



# BIT BY BIT

## 자료구조

그래프이론의 기초, 트리, 인접행렬, 인접리스트, 맵, 셋, 해시테이블, 힙, 배열, 연결리스트

# 스터디 세션

10분 opening

60분 공유 세션

10분 마무리

# Opening

- 강의 내용은 좀 어땠나요?
- 공유 세션 시작하기 전에 하고 싶은 얘기가 있나요?

# Contents

그래프 이론의 기초 (Graph, Vertex, Edge, Weight)

트리 (기본, 이진트리, 이진탐색트리)

인접행렬, 인접리스트

맵 (Map)

셋 (Set)

해시테이블 (Hash Table)

힙 (Heap)

자료구조의 시간복잡도 총정리

배열과 연결리스트 차이

# 그래프 이론의 기초

Graph, Vertex, Edge, Weight

정점과 간선에 대해 설명해 보시오.

Q. ‘양방향 간선’ 과 ‘무방향 간선’ 은 같은 개념인가?

정점으로 나가는 간선을 해당 정점의 \_\_\_\_ 라고 하며, 들어오는 간선을 해당 정점의 \_\_\_\_ 라고 한다.

가중치란 무엇인가요?

# 트리

기본, 이진 트리, 이진 탐색 트리

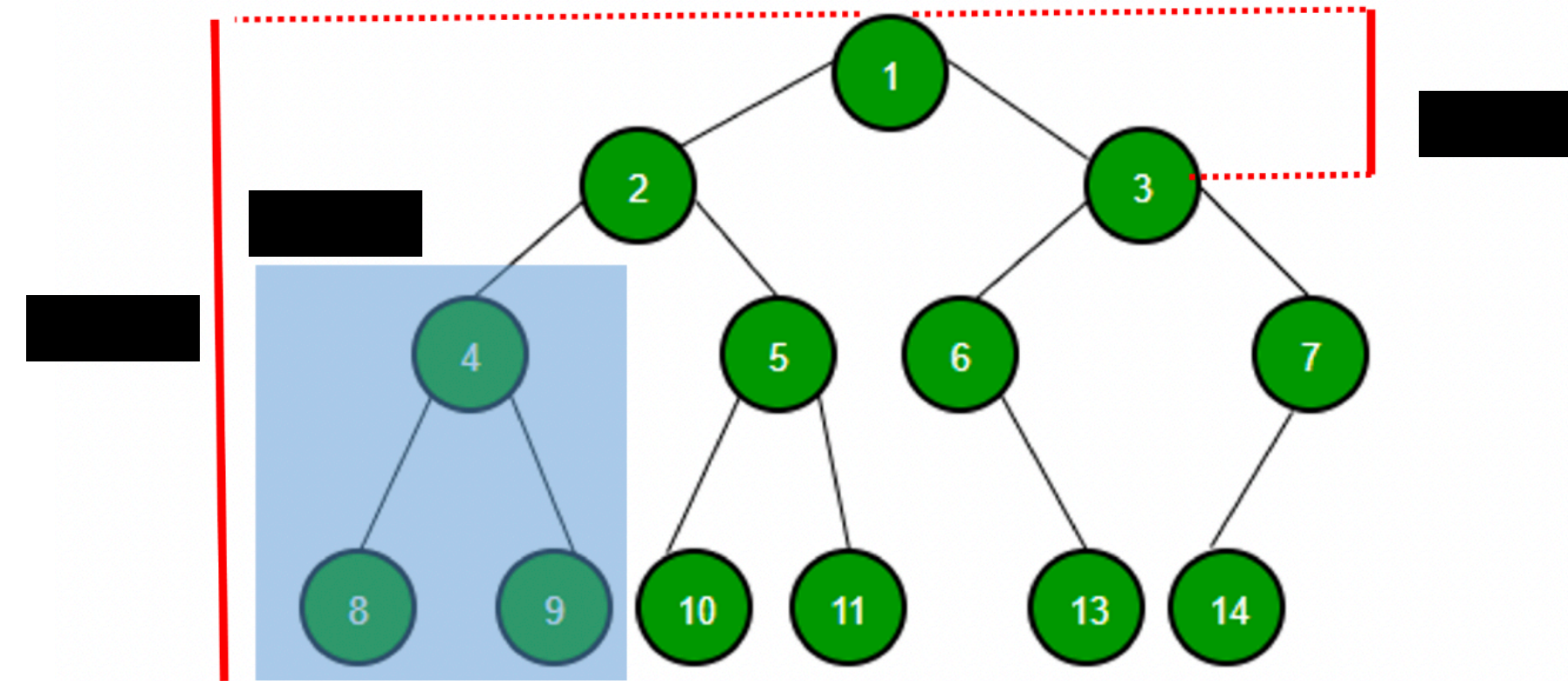


트리의 개념에 대해 설명해 보시오.

