



Section3. 비선형 자료구조

비선형 자료구조란?

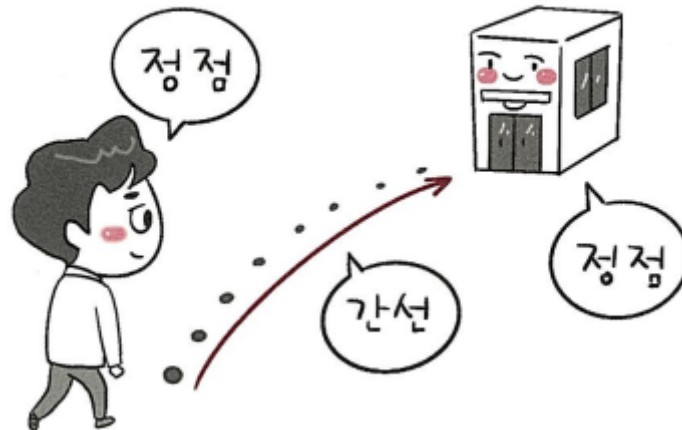
- 일렬로 나열하지 않고, 자료 순서나 관계가 복잡한 구조.
- 일반적으로 트리나 그래프를 말함.

5.3.1 그래프

- 정점과 간선으로 이루어진 자료구조.

정점과 간선이란?

▼ 그림 5-9 정점과 간선



특정 점에서 특정 점으로 무언가를 통해 이동한다했을 때,

- **정점** : 특정 점은 정점.
- **간선** : 무언가는 간선.
 - 양방향 간선, 단방향 간선
 - 정점으로 나가는 간선을 정점의 outdegree
 - 들어오는 간선을 정점의 indegree 라고 함.

가중치란?

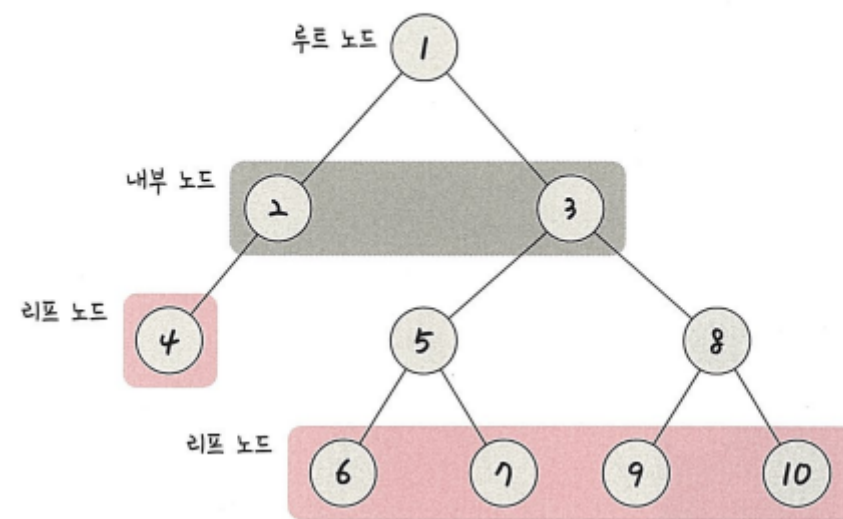
- 간선의 정점 사이에 드는 비용.

5.3.2 트리

- 그래프 중 하나로, 그래프의 특징처럼 정점과 간선으로 이루어짐.
- 트리구조로 배열된 일종의 계층적 데이터 집합.
- 루트노드, 내부노드, 리프노드 등으로 구성됨.
- 트리로 이루어진 집합을 숲이라고 함.

트리의 특징

▼ 그림 5-13 트리의 특징



트리의 특징.

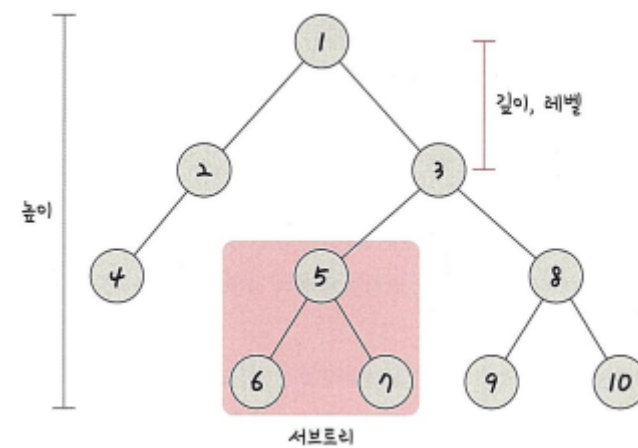
- 부모, 자식 계층 구조를 가짐.
 - 같은 경로상에서 어떤노드보다 위면 부모 아래면 자식
- $V - 1 = E$
 - 즉, 정점개수 - 1 은 간선 수
- 임의의 두 노드 사이의 경로는 유일무이하게 존재함.
- 트리 내에서 어떤 노드와 어떤노드까지의 경로는 반드시 있다.

