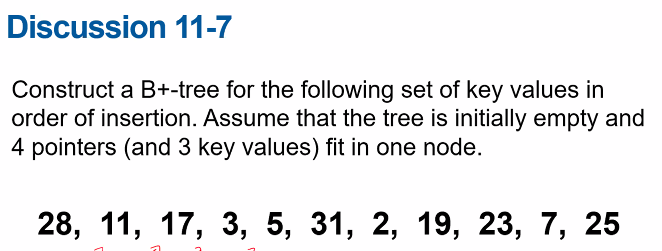
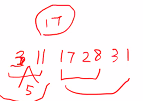
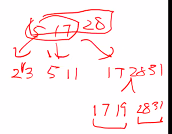
**Discussion 05/23**

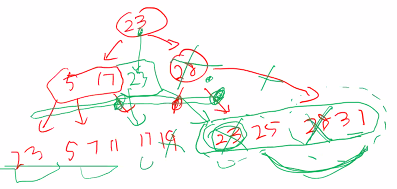


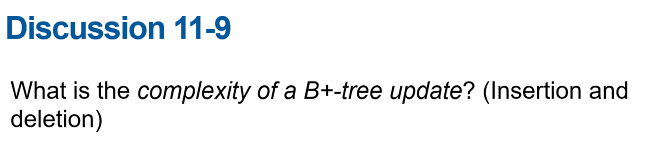
=>  =>  =>  =>



**Discussion 11-8**

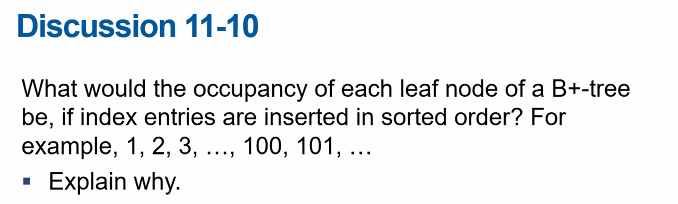
28, 31, 19, 23을 삭제한 뒤의 트리 – 28을 삭제하면 23 25 31이 합쳐지고 23이 위로 올라감. / 31을 삭제하면 그냥 삭제 / 19를 삭제하면 5 7과 11 17이 분배되고, 상위 노드가 5 11이 됨. / 23을 삭제하면 해당 노드가 25 31이 되고 상위 노드가 25가 됨.

=> 

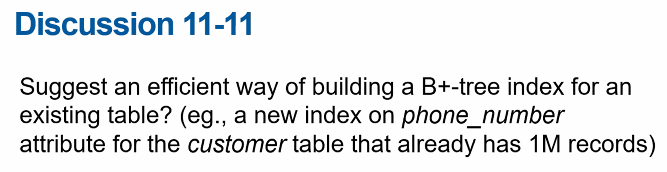


Leaf node까지 찾아가는 데 O(logn)이고 스플릿을 할 경우 두 노드의 value의 개수가 k개라 하면 O(k). 루트까지 전파한다 해도 이 과정을 logn번 반복하니까 O(log\_(k/2)n) + O(klog\_(k/2)n) 그래서 best는 O(log\_(k/2)n), worst는 O(klog\_(k/2)n).

Delete도 마찬가지일 듯?



맨 오른쪽 노드를 제외하고 모두 반씩 차있을 것. 왜냐면 정렬된 순서대로 들어오면 채워지다가 다 차면 반으로split되고가 반복인데 split 된 노드에 value가 들어올 일이 없음.



phone\_number를 기준으로 정렬한 다음에..?

=> 정렬한 다음에 꽉꽉 채워서 leaf node를 만들고 그 leaf node들로 또 상위 노드를 만들고 이런 식으로 하면 된다고 한다.