

알고리즘3

본 문서는 학습 참고용으로만 사용하시고 웹이나 타인에게 배포를
금합니다

선형검색

이진검색

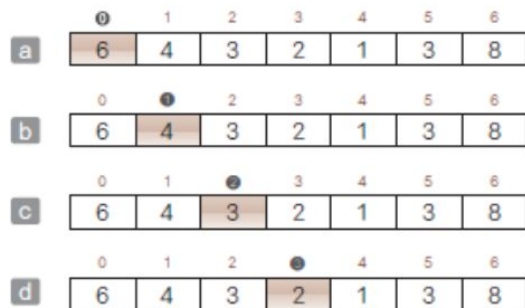
Stack구현

Queue구현

ArrayList구현

LinkedList구현

선형검색-Linear Search

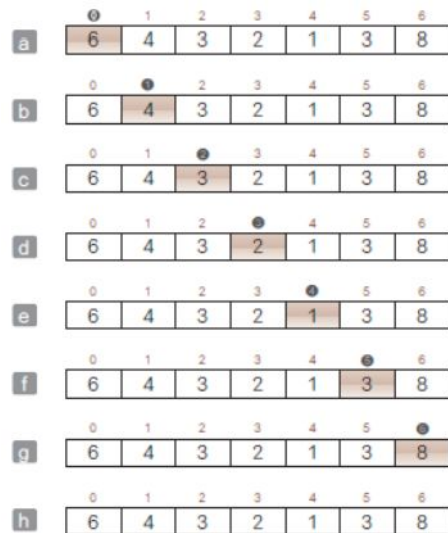


검색 성공!
검색할 값과 같은 요소를 발견

[그림 3-2] 선형 검색의 예(2를 검색 : 검색 성공)

선형검색 성공시

배열의 요소를 맨 앞부터
순서대로 검색합니다.



검색 실패!
배열의 끝을 지나갔습니다.

[그림 3-3] 선형 검색의 예(5를 검색 : 검색 실패)

배열의 요소를 맨 앞부터
순서대로 검색합니다.

선형검색 실패시

선형검색 개선-보초법(Sentinel) 사용

보초 법이란 반복의 종료를 알리는 특정한 값인 **보초(Sentinel) 값**을 사용하여

종료 조건중 검색 실패 조건을 제거하여 판단 횟수를 줄이는 방법이다

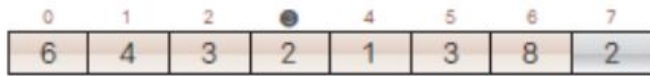
검색하고자 하는 키 값을 기존 리스트의 맨 끝 자리에 추가하고 이를 보초 값으로 사용한다

이렇게 하면 배열에 찾던 값이 없어도 보초 값까지 검색하면 "종료 조건 1(검색 실패)"을 만족하게 됩니다.

이렇게 보초는 반복문에서 종료 판단 횟수를 줄이는 역할을 하게 됩니다. $O(n)$

보초법 (Sentinel)

a 2를 검색(검색 성공)



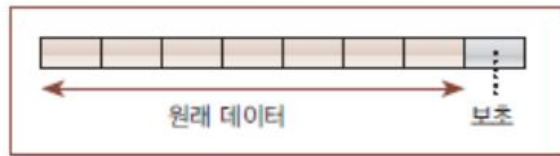
검색할 값과 같은 요소를 발견

b 5를 검색(검색 실패)



