

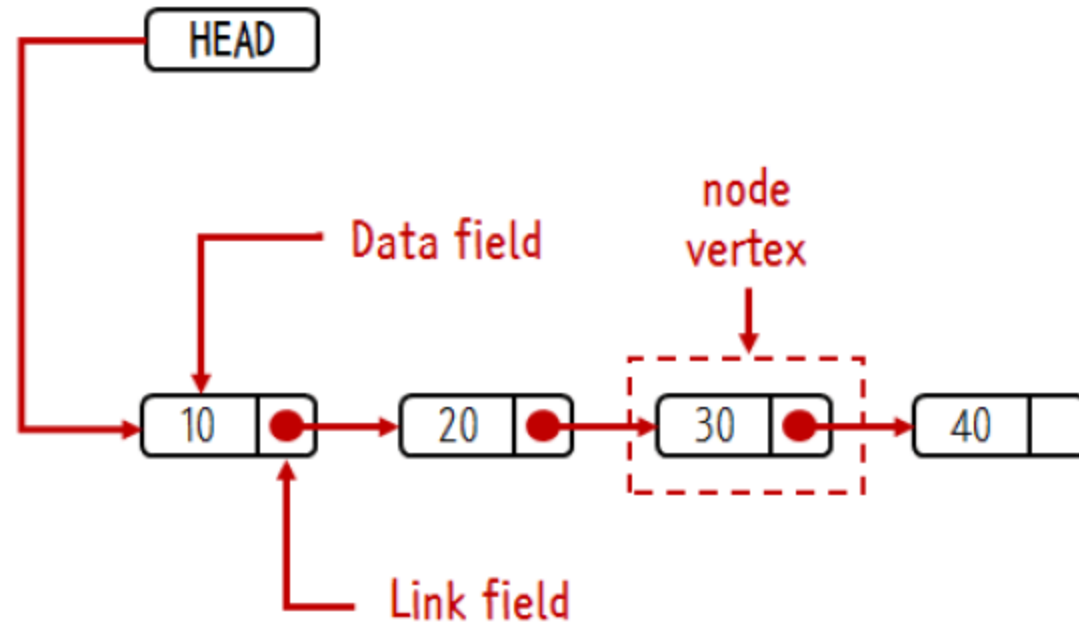
알고리즘

정다현

목 차

- 연결리스트 소개
- 배열과 연결리스트 차이
- 스택 소개
- 연결리스트를 이용한 스택의 구현

연결리스트(LinkedList)란?



- 노드와 노드간의 연결을 이용해서 리스트를 구현한 것
- 노드 : 데이터와 다음 노드의 위치를 저장

연결리스트 구현

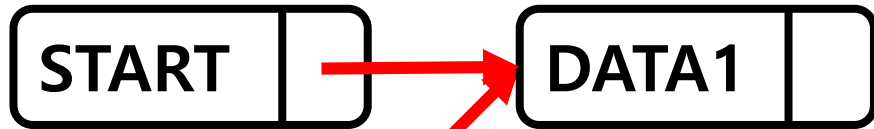
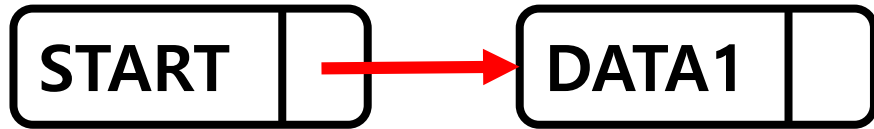
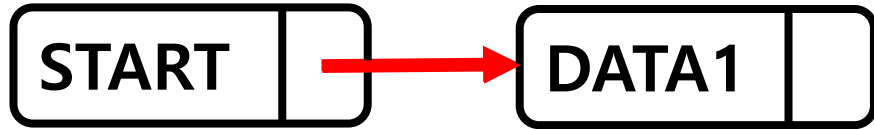