트리의 구성

- 루트노드
 - 가장 위에 있는 노드
- 내부노드
 - 루트 노드와 내부 노드 사이의 노드
- 리프노드
 - 자식노드가 없는 노드

트리의 높이와 레벨

▼ 그림 5-15 트리의 높이와 레벨

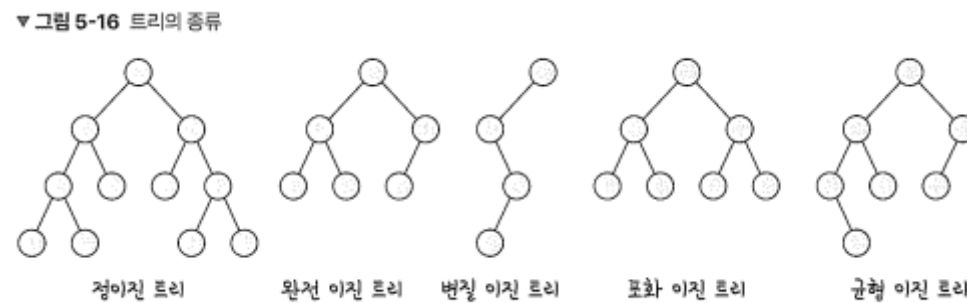


- 깊이
 - 루트 노드 부터 특정 노드까지 최단거리로 갔을 때의 거리
- 높이

- 루트노드부터 리프노드까지 거리가 가장 긴 거리
- **레벨**
 - 보통 깊이와 같은 의미를 지님.
 - 1번노드는 0레벨, 2번 노드, 3번 노드까지는 1레벨 ~~..
- **서브트리**
 - 트리 내의 하위 집합.

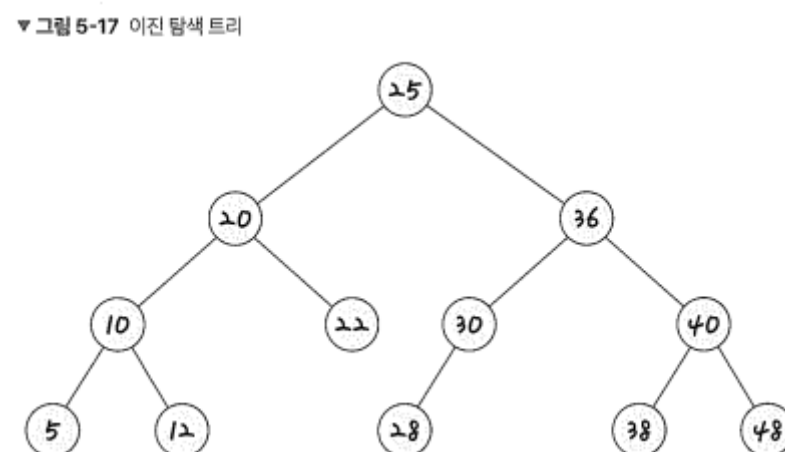
이진트리

- 자식의 노드 수가 두 개 이하인 트리.



- **이진트리의 종류**
 - **정이진 트리**
 - 자식 노드가 0 또는 두 개인 이진트리
 - **완전 이진 트리**
 - 왼쪽에서 부터 채워져있는 이진트리
 - **변질 이진 트리**
 - 자식 노드가 하나 밖에 없는 이진 트리
 - **포화 이진 트리**
 - 모든 노드가 짝 차 있는 이진 트리
 - **균형 이진트리**
 - 왼쪽과 오른쪽의 높이 차이가 1이하인 이진트리.
 - map과 set을 구성하는 레드 블랙트리는 균형이진트리 중 하나.

