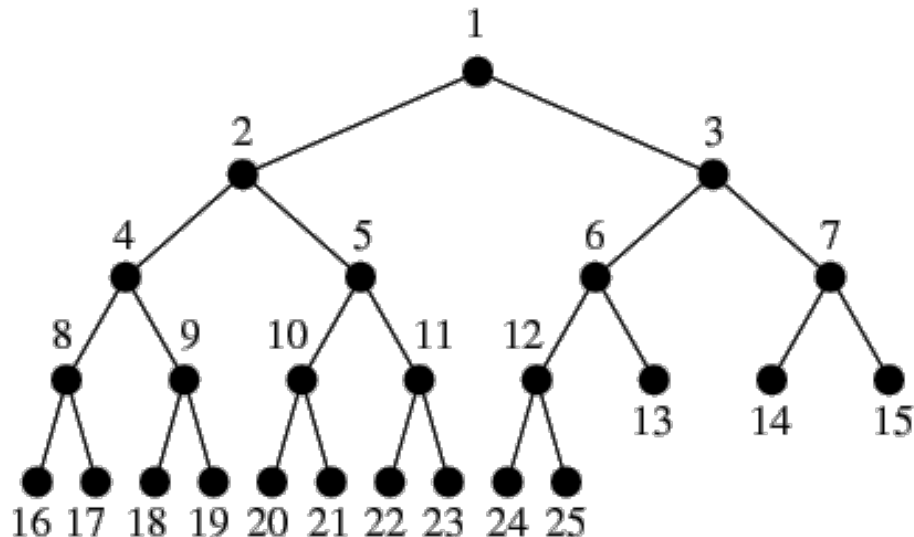


Heap

이번에는 Heap에 대해 배워 봅시다.

자료구조 중에서도 입력된 데이터중, 가장 우선순위가 큰, 즉, 가장 큰 값이나 가장 작은 값을 우선적으로 빼내는 자료구조를 **Heap** 이라고 합니다. 그렇다면 힙은 어떻게 만들어질까요?

우선 힙은 Complete Binary Tree로 구성되어 있습니다. Complete Binary Tree 가 무엇일까요? Complete Binary Tree 는 먼저 Level이 제일 낮은, 즉, Depth가 가장 얇은 곳을 먼저 채우고 만약 가장 마지막 레벨이었다면 왼쪽부터 채워나가는 것을 의미합니다. 또한 어떤 레벨에 대해 다음 레벨이 존재한다면 그 레벨에는 반드시 노드가 2^{Level} 만큼 채워져 있어야 합니다. (Level 은 0부터 시작)



Complete Binary Tree 의 예시

그렇다면 이러한 구조를 어떻게 활용해야 힙을 사용할 수 있을까요? 우선 예전에 배운 ArrayList(Vector)를 활용 해 볼 것입니다. 먼저, 위 그림에서 각 노드의 특성을 봅시다. 자기 자신의 부모를 알고 싶다면 현재 자신의 번호의 /2 한값의 버림을 구하면 됩니다. C/C++ 에서는 `cur / 2` 하면 자동적으로 버림이 됩니다. 그리고 자신의 왼쪽 자식은 현재 자신의 번호에서 곱하기 2를 한값이고 오른쪽 자식은 그 값에 +1 을 한 값입니다. 이것을 통하여 배열에서 트리를 구성하는 법을 알았습니다. 그렇다면 이제 본격적으로 힙을 사용하는 법을 알려드리겠습니다. 우선 size를 늘려나가면서 1-base 기준으로 수를 채워 나갑니다. 이때 채운 다음엔 자신의 부모노드보다 자신이 큰지 비교한 다음 바꿔줍니다. 그리고 다시 한 번 부모노드랑 비교하고, 이것이 루트가 되거나 자신이 더 작은경우 멈춥니다. 만약 Min Heap 을 구하고 있다면 자신이 부모노드 보다 작은지 비교하면 됩니다. 자세한것은 코드를 참고해 주세요.

그렇다면 입력된 자료중 가장 큰값은 어디에 위치해 있을까요. 네, 바로 1, 루트에 위치해 있습니다. 입력의 시간복잡도는 Binary Tree 의 특성상 $O(\log n)$ 이 됩니다. 그리고 제일 큰 값을 알아내는데에는 $O(1)$ 이 됩니다. 그렇다면 삭제는 어떻게 이루어질까요?