=> 트리는 \_\_\_\_\_ 와 \_\_\_\_\_ 로 이루어진 \_\_\_\_\_ 인 구조를 가지며 \_\_\_\_\_ 그래프의 일종이자 \_\_\_\_\_ 이 없는 자료구조를 의미한다.

Q. 트리의 정점의 개수를  $V$  개 라고 한다면, 간선( $E$ )의 개수는 무엇인가?

Q. 트리 내 서로 연결이 안 돼 있는 노드가 존재할 수 있다. (T / F)



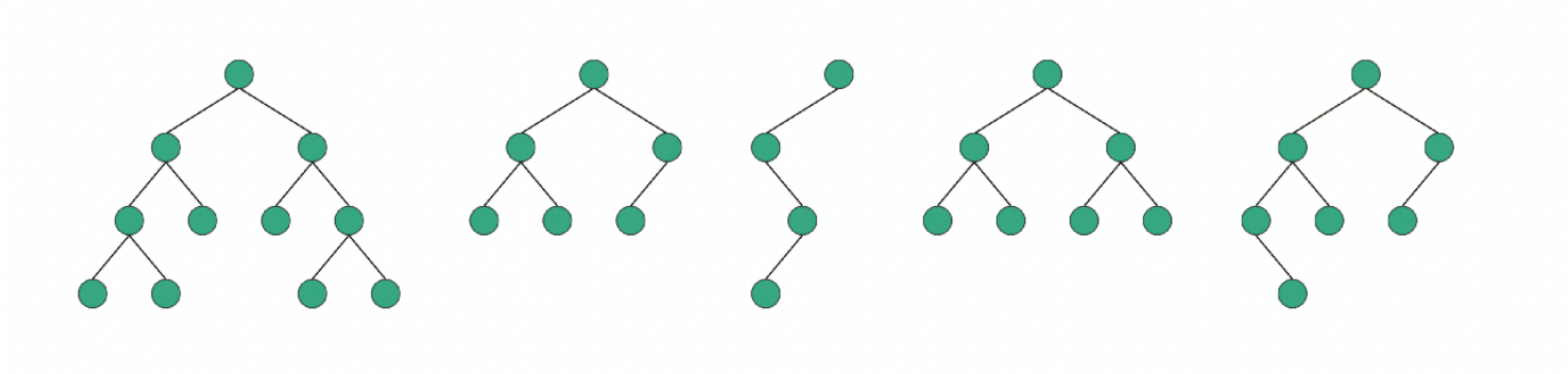
루트 노드에서 특정 노드까지 최단거리로 갔을 때의 거리란 무엇인가?

루트 노드부터 리프 노드까지의 거리 중 가장 긴 거리를 무엇이라 하는가?

레벨(level)이란 무엇인가?

트리 내의 하위 집합을 무엇이라 하는가?

트리로 이루어진 집합을 무엇이라 하는가?



완전 이진 트리 (complete binary tree)