이진 탐색 트리(BST)

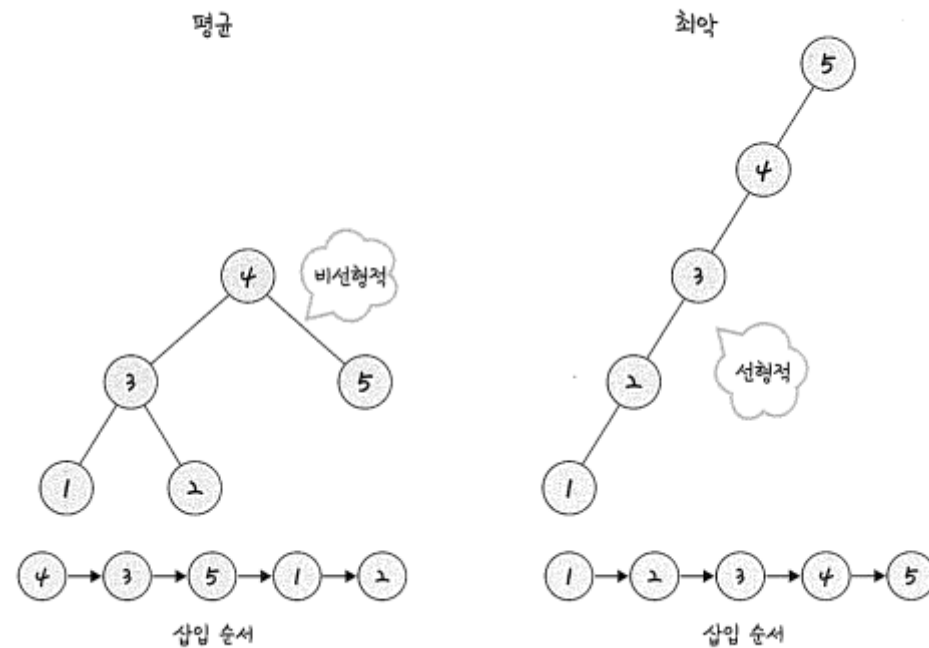


- 노드의 오른쪽 하위 트리에는 **노드 값보다 큰 값**만 포함.
- 왼쪽 하위 트리에는 **노드 값보다 작은 값**만 포함.

복잡도

- 요소 탐색시 평균 $O(\log n)$ 소요.
- 최악의 경우 $O(n)$ 소요

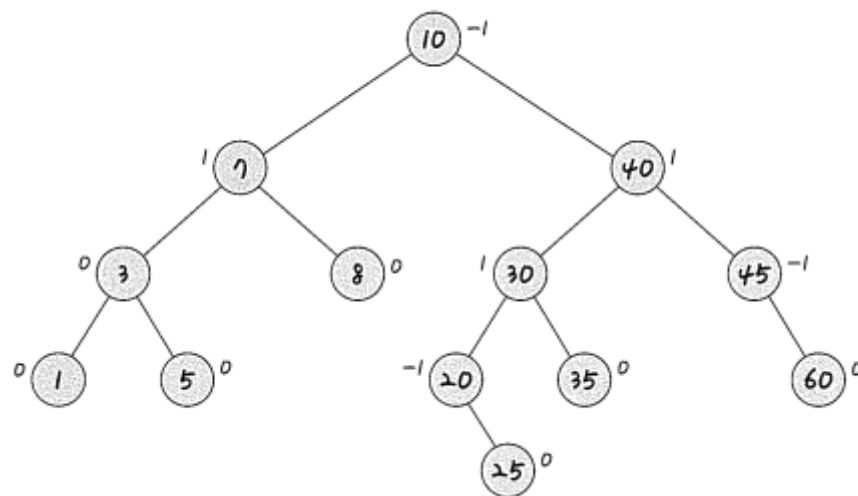
▼ 그림 5-18 이진 탐색 트리의 선형적인 모습



AVL 트리

- 최악의 경우, 선형적인 트리가 되는 것을 방지.
- 스스로 균형을 잡는 이진 탐색 트리
- 두 자식 서브트리 높이는 항상 최대 1만큼 차이난다는 특징.

