

# CS 스터디 자료구조 1주차

이서연

# CONTENTS

---

## 01 Intro

01 자료구조의 개념

02 자료구조와 알고리즘

03 자료구조의 종류

# 01 Intro

---

## 01 자료구조의 개념

02. 자료구조와 알고리즘

03. 자료구조의 종류

---

# 자료구조

---

- 개념
  - 모델링된 데이터를 컴퓨터에 저장/운영하는 방식
- 종류
  - 선형 구조 : 리스트, 스택, 큐
  - 비선형 구조 : 트리, 그래프

# 01 Intro

---

01. 자료구조의 개념

## 02 자료구조와 알고리즘

03. 자료구조의 종류

---

# 자료구조와 알고리즘

---

- 자료구조
  - 데이터를 저장/운영하는 방식
  - 공간 복잡도와 연관성 有
- 알고리즘
  - 작업의 구체적 수행 방식
  - 시간 복잡도와 연관성 有
- 프로그램 = 자료구조 + 알고리즘

# 01 Intro

---

01. 자료구조의 개념

02. 자료구조와 알고리즘

**03 자료구조의 종류**

---

## 배열 (Array)

---

- 정적으로 필요한 만큼만 원소를 저장할 수 있는 공간 할당
- 각 원소의 주소는 연속적으로 할당됨
- index를 통해  $O(1)$ 에 접근 가능
- 삽입 및 삭제는  $O(N)$
- 지정 개수를 초과하면, 배열 크기를 재할당한 후 복사

# 01 Intro

---

01. 자료구조의 개념

02. 자료구조와 알고리즘

**03 자료구조의 종류**

---

## 리스트 (List)

---

- 노드(Node)들의 연결로 이루어짐
- 크기 제한이 없음
- 다음 노드에 대한 참조를 통해 접근  $\rightarrow O(N)$
- 삽입과 삭제 용이  $\rightarrow O(1)$

# 01 Intro

---

01. 자료구조의 개념

02. 자료구조와 알고리즘

**03 자료구조의 종류**

---

## ArrayList

---

- 동적으로 크기가 조정되는 배열
- 배열이 가득 차면, 알아서 배열의 크기를 2배로 할당하고 복사 수행
- 재할당에 걸리는 시간은  $O(N)$
- 재할당은 자주 발생하지 않으므로 접근 시간은  $O(1)$

# 01 Intro

---

01. 자료구조의 개념

02. 자료구조와 알고리즘

**03 자료구조의 종류**

---

## 스택 (Stack)

---

- LIFO(Last In First Out) 방식
- 원소의 삽입, 삭제가 한쪽 끝에서만 이루어짐 (top)
- 함수 호출 시, 지역변수와 매개변수를 스택에 저장



# 01 Intro

---

01. 자료구조의 개념

02. 자료구조와 알고리즘

**03 자료구조의 종류**

---

## 큐 (Queue)

---

- FIFO(First In First Out) 방식
- 원소의 삽입, 삭제가 양쪽 끝에서 일어남 (front, rear)
- 은행 대기열 등에 사용

# 01 Intro

---

01. 자료구조의 개념

02. 자료구조와 알고리즘

**03 자료구조의 종류**

---

## 우선순위 큐 (Priority Queue)

---

- FIFO 방식이 아닌 데이터를 근거로 한 우선순위 판단
- 우선순위가 높은 것부터 나감
- 배열, 연결리스트, 힙으로 구현 가능

# 01 Intro

---

01. 자료구조의 개념

02. 자료구조와 알고리즘

**03 자료구조의 종류**

---

## 우선순위 큐 (Priority Queue)

---

- 배열
  - 간단하게 구현 가능
  - 데이터 삽입 및 삭제 시,  $O(N)$ 으로 비효율적
  - 우선순위가 가장 낮은 경우, 삽입 위치를 찾기 위해 배열의 모든 데이터를 탐색해야 함

# 01 Intro

---

01. 자료구조의 개념

02. 자료구조와 알고리즘

**03 자료구조의 종류**

---

## 우선순위 큐 (Priority Queue)

---

- 연결리스트
  - 데이터의 삽입 및 삭제에 걸리는 시간은  $O(1)$
  - 삽입 위치를 찾을 때에는 배열과 같이 비효율적

# 01 Intro

01. 자료구조의 개념

02. 자료구조와 알고리즘

**03 자료구조의 종류**

## 우선순위 큐 (Priority Queue)

- 힙
  - 삽입 및 삭제와 삽입 위치 찾기 모두 효율적 처리 가능
  - 우선순위 큐는 대부분 힙으로 구현
  - 완전 이진트리의 성질을 만족하므로 1차원 배열로 표현 가능 → 접근 시간 :  $O(1)$
  - 데이터의 삽입은 leaf node부터, 삭제는 root node부터 수행