1. 노드 생성



2. 연결

연결리스트 - 노드 추가

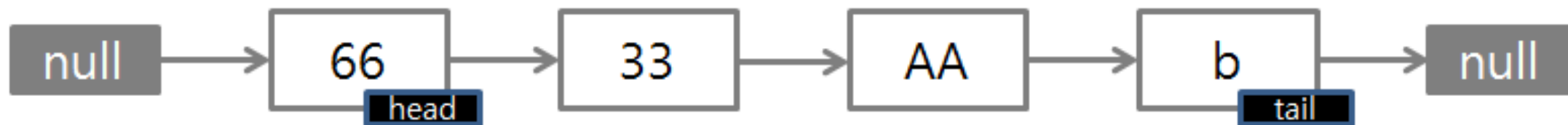


연결리스트 - 노드 삭제

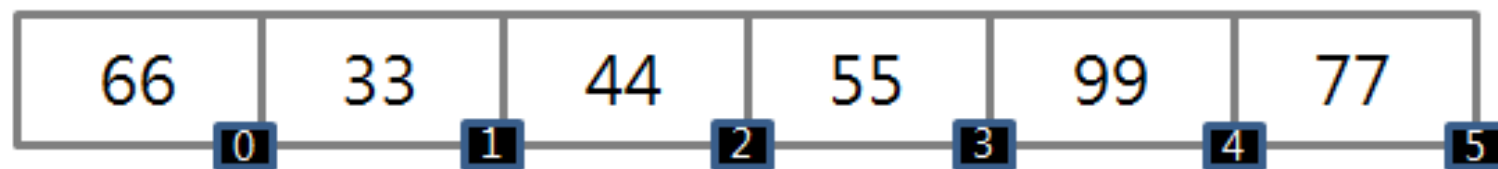


배열과 연결리스트 소개

Linked List



Array List



배열과 연결리스트 차이

연결리스트의 장단점

장점 : 빈번하게 원소의 추가나 삭제를 반복하는 경우
리스트의 크기에 제한이 없다

단점 : 인덱스를 통하여 탐색할 때 시간이 좀 더 소요

ArrayList

```
public static void main(String[] args) {  
    ArrayList<Object> numbers = new ArrayList<>();  
    for(long i = 0 ; i < 10000000 ; i++) {  
        numbers.add(i);  
    }  
    long start = System.currentTimeMillis();  
    numbers.get(8888888);  
    long end = System.currentTimeMillis();  
    System.out.println( "실행 시간 : " + ( end - start )/1000.0 + "초");  
}
```

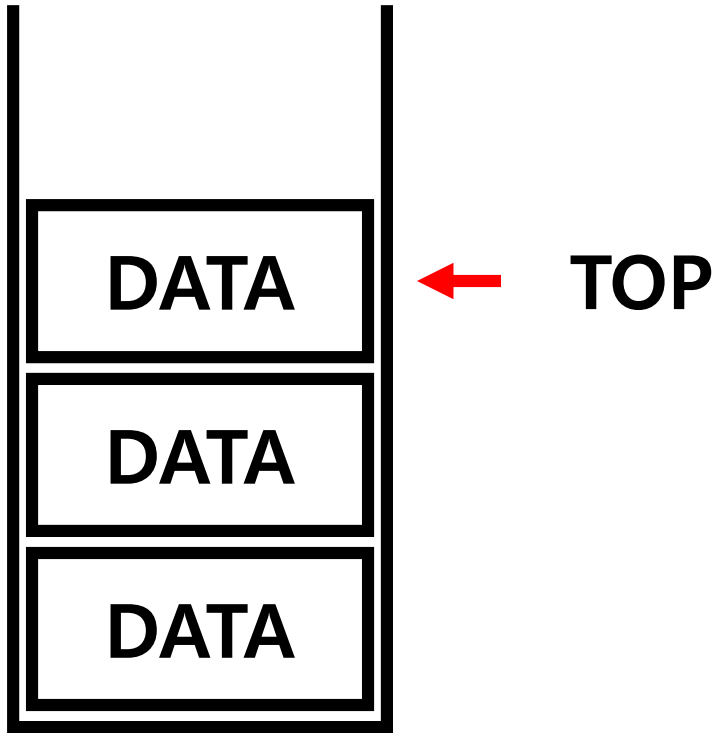
실행 시간 : 0.0초

LinkedList

```
public static void main(String[] args) {  
    ArrayList<Object> numbers = new ArrayList<>();  
    for(long i = 0 ; i < 10000000 ; i++) {  
        numbers.add(i);  
    }  
    long start = System.currentTimeMillis();  
    numbers.get(8888888);  
    long end = System.currentTimeMillis();  
    System.out.println( "실행 시간 : " + ( end - start )/1000.0 + "초");  
}
```

실행 시간 : 0.042초

스택(STACK)이란?



- STACK : '쌓이다', '더미'
- LIFO(LAST IN FIRST OUT)
- push : 스택에 새 데이터 추가
- pop : TOP 삭제

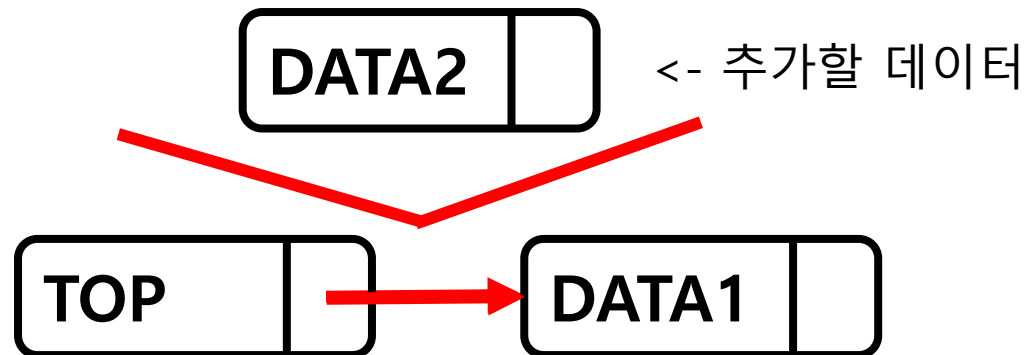
연결리스트를 이용한 스택의 구현

배열 : 크기가 정해져 있음

-> 크기 이상의 데이터가 push 되면 오버플로우 발생

연결리스트 : 동적으로 할당 가능

-> 메모리가 고갈될 때까지 push 가능



끝!