stack<Integer> stack = new Stack<>(); //int형 스택 선언
stack.push(1); // stack에 값 1 추가
stack.push(2); // stack에 값 2 추가
```

## 스택 삭제

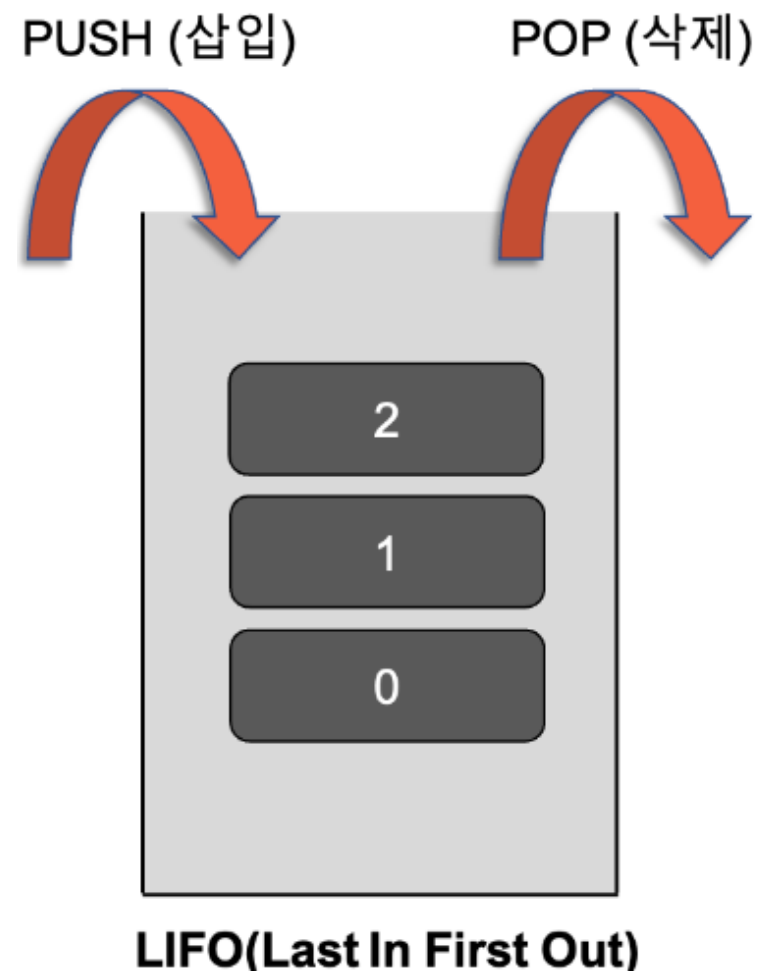
```
s1.pop(); //스택의 최상단 값을 삭제
s1.clear(); //stack의 전체 값 제거 (초기화)
stack.pop(); // stack에 값 제거
stack.clear(); // stack의 전체 값 제거 (초기화)
```

## 스택의 최상단 값

```
s1.top() //현재 스택에 가장 위에 있는 값을 반환
stack.peek(); // stack의 가장 상단의 값 출력
```



## 스택의 연산



### 기타 메소드

스택이 비어있는지 검사

`s1.empty();` // stack s1이 비어있는지 check (비어있다면 true)

스택의 스왑

`s1.swap(s2);` // stack s2와 요소 바꾸기

`Stack<Integer> stack = new Stack<>();` //int형 스택 선언

`stack.push(1);` // stack에 값 1 추가

`stack.push(2);` // stack에 값 2 추가

`stack.size();` // stack의 크기 출력 : 2

`stack.empty();` // stack이 비어있는지 check (비어있다면 true)

`stack.contains(1)` // stack에 1이 있는지 check (있다면 true)

# 03

## 스택 문제

# 스택 문제

https://www.acmicpc.net/problem/10828

4 10828번

제출

맞힌 사람

숏코딩

재채점 결과

채점 현황

내 제출

난이도 기여

강의

질문 검색

스택

성공

4 실버 IV

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
0.5 초 (추가 시간 없음)	256 MB	157850	60469	43556	38.085%

문제