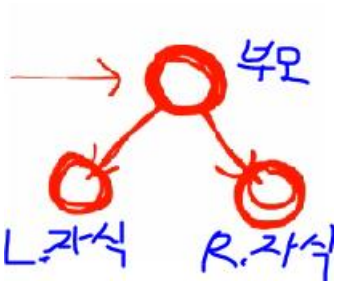
<6강.힙 정렬>

- 힙 구조의 특성을 이용
- 합병정렬과 동일한 수행시간 (nLogn)
- 삽입 정렬과 동일한 제자리 정렬

힙(The shape of a heap)

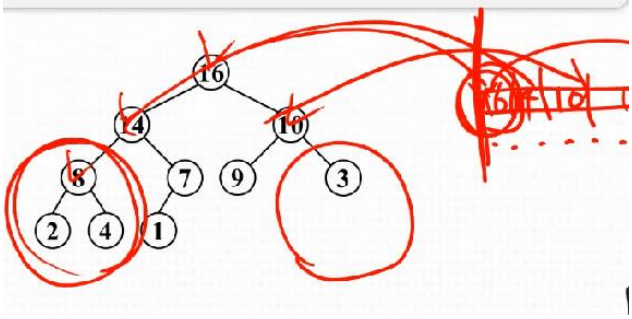
-완전 이진 트리(Root노드부터 Leaf노드까지)에 가까운 형태

-이진 트리는 자식 수가 최대 2개 (Binary tree)



[이진 트리]

부모=Root ,자식=Leaf



완전 이진트리는 자식이 왼쪽부터 채워나가야함

배열로 표시하면 다음과 같다.

[Max – Heap Property]

부모 노드의 값은 항상 자식 노드 값보다 큼

따라서 Root 노드 값이 가장 큼

또한 각 하위 트리구조의 Root 노드가 가장 큰 값

[Min – Heap Property]

자식 노드의 값은 항상 부모 노드의 값보다 큼

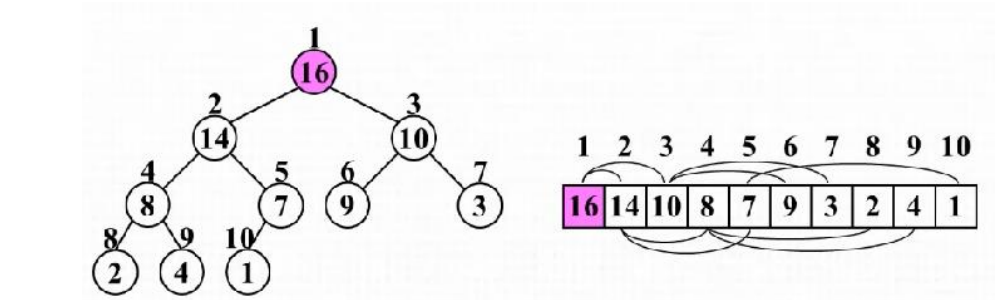
따라서 Root 노드 값이 가장 작다.