검색할 값과 같은 요소를 발견

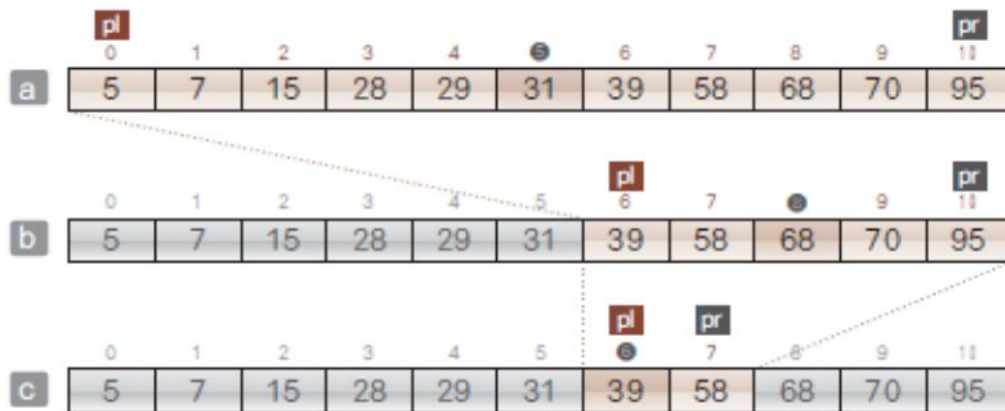
※ 단, 발견한 것은 보초



[그림 3-4] 보초법을 이용한 선형 검색

이진 검색-Binary Search

- 이진 검색은 정렬되어 있는 것을 전제로 해야 한다.
- 정렬된 요소를 입력받고 원하는 요소가 있으면 그 요소의 위치를, 아니면 null 또는 -1을 반환한다.
- 탐색 범위의 중간부터 시작하여 비교를 통해 절반의 숫자를 제거한다
- 매 단계마다 남은 범위의 가운데 수를 말하고 대답에 따라 절반 제거를 반복한다. $O(\log n)$



검색 성공!
검색할 값과 일치하는 요소를 발견

[그림 3-5] 이진 검색의 한 예(39를 검색 : 검색 성공)

좌표 정렬하기

N개의 평면상의 좌표(x, y)가 주어지면 모든 좌표를 내림차순으로 정렬하는 프로그램을 작성하세요.

[입력]

첫째 줄에 좌표의 개수인 $N(3 \leq N \leq 100,000)$ 이 주어집니다.

두 번째 줄부터 N개의 좌표가 x, y 순으로 주어집니다. x, y값은 양수만 입력됩니다.

[출력] N개의 좌표를 정렬하여 출력하세요

[예시] 오른쪽 이미지 참조

```
<terminated> SortPoint [Java Applica
5
7 3
2 5
4 8
7 9
1 5
7 9
7 3
4 8
2 5
1 5
```

← 좌표개수

← 좌표입력

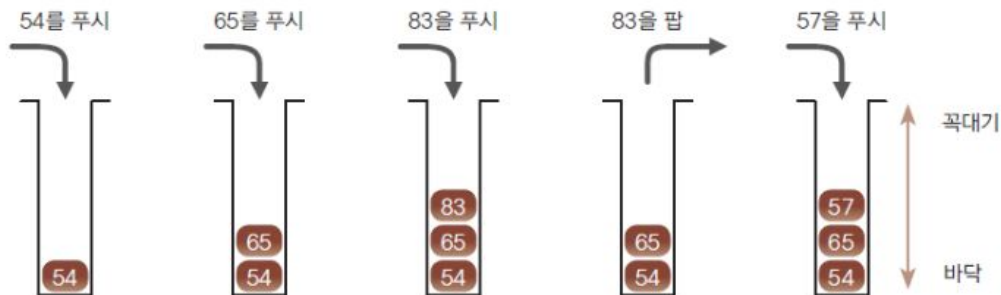
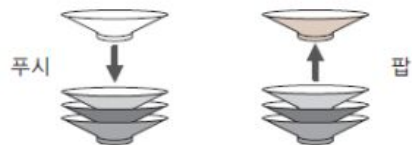
← 결과

Stack이란?

후입선출(LIFO)