▼ 그림 5-19 AVL 트리

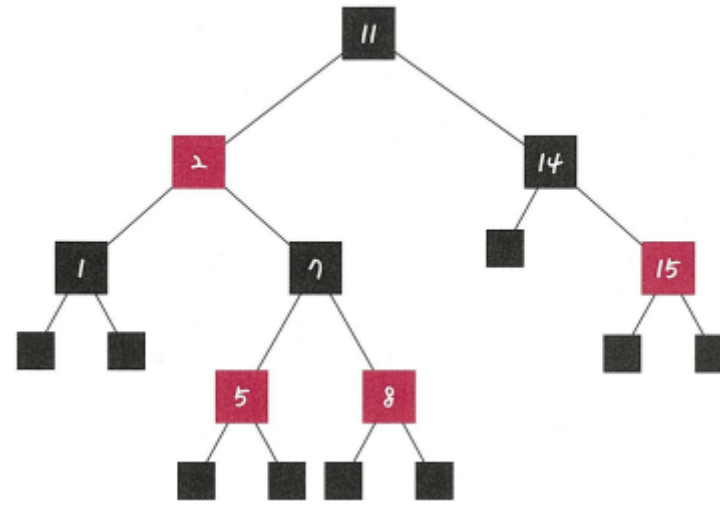


- 탐색, 삽입, 삭제 모두 $O(\log n)$
- 삽입 삭제시 마다 균형이 안맞는 것을 맞추기위해, 트리 일부를 오른쪽 혹은 왼쪽으로 회전시키며 균형을 잡음.

레드 블랙트리

- 균형 이진 탐색트리.
- 탐색, 삽입, 삭제 모두 $O(\log n)$
- 각 노드는 빨간색 혹은 검은색을 나타내는 추가비트를 저장.
 - 이는 삽입 삭제 중 트리가 균형을 유지하는데 사용됨

▼ 그림 5-20 레드 블랙 트리

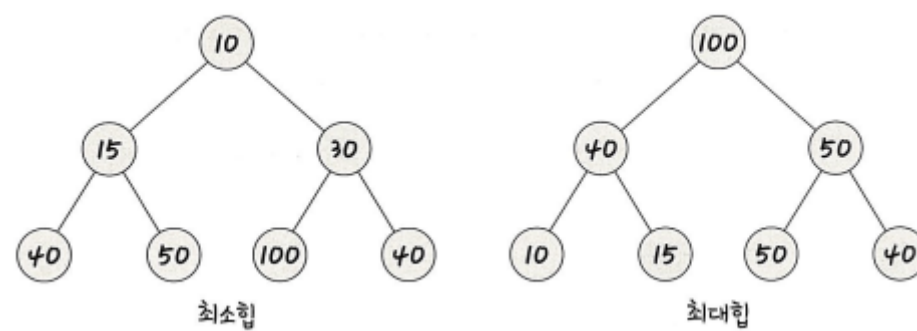


- 모든 리프노드와 루트노드는 블랙.
- 어떤 노드가 레드이면, 그 노드의 자식은 반드시 블랙.

5.3.3 힙

- 힙이란?
 - 완전 이진 트리 기반의 자료구조
- 힙의 종류

▼ 그림 5-21 최소힙과 최대힙

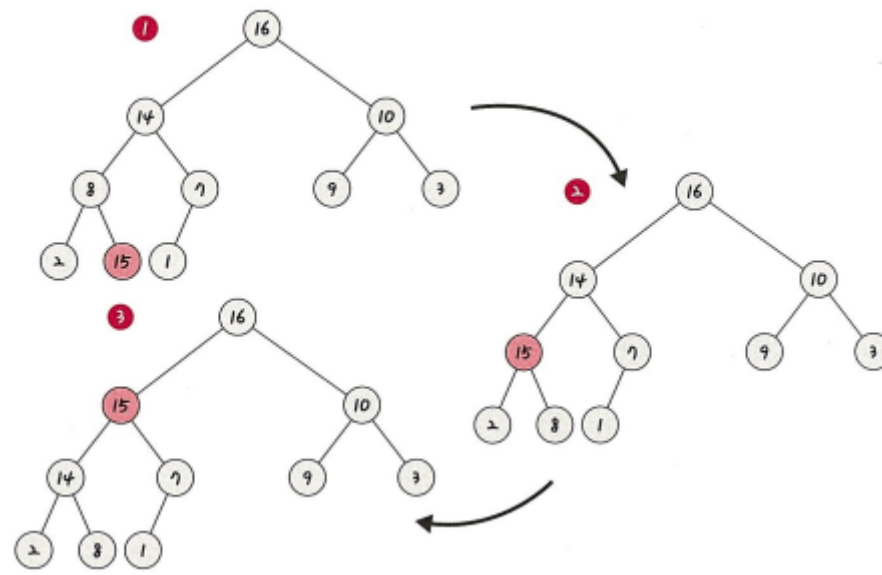


- 최소힙
 - 루트노드에 있는 키는 모든 자식에 있는 키중 가장 커야 함.
 - 각 노드의 자식 노드와 관계도 이와 같은 특징이 재귀적으로 이뤄져야 함.
- 최대힙
 - 루트 노드에 있는 키는 모든 자식에 있는 키 중 최솟값이어야 함.
 - 각 노드의 자식 노드와의 관계도 이와 같은 특징이 재귀적으로 이뤄져야 함.

