1주차 3차시 기본자료 구조(선형구조)

【학습목표】

1. 배열과 리스트를 이해할 수 있다.
2. 스택과 큐의 개념을 이해할 수 있다.

학습내용1 : 배열과 리스트

1. 배열(Array)

배열(Array)은 동일한 자료형의 데이터를 여러 개를 저장하기 위해 한 번에 많은 기억공간을 만들어 놓고 사용하는 가장 기본적인 자료구조이다. 행렬에 관한문제나 탐색, 정열(SORT)등에 사용되며 대부분 많은 변수를 선언해야 하는 프로그램에 사용된다. 배열은 차례대로 접근하는 순차처리 사용되며 반면 새로운 데이터를 삽입하거나 삭제하는 작업에는 경우에 따라서 전체의 데이터를 움직이어야 하기 때문에 많은 시간이 소요된다.
이러한 배열의 단점을 보완한 자료구조가 연결리스트(Linked List)이다.

A(1)	A(2)	A(3)	A(4)	A(5)	A(6)	A(7)	A(8)	A(9)	A(10)	A(11)
10	20	30	40	50	60	70	80	90	100	

Ⓐ 11개의 원소를 갖는 배열

A(1)	A(2)	A(3)	A(4)	A(5)	A(6)	A(7)	A(8)	A(9)	A(10)	A(11)
10	15	20	30	40	50	60	70	80	90	100

Ⓑ A[2]에 15의 데이터를 삽입

A(1)	A(2)	A(3)	A(4)	A(5)	A(6)	A(7)	A(8)	A(9)	A(10)	A(11)
15	20	30	40	50	60	70	80	90	100	

Ⓒ A[1]의 데이터 10이 삭제

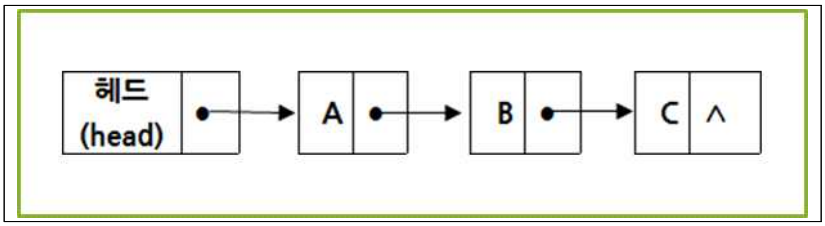
배열의 초기 형태와 삽입과 삭제후의 형태

2. 연결 리스트 (Linked List)

데이터들을 임의의 공간에 기억시키고 자료의 항목에 따라 노드의 포인터 부분 포인터를 사용하여 서로 연결시킨 자료구조 이다. 연결 리스트에서 원소가 저장되어 있는 것을 노드(node)라 하며, 노드는 데이터 부분(data field)과 링크 부분(link field)으로 구성되어 있다. 데이터 부분은 원소의 실제 값이 들어 있으며, 링크 부분은 다음 원소가 저장되어 있는 주소를 가리키는 포인터(pointer)이다. 가장 단순한 단순 연결 리스트와 이중 연결 리스트, 단순 원형 연결 리스트, 이중 원형 연결 리스트 등이 있다.

DATA	LINK
데이터	링크 부분

* 연결 리스트의 구조



* 장점
Ⓐ 자료의 삽입 및 삭제가 용의 하다.
Ⓑ 비연속적 (한 리스트를 여러 개의 리스트로 분리하기가 쉽다.)

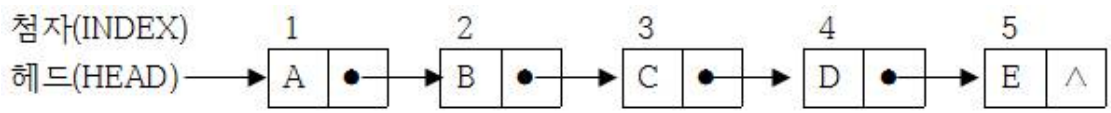
* 단점
Ⓐ Access Time이 느리다.
Ⓑ 링크 부분만큼 포인터를 위한 추가 공간이 필요하다.
Ⓒ 중간에 노드의 연결이 끊어지면 그 다음 노드를 찾기가 어렵다.