# 01 Intro

---

01. 자료구조의 개념

02. 자료구조와 알고리즘

**03 자료구조의 종류**

---

## 트리 (Tree)

---

- 사이클이 없는 무방향 그래프
- 완전 이진트리 기준, 높이는  $\log N$
- 트리 순회 방법 4가지
  1. 중위 순회 : left - root - right
  2. 전위 순회 : root - left - right
  3. 후위 순회 : left - right - root
  4. 레벨 순서 순회 : BFS와 동일하여 큐로 구현 가능

# 01 Intro

---

01. 자료구조의 개념

02. 자료구조와 알고리즘

**03 자료구조의 종류**

---

## 이진탐색트리 (BST)

---

- left는 root보다 작은 값들, right는 root보다 큰 값으로 구성
- 삽입 및 삭제, 탐색까지 이상적인 경우에는 모두  $O(\log N)$  소요
- 최악의 경우(편향된 트리),  $O(N)$

# 01 Intro

---

01. 자료구조의 개념

02. 자료구조와 알고리즘

**03 자료구조의 종류**

---

## 해시 테이블 (Hash Table)

---

- 효율적 탐색을 위한 자료구조
- {key, value} 쌍으로 이루어짐
- 해시 함수를 통해 입력받은 key를 index로 대응시킴
- 충돌에 대한 고려 필요
- 해시 함수는 충돌 정책에 따라 성능 차이가 큼



# 01 Intro

01. 자료구조의 개념

02. 자료구조와 알고리즘

**03 자료구조의 종류**

## 해시 테이블 (Hash Table)

- 충돌 해결 방안
  1. 선형 조사법 (linear probing)
    - 충돌이 일어난 항목을 해시 테이블의 다른 위치에 저장
  2. 이차 조사법
    - 선형 조사법에서 발생하는 집적화 문제 완화

# 01 Intro

01. 자료구조의 개념

02. 자료구조와 알고리즘

**03 자료구조의 종류**

## 해시 테이블 (Hash Table)

### 3. 이중 해시법

- 재해싱 (rehashing)이라고도 부름
- 충돌로 인해 비어있는 버킷을 찾을 때, 추가적인 해시 함수  $h'()$ 를 사용

# 01 Intro

01. 자료구조의 개념

02. 자료구조와 알고리즘

**03 자료구조의 종류**

## 해시 테이블 (Hash Table)

### 4. 체이닝

- 각 버킷을 고정된 개수의 슬롯 대신, 유동적 크기를 갖는 연결리스트로 구성
- 충돌뿐만 아니라 오버플로우 문제도 해결 가능
- 버킷 내에서 항목을 찾을 때에는 연결리스트 순차 탐색

활용

# 01 Intro

---

01. 자료구조의 개념

02. 자료구조와 알고리즘

**03 자료구조의 종류**

---

## 해시 테이블 (Hash Table)

---

### 5. 해싱 성능 분석

- $a = n / M$
- $a$  : 적재 비율
- $n$  : 저장되는 항목 개수
- $M$  : 해시테이블 크기

# 01 Intro

01. 자료구조의 개념

02. 자료구조와 알고리즘

**03 자료구조의 종류**

## 해시 테이블 (Hash Table)

- 맵 (Map)
  - 이진탐색트리(BST)를 사용하다가 최근에는 레드블랙트리 사용
  - key 값을 이용해 트리를 탐색하는 방식
  - 데이터 접근, 삽입, 삭제는  $O(\log N)$
- 해시맵 (HashMap)
  - 해시 함수를 활용에  $O(1)$ 에 접근 가능

# 01 Intro

01. 자료구조의 개념

02. 자료구조와 알고리즘

**03 자료구조의 종류**

## 해시 테이블 (Hash Table)

- 레드블랙트리 (Red-black Tree)
  - 일종의 자기균형 이진탐색트리 (Self-Balancing BST)
  - 삽입 및 삭제 과정에서 트리의 모양이 균형을 이루도록  
각 노드들은 red 또는 black 색상을 가짐
  - 삽입 및 삭제 시, 최악의 경우에도  $O(\log N)$  보장