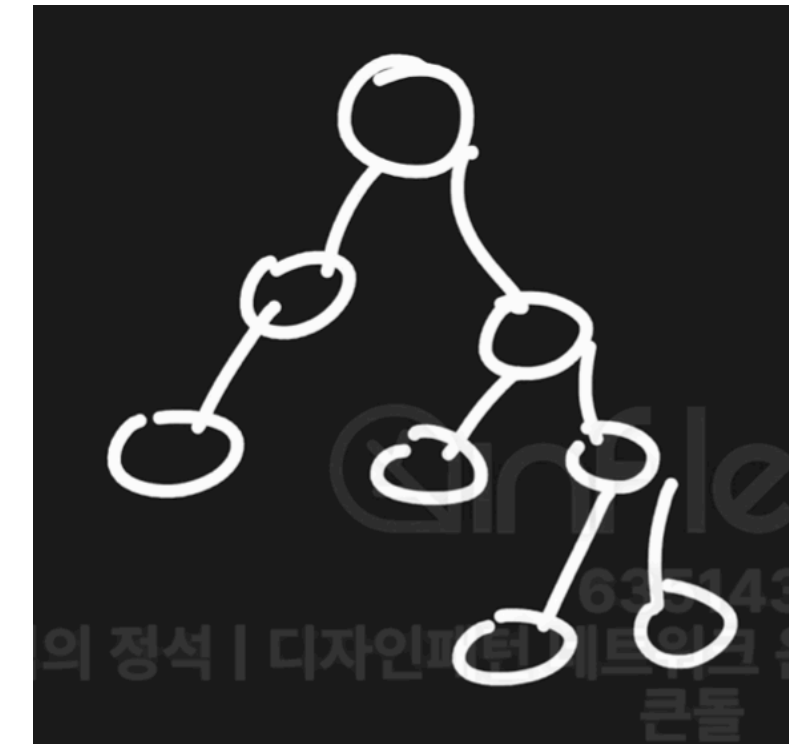
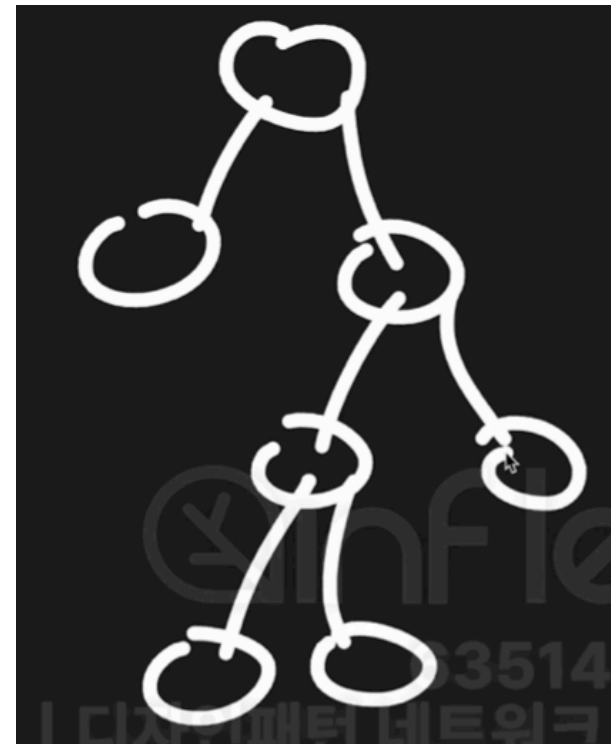
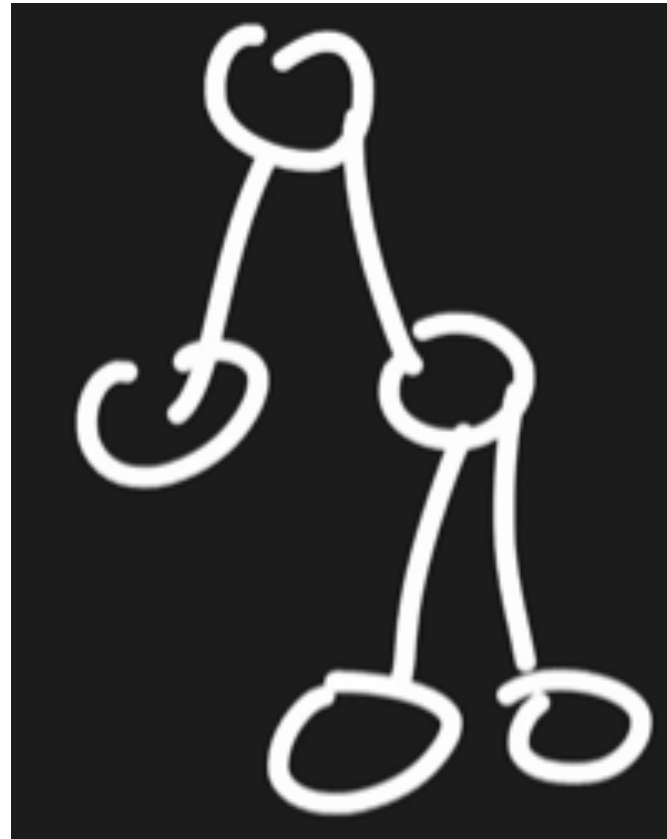
균형 이진 트리 (balanced binary tree)

포화 이진 트리 (perfect binary tree)

정이진 트리 (full binary tree)

변질 이진 트리 (degenerate binary tree)

균형 이진 트리인지 맞혀봅시다~!



이진탐색트리(binary search tree)의 개념을 설명해 보시오.

이진탐색트리의 시간복잡도?

이진탐색트리는 \_\_\_\_\_의 영향을 받아, 경우에 따라 선형적인 자료구조가 될 수 있다.

=> \_\_\_\_\_, \_\_\_\_\_와 같은 방식을 사용해 “균형잡히게” 만들 수 있다.

# 인접 행렬, 인접 리스트

인접행렬(adjacency matrix)의 개념을 설명해보시오.

a 라는 2차원 배열이 있을 때, 3번 노드에서 5번 노드로 가는 단방향 경로가 있고 이를 인접행렬로 표현한다면?