3. 단순 연결 리스트(singly linked list)

단순 연결 리스트는 하나의 노드를 데이터 부분과 링크부분으로 구성하여 링크 부분에 다음노드를 가르키는 포인터를 저장하고, 마지막 노드의 링크 부분에는 널 링크(∧)를 저장한다. 헤드 노드는 연결 리스트의 첫 번째 원소를 가리키는 주소를 가지고 있다.

단순 연결 리스트에서 각 원소의 순서는 원소의 링크 부분에 의해 정해지며, 링크의 값은 다음 원소를 가리키는 포인터로서 연결되어 있는 다음 원소의 주소를 나타낸다.

1) 단순 연결 리스트 구조



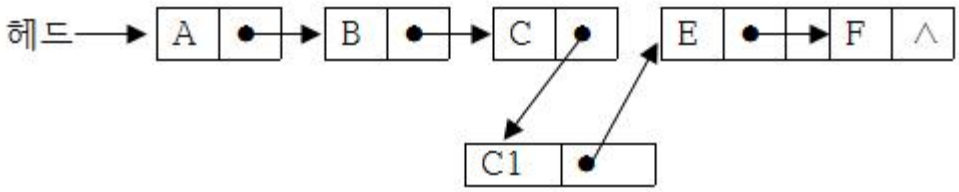
(단순 연결 리스트 구조 예)

첨자	자료	링크
1	A	2
2	B	3
3	C	4
4	D	5
5	E	∧

(단순 연결 리스트의 배열 표현)

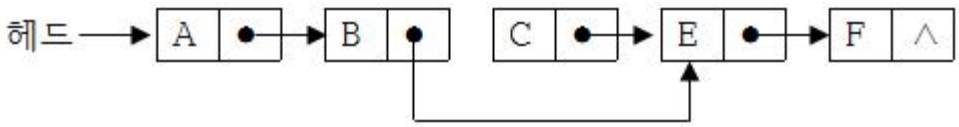
2) 단순 연결 리스트 삽입

단순 연결 리스트에 새로운 노드를 삽입하는 알고리즘은 다음과 같다.



(단순 연결 리스트 C1 삽입 예)

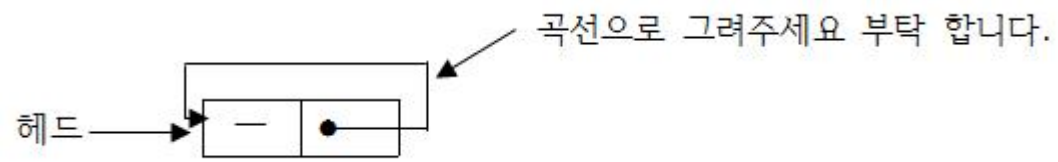
3) 단순 연결 리스트 삭제



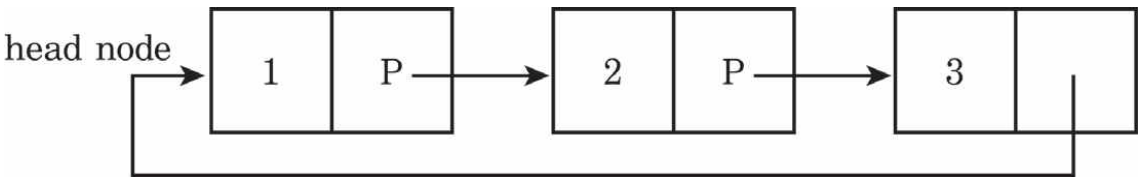
(단순 연결 리스트 C 삭제 예)

4. 원형 연결 리스트(Circular linked list)

단순 연결 리스트에서는 리스트의 마지막 노드의 링크 값이 널(∧)이었지만 원형 연결 리스트는 마지막 노드의 링크 값이 널(∧)이 아닌 리스트의 첫 번째 주소를 가리키도록 하는 구조로 되어 있다.



0개의 노드로 구성된 원형 연결 리스트(공백 리스트)의 구조



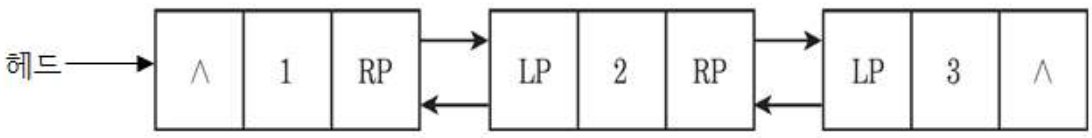
원형 연결 리스트(circular linked list 예)

5. 이중 연결 리스트(Doubly linked list)

