

Trees

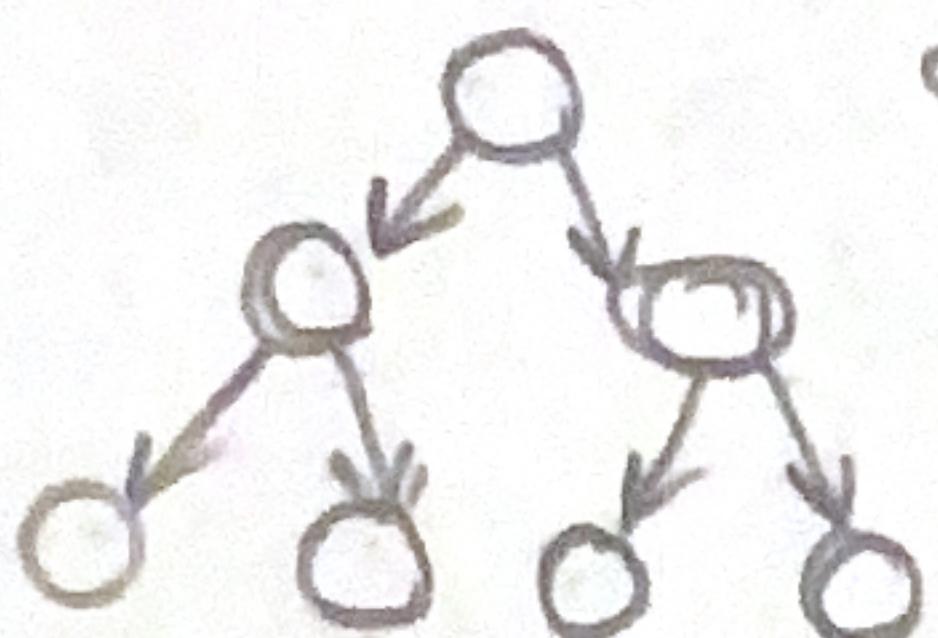
Binary Tree



D	$\frac{N}{2}$
0	1
1	3
2	7
3	15
4	31
5	63
6	127

depth nodes

Complete Binary Tree

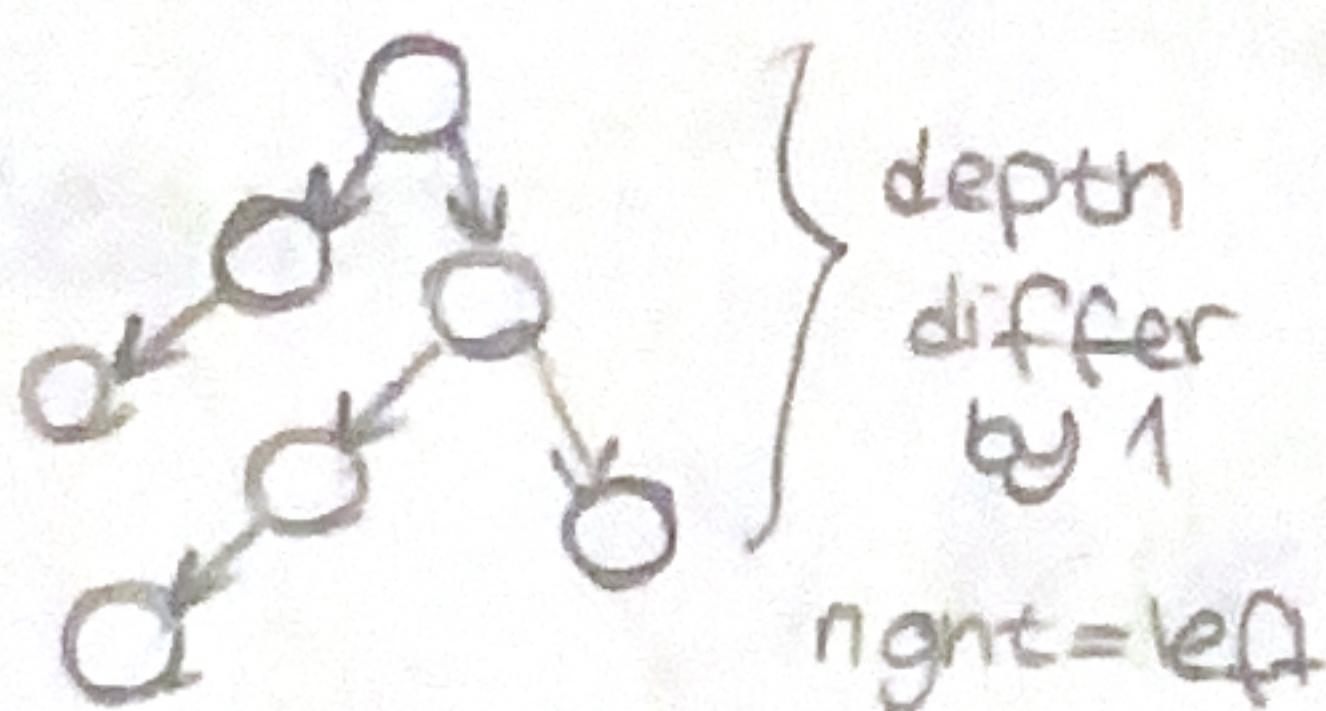


$$\text{depth} = \log_2(N+1) - 1$$

$$\text{vertices} = 2^{\lfloor \frac{d+1}{2} \rfloor} - 1$$

vertices
depth

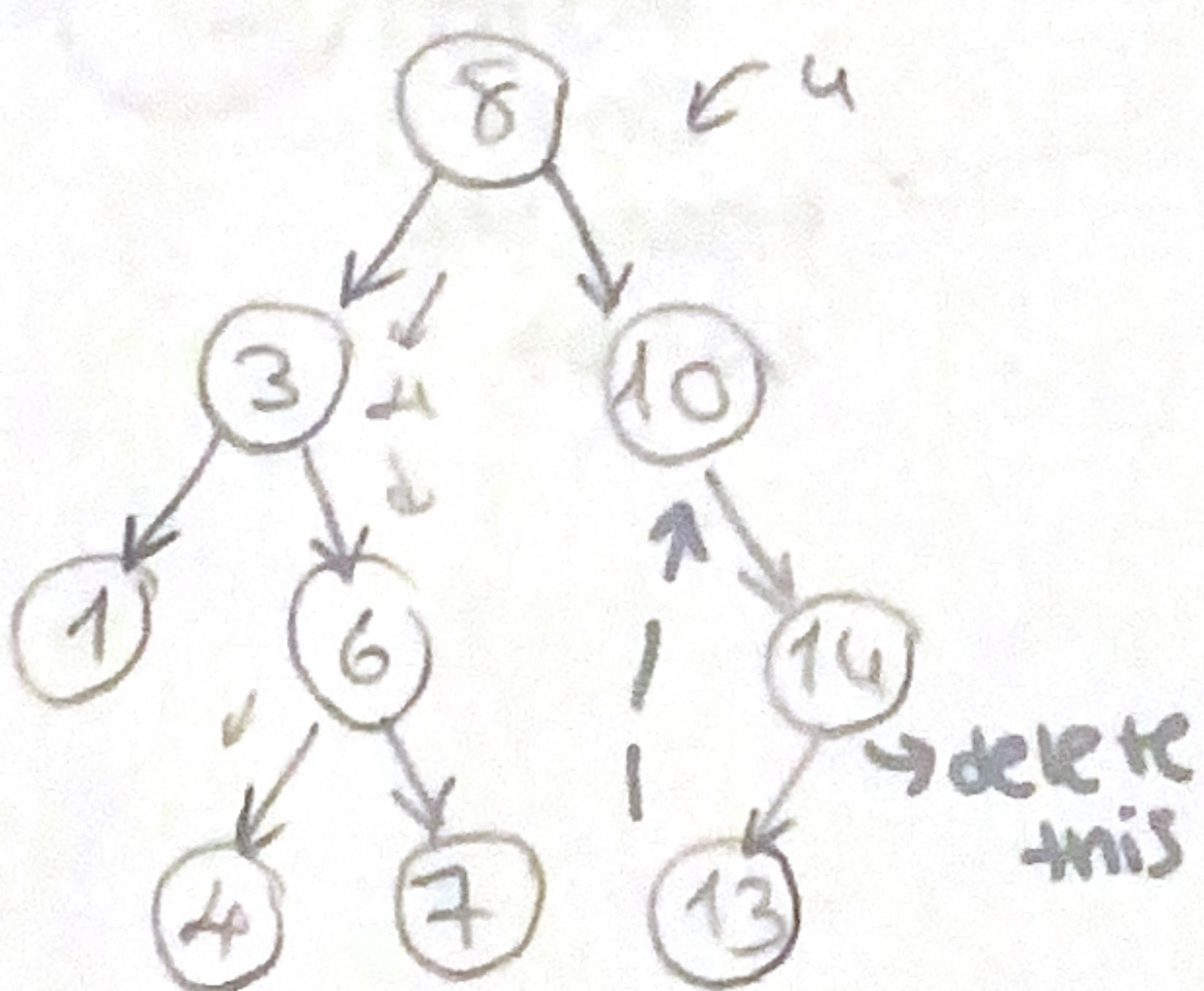
