2-4 Trees

Goal

Keep sorted lists subject to the following operations:

find(x,L)

insert(x,L)

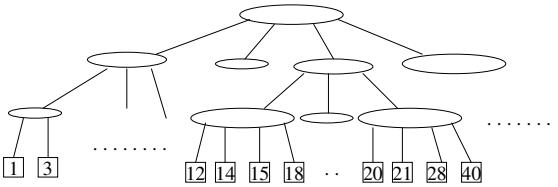
delete(x,L)

catenate(L1,L2) : Assumes that all items in L2 are greater than all items in L1.

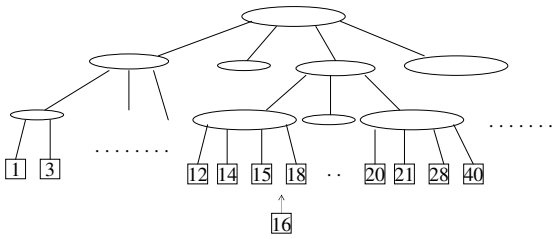
split(x,L) : returns two lists one with all item less than or equal to x and the other with all items greater than x.

2-4 trees

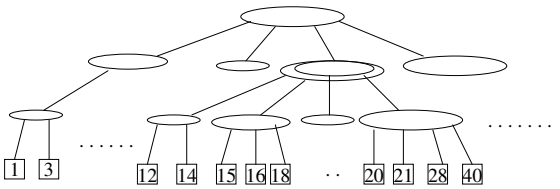
- Items are at the leaves.
- All leaves are at the same distance from the root.
- Every internal node has 2, 3, or 4 sons.



Insert

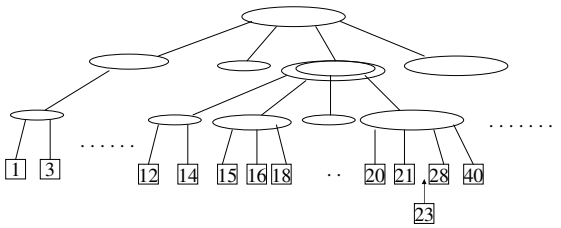


Insert (cont)



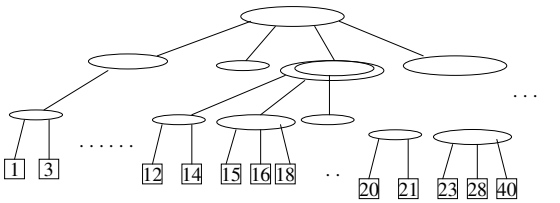
5

Insert (cont)



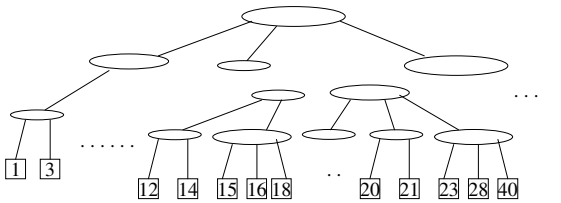
6

Insert (cont)



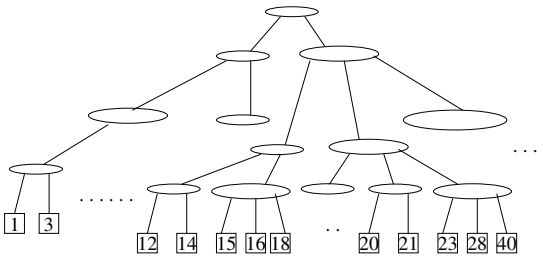
7

Insert (cont)



8

Insert (cont)



9

Insert -- definition

Add the new leaf in its position. Say under a node v.

(*) If the degree of v is 5 split v into a 2-node u and a 3-node w.

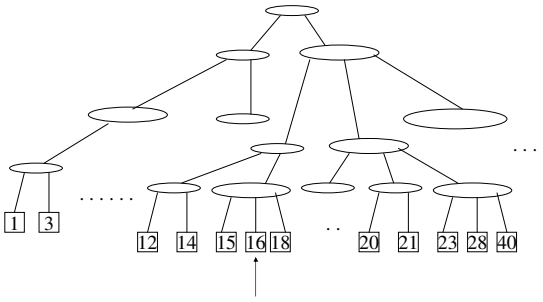
If v was the root then create a new root r parent of u and w and stop.

Replace v by u and w as children of p(v).

Repeat (*) for v := p(v).

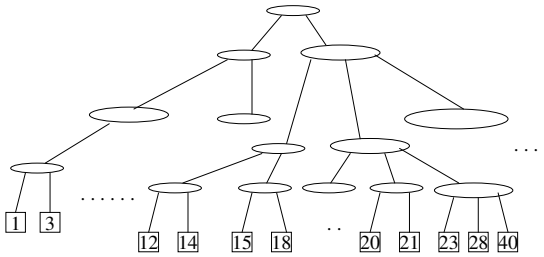
10

Delete



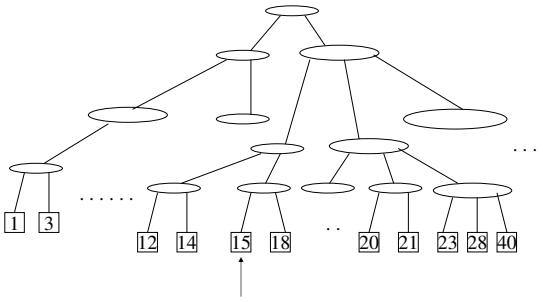
11

Delete



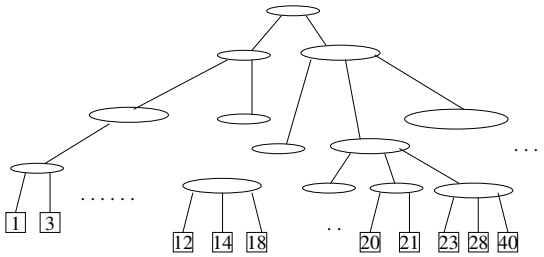
12

Delete



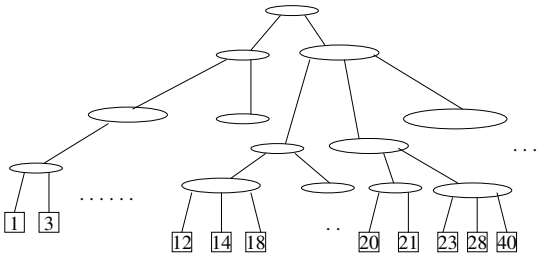
13

Delete



14

Delete



15

Delete -- definition

Remove the leaf. Let v be the parent of the removed leaf.

(*) If the degree of v is one, and v is the root discard v.

Otherwise (v is not a root), if v has a sibling w of degree 3 or 4, borrow a child from w to v and terminate.

Otherwise, fuse v with its sibling to a degree 3 node and repeat (*) with the parent of v.

16

Summary

Delete and insert take O(log n) on the worst case.

Theorem (homework) : A sequence of m delete and insert operations on an initial tree with n nodes takes O(m+n) time

Can do catenate and split in a way similar to red-black trees.

17