단순 연결 리스트(singly linked list)와 원형 연결 리스트(Circular linked list)는 각 노드에 다음노드를 가리키는 한 개의 링크만을 사용하기 때문에 특정한 노드를 검색하거나, 삽입 또는 삭제를 할 경우 한 쪽 방향으로만 해당 노드의 위치를 찾을 수 있지만 이중 연결 리스트는 양쪽방향으로 해당 노드를 검색하면 쉽게 찾을 수 있다.
이중 연결 리스트는 노드의 선행 노드를 가리키는 좌링크(LLINK), 자료(DATA), 후행노드를 가리키는 우링크(RLINK)의 세 개의 영역으로 각 노드를 구분하여 양방향으로 특정 노드를 검색할 수 있는 구조이다.

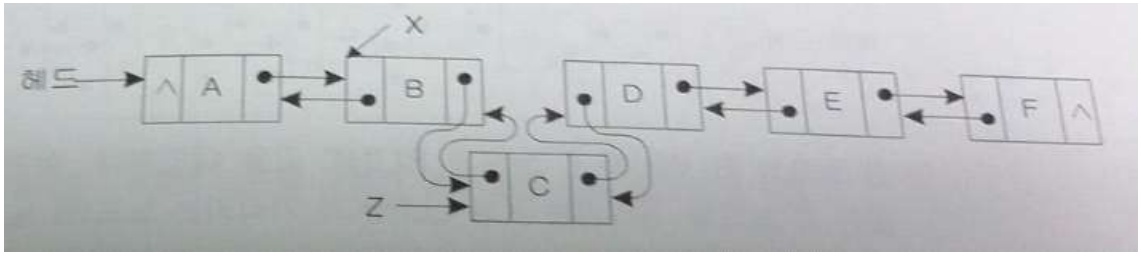
LLINK	DATA	RLINK

LLINK : 선행 노드의 포인터
DATA : 자료
RLINK : 후행 노드의 포인터

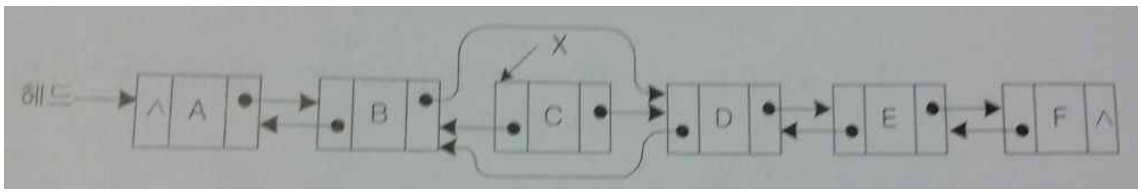


(이중 연결 리스트의 구조)

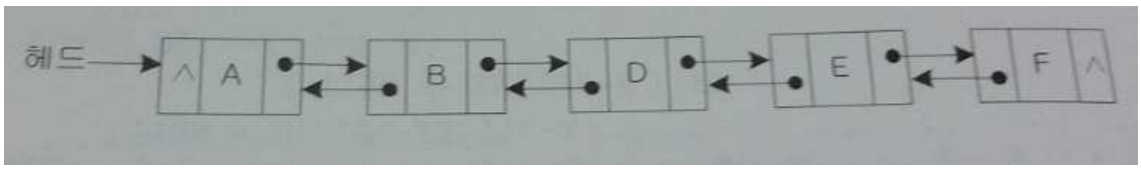
1) 이중 연결 리스트(Doubly linked list)삽입
이중 연결 리스트를 구성하는 임의의 노드 X의 오른쪽에 노드 Z를 삽입하는 알고리즘은 아래와 같다.



2) 이중 연결 리스트(Doubly linked list)삭제
이중 연결 리스트를 구성하는 임의의 노드 X를 삭제하는 알고리즘은 아래와 같다.

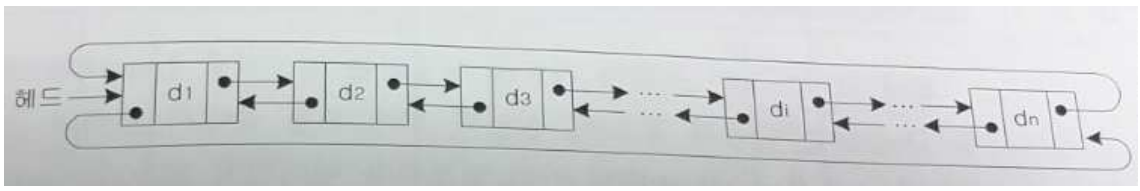


X가 가리키고 있는 C 노드를 가용 기억공간으로 되돌린다. 즉 C 노드는 이중 연결 리스트에서 삭제 되었다.