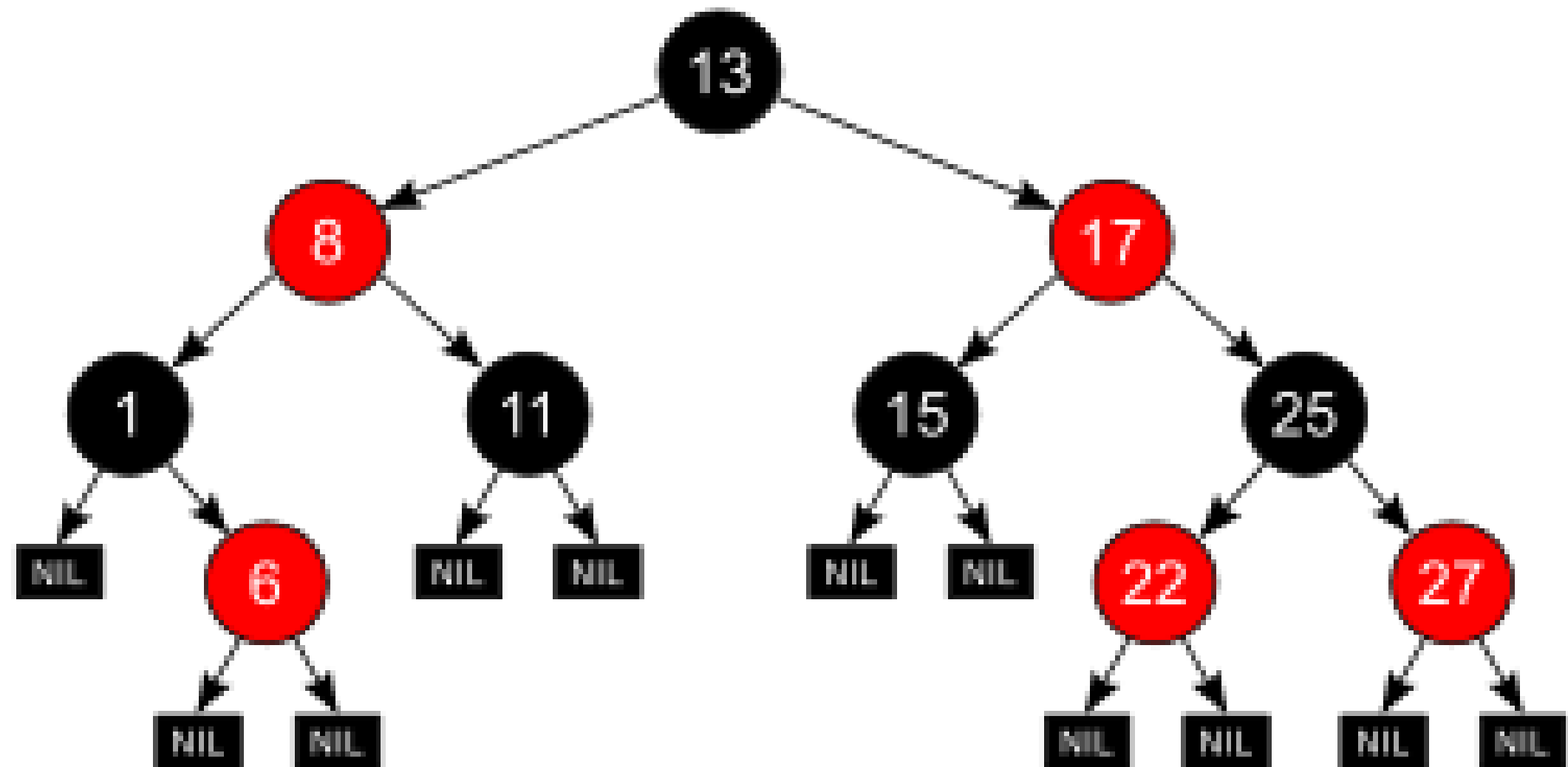
# 01 Intro

01. 자료구조의 개념

02. 자료구조와 알고리즘

**03 자료구조의 종류**

## 해시 테이블 (Hash Table)



---

# 참고 자료

---

- <https://github.com/gyoogle/tech-interview-for-developer>
- 2020 자료구조 김승태 교수님 강의 자료
- <https://suhwanc.tistory.com/197?category=730826>



---

감사합니다

---