Balanced Tree



Binary Search Trees

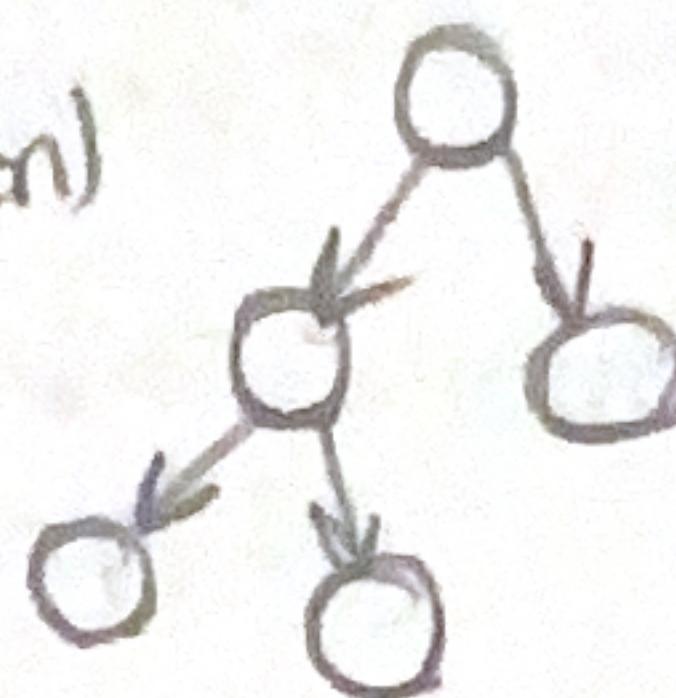
Left subtree < Node key

Node key < Right subtree

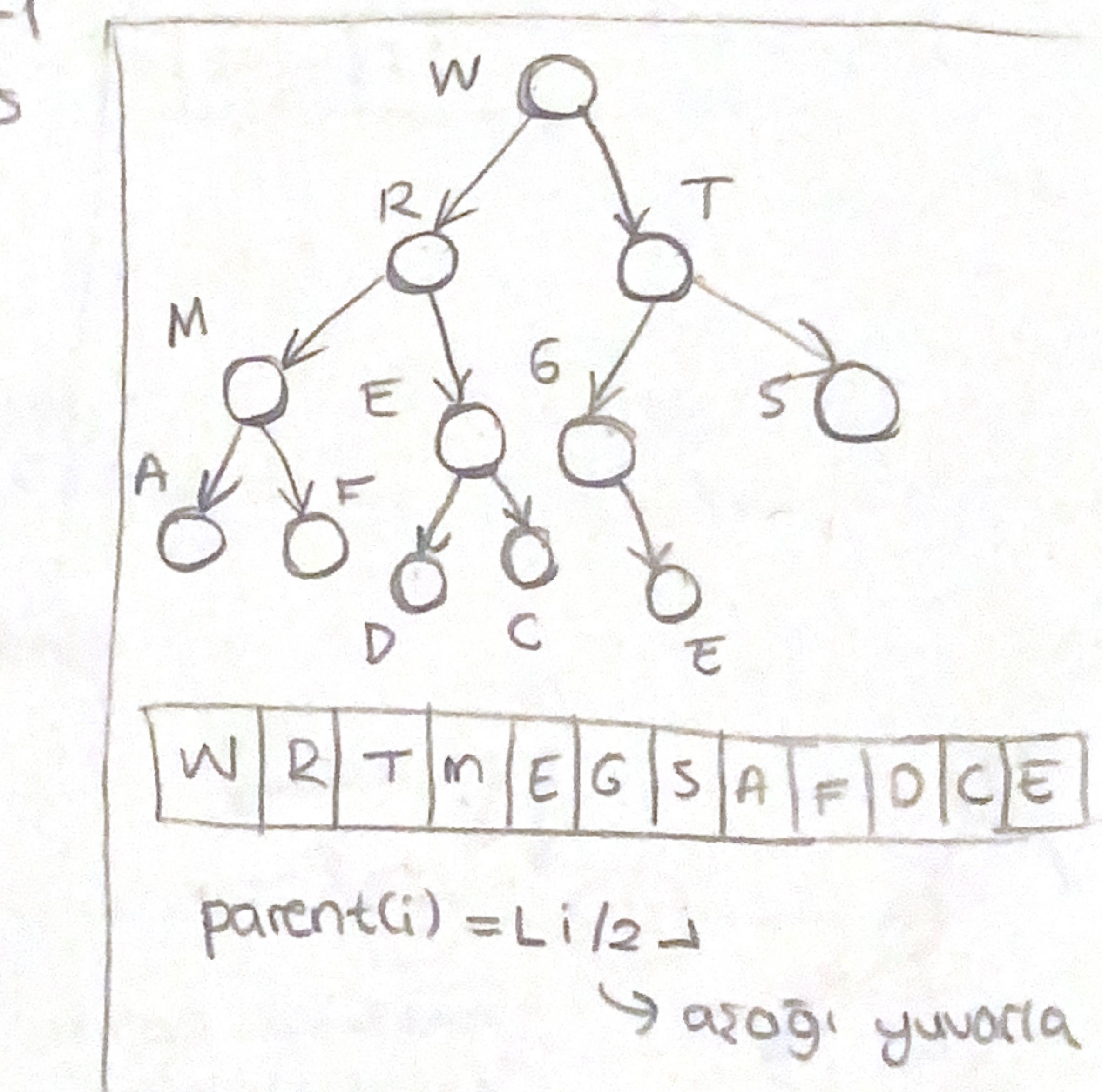


Almost complete tree üzerinde
deletion'da insertion'da
ve sondakine uygun

Strictly Binary Tree



0 or 2 child



Insertion in BST

1. Test key to be inserted against root. If it's smaller, insert it on the left; otherwise (≥) insert it on the right.
2. Search until subtree is empty. (Link to a new node with respective parent)

Deletion

If leaf, disconnect.

If t has a left child but no right child, connect parent to that child.
Otherwise find t's LNR successor