6. 이중 원형 연결 리스트(Doubly circular linked list)

이중 원형 연결 리스트는 이중 연결 리스트와 원형 연결 리스트를 혼합한 형태이다. 즉, 각 노드가 선행노드와 후속노드를 가리키기 위한 두 개의 포인터를 가지고 있으며, 마지막 노드의 RLINK는 처음 노드의 위치를 가리킨다. 또한 처음 노드의 LLINK는 마지막 노드의 위치를 가리키는 형태를 취한다. 이중 원형 연결 리스트는 어떠한 노드도 널 링크를 갖지 않는다.

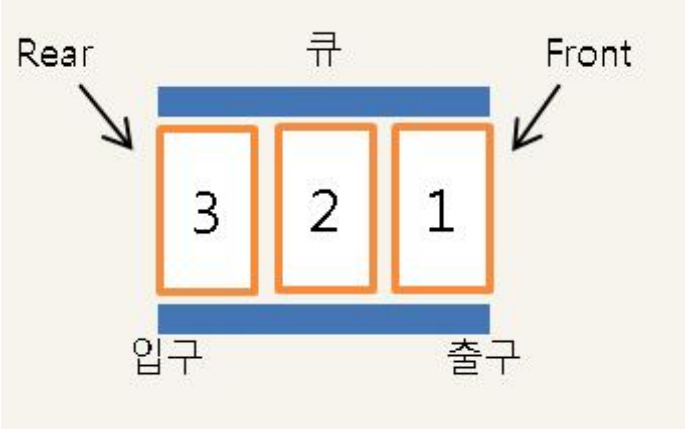


(이중 원형 연결 리스트의 구조)

학습내용2 : 큐와 스택

1. 큐(QUEUE)

큐는 주기억장치에서 연속적인 공간을 배정하여 데이터를 Rear라는 포인터가 가르치는 방향에서 데이터가 삽입되며 Front 라는 포인터가 가리키는 방향에서 삭제 되도록 하는 알고리즘으로 구성되어 있는 순서리스트와 같은 자료 구조이다. 큐의 응용분야로는 운영체제의 작업 스케줄링 등에 사용 된다.

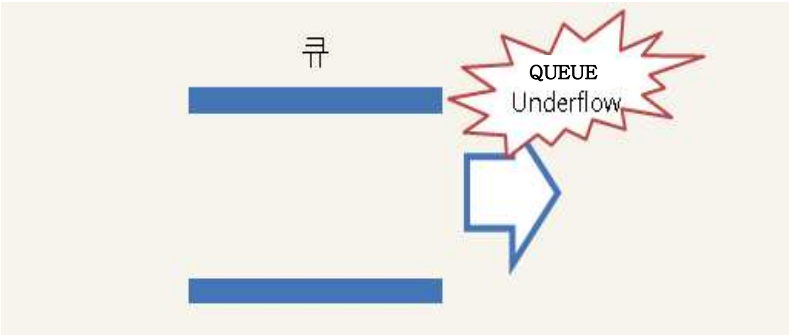


QUEUE

* 큐(QUEUE) 알고리즘

Ⓐ FIFO(First in First Out) 선입선출 이며 FCFS(First Come First Service)라고도 한다.

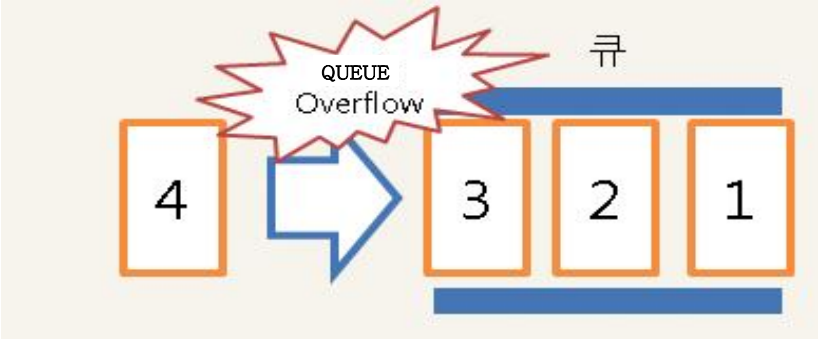
Ⓑ 아무것도 없는 큐에서 데이터를 삭제할 경우 아래 그림 같이 Under Flow Error가 발생 한다.