최대힙의 삽입

- 최소힙도 비슷하게 동작.

▼ 그림 5-22 최대힙의 삽입



• 동작과정

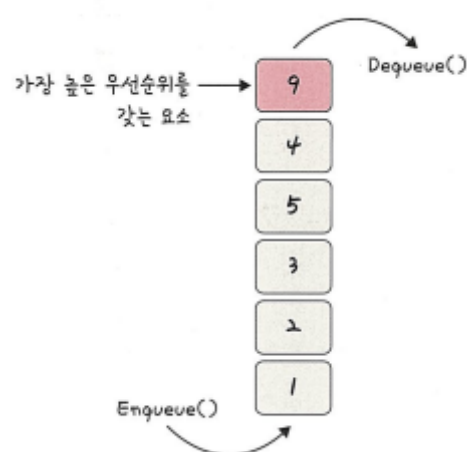
1. 힙에 새로운 요소가 들어옴.
2. 새로운 노드를 힙의 마지막에 이어서 삽입함.
3. 새로운 노드를 부모 노드들과의 크기를 비교하며 교환해서 힙의 성질을 만족시킴.

최대 힙의 삭제

1. 최댓값은 루트 노드이므로 루트노드가 삭제됨.
2. 마지막 노드와 루트 노드를 스왑하여 또 다시 스왑 등의 과정을 거쳐 재구성됨.

5.3.4 우선순위 큐

▼ 그림 5-23 우선순위 큐



- 우선순위 대기열이라고도 함.
- 대기열에서 우선순위가 높은 요소가 우선 순위가 낮은 요소보다 먼저 제공되는 자료 구조.

5.3.5 맵

- 특정 순서에 따라 키와 매핑된 값의 조합으로 형성된 자료 구조
- 레드 블랙 트리 자료구조를 기반으로 형성됨
- 삽입 시 자동정렬.
- 특징
 - `map<String, int>` 형태로 구현.
- 주요 메소드

- 배열과 비슷하게 clear()로 모든 요소 삭제가능.
- size()로 크기 구할 수 있음.
- erase)로 키와 키에 매핑된 값을 지울 수 있다.

5.3.6 셋

- 특정 순서에 따라 고유한 요소를 저장하는 컨테이너
- 중복되는 요소는 없고, 희소한 값만 저장하는 자료구조

5.3.7 해시테이블

- 무한에 가까운 데이터들을 윗나한 개수의 해시 값으로 매핑한 테이블.
- 삽입, 삭제, 탐색 시 평균적으로 $O(1)$ 의 시간복잡도를 가짐.
- 작은 크기의 캐시 메모리로도 프로세스를 관리할 수 있음.

예상질문.

그래프와 트리의 차이점

- 그래프는 정점과 간선으로 이루어진 자료구조를 말함.
- 트리는 그래프의 종류 중 하나로 그래프의 특징처럼 정점과 간선으로 이루어짐.
 - 트리구조로 배열된 일종의 계층적 데이터 집합.
 - 루트노드, 내부노드, 리프노드 등으로 구성됨.
 - $V-1 = E$

이진 탐색트리의 문제점과 이를 해결하기 위한 트리를 설명하라.

- 이진탐색트리
 - 선형적으로 구 성될 대 시간복잡도가 $O(n)$ 으로 커지는 문제.
- 이를 해결하기 위한 트리
 - AVL 트리
 - 스스로 균형을 잡는 이진 탐색 트리
 - 두 자식 서브트리의 높이는 항상 최대 1만큼 차이가 남.
 - 탐색 삽입 삭제 모두 $O(\log n)$ 소요.
 - 삽입 삭제시마다 균형을 맞추기 위해 트리 일부를 오른쪽 혹은 왼쪽으로 회전.
 - 레드블랙트리