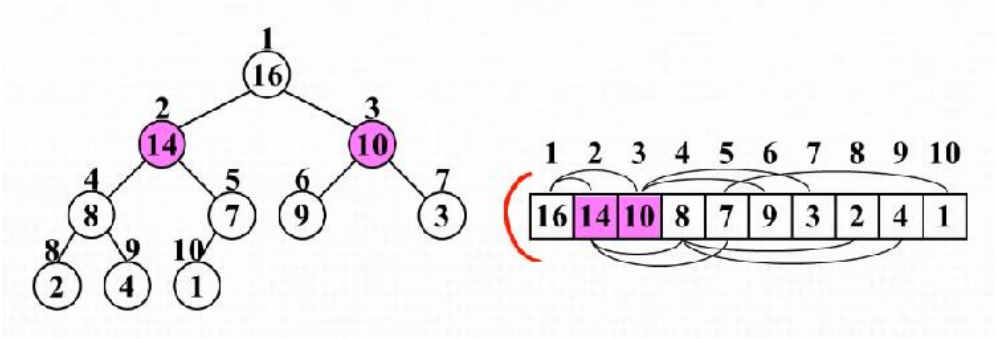
또한 각 하위 트리구조의 Root 노드가 가장 작은 값

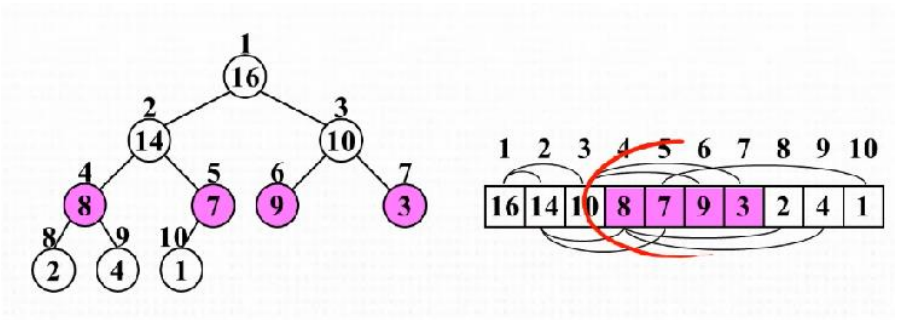
-Heap 배열 저장 방식

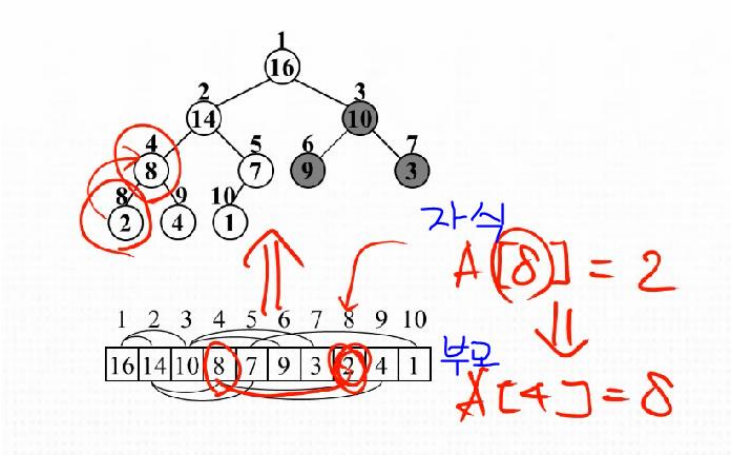
Root 노드는 배열의 첫번째 A[1]에 저장

각각의 노드들은 레벨별로 저장









Parent(I)

Return[I/2]

부모노드는 자식의 1/2

Left(I)

Return[2i]

왼쪽 자식은 부모 노드 * 2

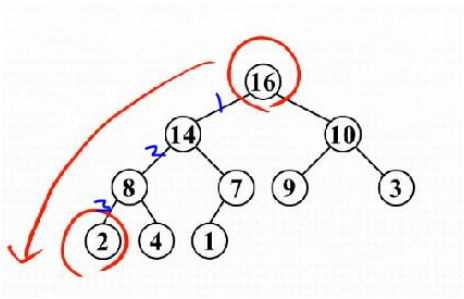
Right(I)

Return[2i+1]

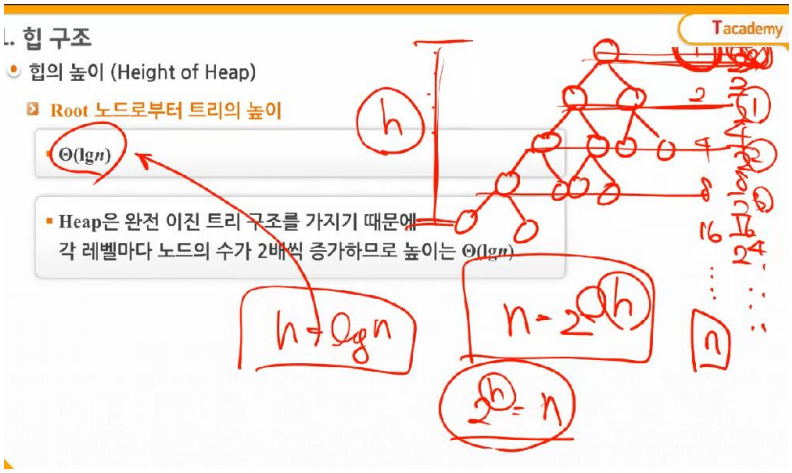
오른쪽 자식은 부모노드 *2 +왼쪽 자식노드 (1)

노드의 높이 (Height of Node)

노드의 높이는 현재 노드에서 Leaf 노드까지 내려갈 때 가장 단순하게 내려가는 가장 긴 경로에서 거쳐야 하는 간선의 수. (Simple path)

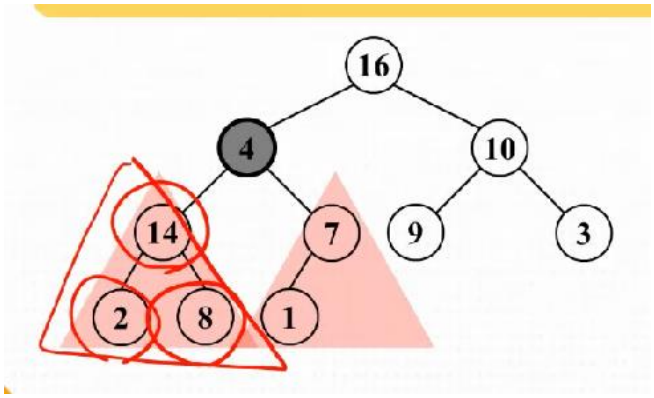


Ex) 16에서 2까지는 3번의 간선



[트리의 높이 구하는 증명]

Max-Heapify



-중간에 4라는 값 Max-heap 조건을 만족하지 않음(문제 발견)

-주어진 노드 값을 "흘려내리게"한다. 14를 올리면 됨