Under Flow Error

© 이미 꽉 찬 큐에 데이터를 삽입하려는 경우 아래 그림 같이 Over Flow Error가 발생 한다.

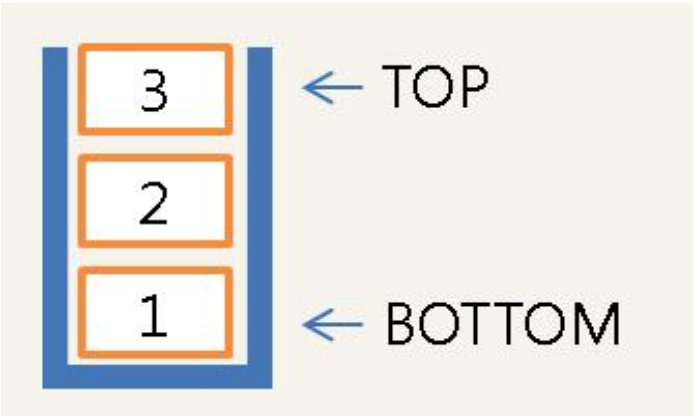
Over Flow Error



2. 스택(Stack)

스택이란 주기억 장치에서 연속적인 공간을 배정하여 데이터를 1,2,3 같이 차례대로 삽입 연산(PUSH)이나 삭제연산(POP)을 실행 할 때 Top 포인터를 이용하여 순서 리스트의 한 쪽 끝에서만 실행되는 알고리즘으로 구성 되어있는 자료구조이다.

- 자료의 삽입 : Top=Top + 1
- 자료의 삭제 : Top=Top - 1
- 스택의 크기를 S 일 때 Top > S 이면 Overflow Error 가 발생 한다.
- FILO(First in Last Out)선입후출, LIFO(Last in First Out)후입선출의 구조를 가진다.



STACK

Ⓐ 삽입알고리즘

```
Top = Top +1
if(Top > S) Then
        Stack_overflow
Else
        Stack(top) ← data
```

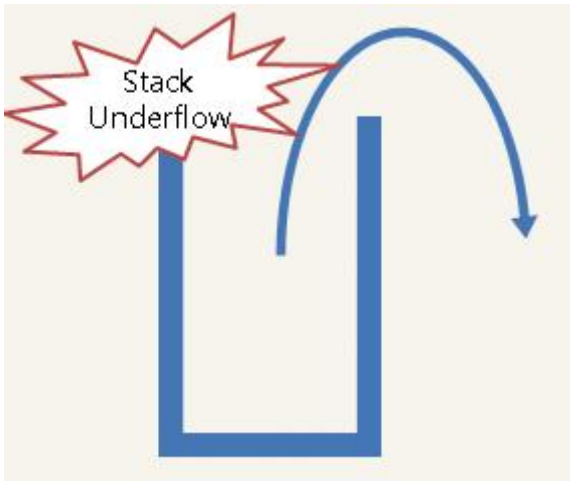
스택 삽입알고리즘

Ⓑ 삭제 알고리즘

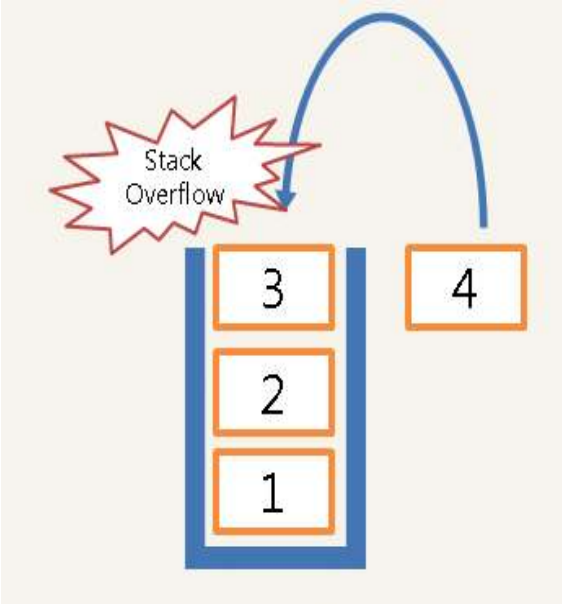
```
if(Top = 0) Then
        Stack_underflow
Else
        Stack(top) ← data
        Top = Top -1
```

스택 삭제 알고리즘

Ⓒ 스택의 용도
- 인터럽트의 처리
- 수식의 계산(산술식 표현)
- 서브루틴의 복귀번지 저장



underflow error
아무것도 없는 Stack에서 Pop()으로 데이터를 꺼내려고 하면 underflow error 발생한다.