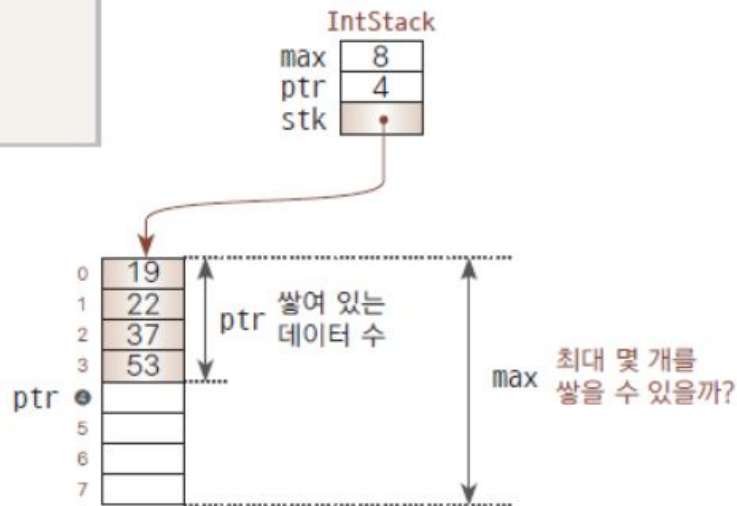
Last In First Out구조를 가진

자료구조



IntStack만들기

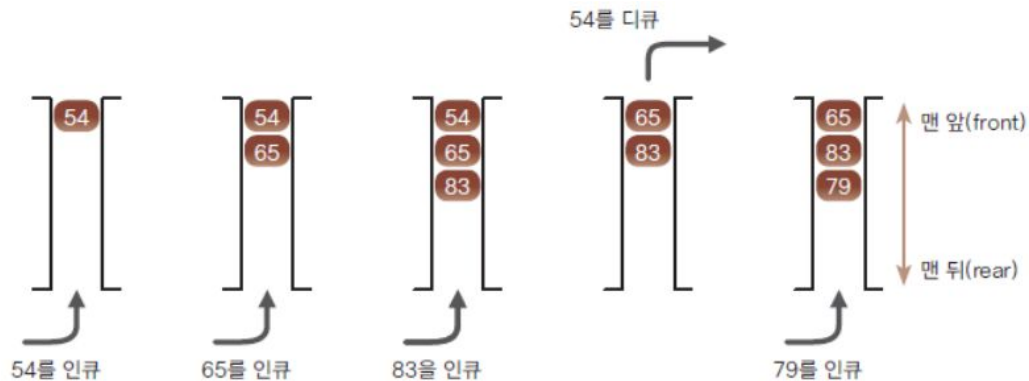
```
class IntStack {  
    int max;           // 스택 용량  
    int ptr;           // 스택 포인터  
    int[] stk;         // 스택의 본체  
}
```



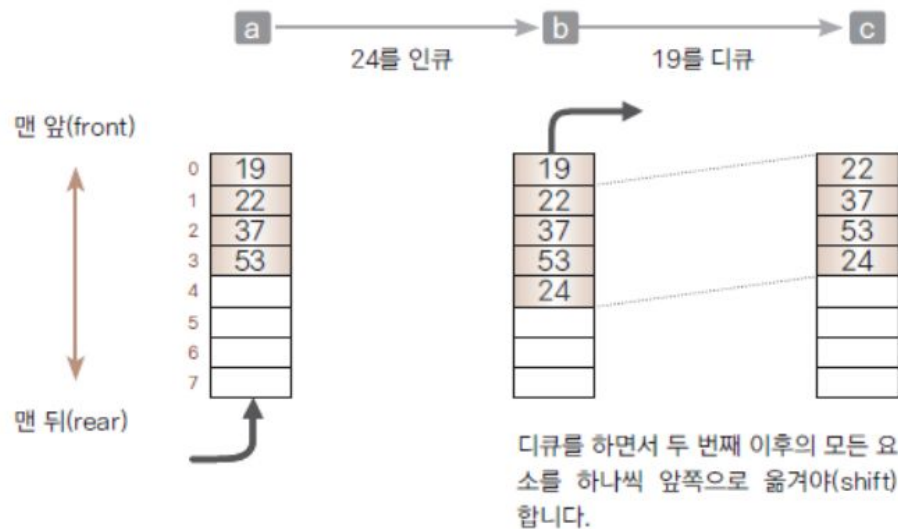
Queue

선입선출(FIFO)

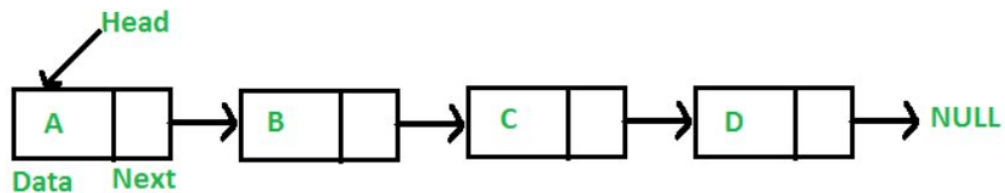
First In First Out 구조를 가진
자료구조



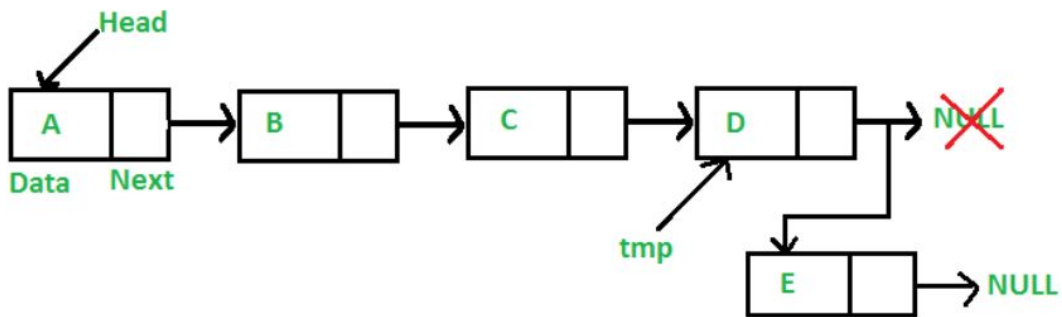
배열로 Queue 만들기



LinkedList



데이터 추가



LinkedList

데이터 삭제