(이어서) 3번 노드에서 5번 노드로 가는 양방향 경로가 있고 이를 인접행렬로 표현한다면?

인접리스트(adjacency list)의 개념과 특징을 설명해보시오.



## 인접행렬 vs 인접리스트

(정점의 개수를  $V$ , 간선의 개수를  $E$  라고 할 때)

Q. 공간복잡도는 각각 무엇일까요?

Q. 간선을 하나 찾는 것의 시간복잡도는 각각 무엇일까요?

Q. 모든 간선을 찾는 것의 시간복잡도는 각각 무엇인가요?

인접리스트와 인접행렬, 각각 어떤 상황에서 어떤 것을 사용하는 것일까요?

**맵 (Map)**

Map 의 개념을 설명해보시오.

Map 에서 참조, 탐색, 삽입, 삭제를 하는 데에 걸리는 시간 복잡도는 각각 무엇인가요?

**셋 (Set)**

Set 의 개념에 대해 설명해보시오.

Set 에서 참조, 탐색, 삽입, 삭제 각각의 시간복잡도는 무엇인가요?

# 해시 테이블 (Hash Table)

Hash Table 의 개념을 설명해 보시오.

Q. 해시란 무엇인가?

Q. 해싱이란 무엇인가?

Q. 해시 함수란 무엇인가?



해시테이블의 시간복잡도

=> 참조, 탐색, 삽입, 삭제의 평균 시간복잡도는 무엇인가요?

=> 참조, 탐색, 삽입, 삭제의 최악 시간복잡도는 무엇인가요?

해시테이블의 충돌 문제(hash collision)

Q. 해시 충돌의 개념에 대해 설명해 보시오.

Q. 해시 충돌에 대한 해결책 중 체이닝(chaining)에 대해 설명해 보시오.

Q. 해시 충돌에 대한 해결책 중 개방 주소법(open addressing)에 대해 설명해 보시오.

# HashMap 과 HashTable 의 차이점은 무엇일까요?

비교 항목	HashMap	HashTable
동기화	비동기 (스레드 안전 X)	동기화됨 (스레드 안전)
성능	빠름	느림
Null 허용	키 null 1개, 값 여러 개 가능	Null 불가능
내부 구조	해시 충돌 시 트리 구조 사용 (Java 8+)	해시 충돌 시 LinkedList 유지
사용 추천	단일 스레드 환경	멀티 스레드 환경 (그러나 ConcurrentHashMap 이 더 좋음)

# 힙 (Heap)

힙(Heap)의 개념에 대해 설명해 보시오.

힙에서 데이터를 삽입하는 과정을 설명해 보시오. (최대힙 상정하고)

힙에서 데이터를 삭제하는 과정을 설명해 보시오. (최소힙 상정하고)

힙(heap)과 이진탐색트리(binary search tree)와의 차이점을 설명해 보시오.

힙의 참조, 탐색, 삽입, 삭제 각각의 시간 복잡도는 무엇인가요?

# 자료구조의 시간복잡도 총정리

배열(vector)

참조	
탐색	
맨 끝에 삽입 / 삭제	
맨 끝 제외 삽입 / 삭제	

스택(stack)

n번째 참조	
가장 앞 부분 참조	
탐색	
삽입 / 삭제(n번째 제외)	

연결리스트(doubly linked list)

참조	
탐색	
삽입 / 삭제	

큐(queue)

n번째 참조	
가장 앞 부분 참조	
탐색	
삽입 / 삭제(n번째 제외)	



# 맵(map)

참조	
탐색	
삽입 / 삭제	

# 셋(set)

참조	
탐색	
삽입 / 삭제	

# 해시테이블(hash table)

평균 - 참조	
평균 - 탐색	
평균 - 삽입/ 삭제	
최악 - 참조	
최악 - 탐색	
최악 - 삽입 / 삭제	

# 힙(heap)

참조 (최대 또는 최솟값)	
탐색	
삽입 / 삭제	

# LRU 페이지 교체 알고리즘

LRU 알고리즘에 대해 설명해 보시오.

핵심 포인트

- 최근에 사용한 페이지는 유지되고, 가장 오랫동안 사용하지 않은 페이지가 삭제됨
- 캐시 교체 정책에서 자주 사용됨
- 구현 방법으로는 LinkedList + HashMap

# 배열과 연결리스트 차이

	배열	연결리스트
메모리 공간		
삽입, 삭제 (맨 끝, 맨 앞 제외)		
n번째 요소 참조		
탐색		

연결리스트를 사용하기 좋을 때와 배열을 사용하기 좋을 때는 각각 언제인가요?

# 마무리

- 추가로 할 얘기가 있을까요
- 다음 세션 논의

**감사합니다**