Overflow error

이미 꽉 찬 Stack에 Push()로 데이터를 삽입하려는 경우 Overflow error 발생한다.

【학습정리】

1. 배열과 리스트

배열(Array)은 동일한 자료형의 데이터를 여러 개를 저장하기 위해 한 번에 많은 기억공간을 만들어 놓고 사용하는 가장 기본적인 자료구조이다. 행렬에 관한문제나 탐색, 정열(SORT)등에 사용되며 대부분 많은 변수를 선언해야 하는 프로그램에 사용된다. 배열은 차례대로 접근하는 순차처리 사용되며 반면 새로운 데이터를 삽입하거나 삭제하는 작업에는 경우에 따라서 전체의 데이터를 움직이어야 하기 때문에 많은 시간이 소요된다.
이러한 배열의 단점을 보완한 자료구조가 연결리스트(Linked List)이다.

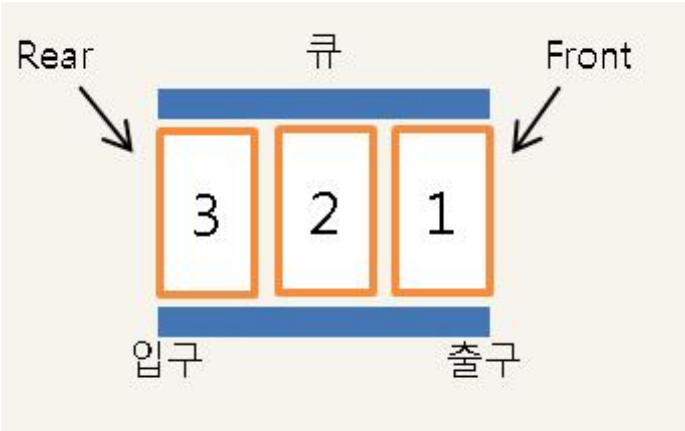
2. 연결 리스트 (Linked List)

데이터들을 임의의 공간에 기억시키고 자료의 항목에 따라 노드의 포인터 부분 포인터를 사용하여 서로 연결시킨 자료구조 이다. 연결 리스트에서 원소가 저장되어 있는 것을 노드(node)라 하며, 노드는 데이터 부분(data field)과 링크 부분(link field)으로 구성되어 있다. 데이터 부분은 원소의 실제 값이 들어 있으며, 링크 부분은 다음 원소가 저장되어 있는 주소를 가리키는 포인터(pointer)이다. 가장 단순한 단순 연결 리스트와 이중 연결 리스트, 단순 원형 연결 리스트, 이중 원형 연결 리스트 등이 있다.

3. 큐와 스택

1) 큐(QUEUE)

큐는 주기억장치에서 연속적인 공간을 배정하여 데이터를 Rear라는 포인터가 가르치는 방향에서 데이터가 삽입되며 Front라는 포인터가 가리키는 방향에서 삭제 되도록 하는 알고리즘으로 구성되어 있는 순서리스트와 같은 자료 구조이다. 큐의 응용분야로는 운영체제의 작업 스케줄링 등에 사용 된다.



큐(QUEUE) 알고리즘

FIFO(First in First Out) 선입선출 이며 FCFS(First Come First Service)라고도 한다.

스택(Stack)
스택이란 주기억 장치에서 연속적인 공간을 배정하여 데이터를 1,2,3 같이 차례대로 삽입 연산(PUSH)이나 삭제연산(POP)을 실행 할 때 Top 포인터를 이용하여 순서 리스트의 한 쪽 끝에서만 실행되는 알고리즘으로 구성되어있는 자료구조이다.
- 자료의 삽입 : Top=Top + 1
- 자료의 삭제 : Top=Top - 1
- 스택의 크기를 S 일 때 Top > S 이면 Overflow Error 가 발생 한다.
- FILO(First in Last Out)선입후출, LIFO(Last in First Out)후입선출의 구조를 가진다.