삭제는 꽤 복잡합니다. 우선 첫번째 노드를 제거하고, 맨 마지막, 즉, size에 위치해 있던 노드를 1로 가져온 뒤, size 에 -1을 해줍니다. 그 다음 첫번째로 가져온 노드를 왼쪽 자식과 오른쪽 자식을 비교해서 더 큰쪽으로 보내줍니다. 이때 더 큰쪽이 루트 자리를 차지 합니다. 그리고 보낸 노드는 다시한번 똑같은 작업을 반복해 줍니다. 만약 나보다 두 자식이 모두 작거나 같고, 자식이 둘다 존재하지 않는다면 멈춥니다. 이러한 작업의 자세한 내용은 코드를 참고 해주세요. 이 경우엔 역시나 Binary Tree 의 특성때문에 시간복잡도는 $O(\log n)$ 이 됩니다.

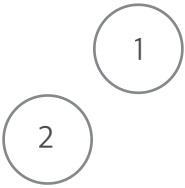
간단한 예시를 통해서 알아볼까요?

입력이 1, 2, 8, 9, 4, 10 이라고 해봅시다.

그러면 맨 처음 트리는 다음과 같이 됩니다.

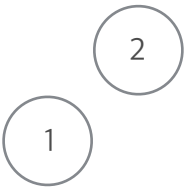
1	2	3	4	5	6	7
1						

그리고 배열은 다음과 같이 되겠지요, 그 다음 2가 들어옵니다.



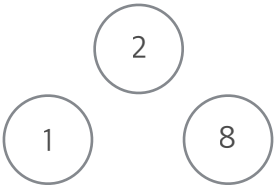
1	2	3	4	5	6	7
1	2					

트리의 배열의 끝은 위와 같이 될것이고, 이때 들어온 입력 2는 1보다 크므로 위치가 바뀝니다.

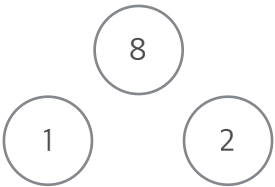


1	2	3	4	5	6	7
2	1					

이후 8이 들어오고

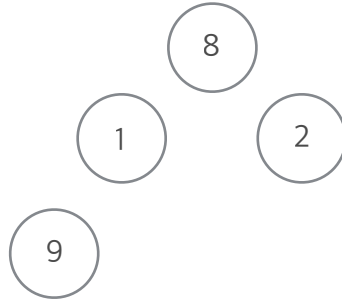


1	2	3	4	5	6	7
2	1	8				

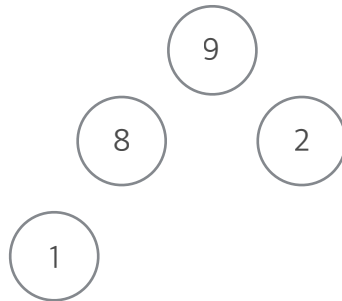


1	2	3	4	5	6	7
8	1	2				

다음과 같은 변화를 보이게 됩니다. 다음 9는 어떻게 될까요?

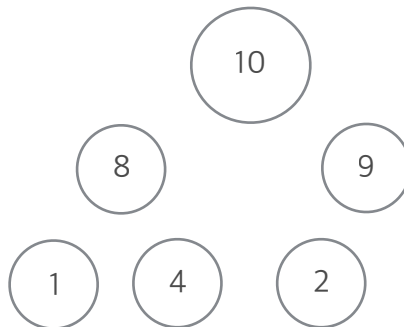


1	2	3	4	5	6	7
8	1	2	9			



1	2	3	4	5	6	7
9	8	2	1			

이런 이동이 됩니다. 이제 남은 4, 10의 입력은 어떤꼴이 될지 예상이 되겠지요?

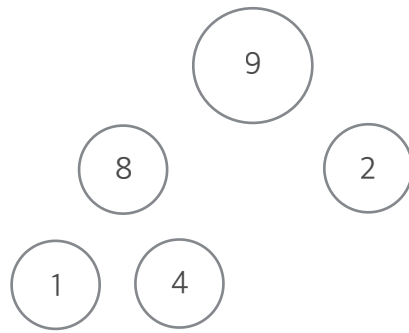


1	2	3	4	5	6	7
10	8	9	1	4	2	

다음과 같이 됩니다. 4는 8보다 작으니 움직이지 않을테고 10은 위로 쪽 올라가게 됩니다.

삭제는 어떻게 이루어 질까요?

10이 사라지고, 2를 루트로 끌어 올리게 됩니다. 그 다음 2는 9의 위치로 가고 9가 오게 됩니다.



1	2	3	4	5	6	7
10	8	9	1	4		

이와 같이 됩니다. 만약 8이 10이었다면 이야기가 왼쪽 자식으로 간 다음 4와 바뀌서 5번째 위치에 가 있겠죠?

힙에 대한 설명은 여기서 마치겠습니다.