

# <힙(Heap)>

<https://github.com/limes22/algorithm/blob/main/go/DataStructure/Heap/heap.go>

힙은, 우선순위 큐를 위해 만들어진 자료구조다.

먼저 **우선순위 큐**에 대해서 간략히 알아보자

**우선순위 큐** : 우선순위의 개념을 큐에 도입한 자료구조

데이터들이 우선순위를 가지고 있음. 우선순위가 높은 데이터가 먼저 나감

스택은 LIFO, 큐는 FIFO

**언제 사용?**

시뮬레이션 시스템, 작업 스케줄링, 수치해석 계산

우선순위 큐는 배열, 연결리스트, 힙으로 구현 (힙으로 구현이 가장 효율적!)

힙 → 삽입 :  $O(\log n)$  , 삭제 :  $O(\log n)$

## 1. 힙(Heap)

완전 이진 트리의 일종

여러 값 중, 최대값과 최소값을 빠르게 찾아내도록 만들어진 자료구조

반정렬 상태

힙 트리는 중복된 값 허용 (이진 탐색 트리는 중복값 허용 X)

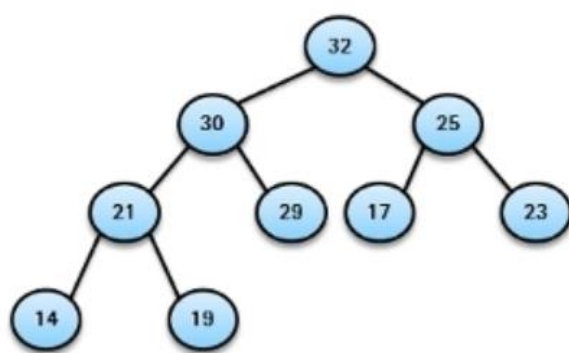
## 2. 힙 종류

## 최대 힙(max heap)

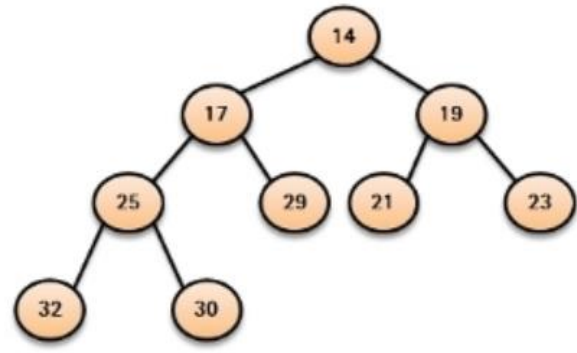
부모 노드의 키 값이 자식 노드의 키 값보다 크거나 같은 완전 이진 트리

## 최소 힙(min heap)

부모 노드의 키 값이 자식 노드의 키 값보다 작거나 같은 완전 이진 트리



- 최대 힙(Max Heap) -



- 최소 힙(Min Heap) -

## 3. 구현

힙을 저장하는 표준적인 자료구조는 배열

구현을 쉽게 하기 위해 배열의 첫번째 인덱스인 0 은 사용되지 않음

특정 위치의 노드 번호는 새로운 노드가 추가되어도 변하지 않음

(ex. 루트 노드(1)의 오른쪽 노드 번호는 항상 3)

부모 노드와 자식 노드 관계

왼쪽 자식  $\text{index} = (\text{부모 index}) * 2$

오른쪽 자식  $\text{index} = (\text{부모 index}) * 2 + 1$

부모  $\text{index} = (\text{자식 index}) / 2$

