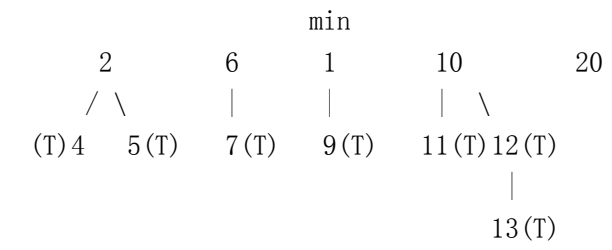
1. F-heap (ChildCut: T(true), F(false))

(a)



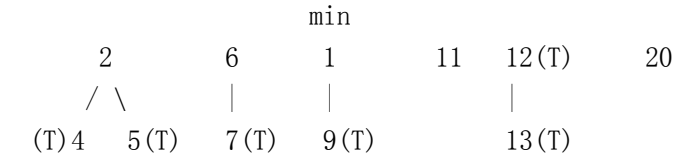
****

(b)

The below solution is right answer for (b).

(6:full points answer)

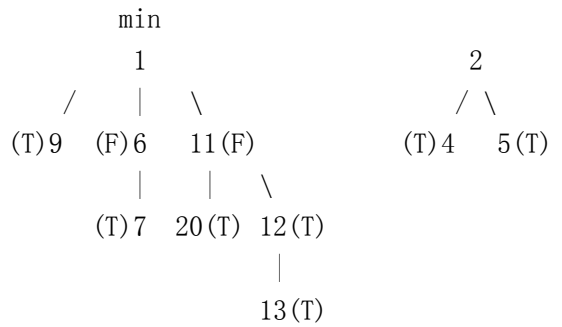
delete operation didn't join(or combine) steps except deletemin operation)



****

the below solution is wrong but I give 4 partial points if you write answer like below.

(*Remember : 10 is not min element in this tree, so you don't need to combine steps*)

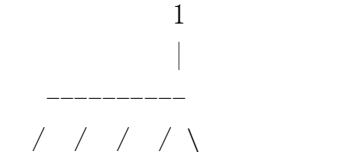


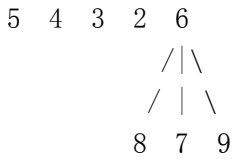
****

2. min 2-pass paring heap

(a)

- meld on every insertion





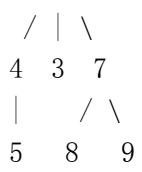
(b)

after pass 1:

after pass 2:

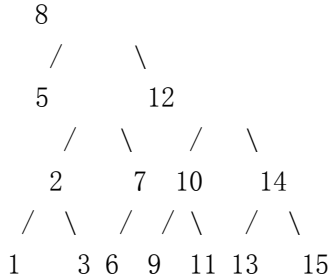
(c)

- remove 6 and merge its children
- insert the merged tree into the list of min tree roots

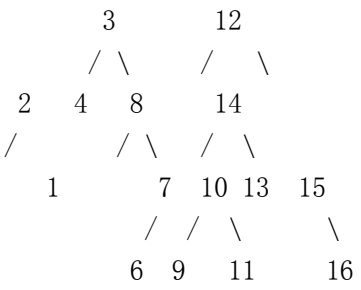


3.

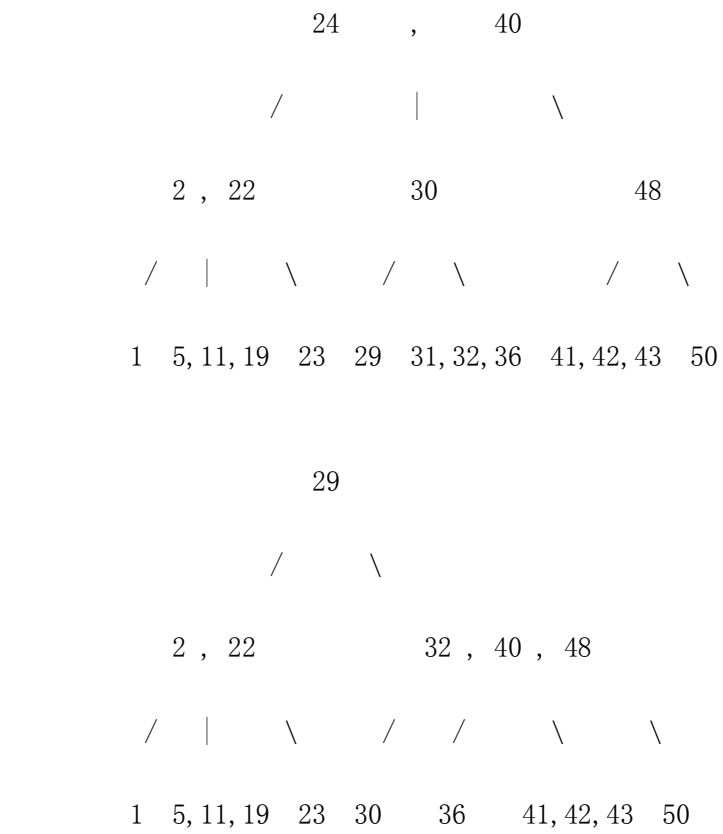
a. Deletion of 4



b. Insertion of 16



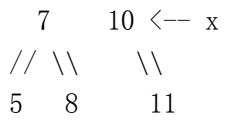
4.



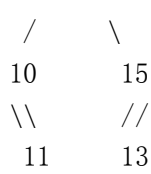
5. red-black tree

(a) Join(S,12,B)

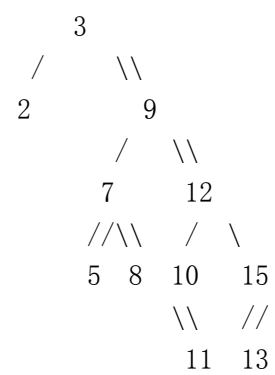
i) follow the right-child pointer until rank(B) == rank(x), where rank(B) == 1, x is a node pointer of tree S.



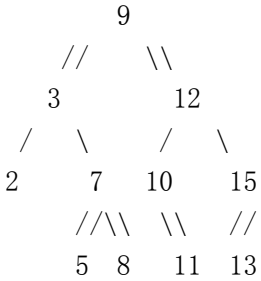
ii) combine subtree x, 12, and tree B



iii) connect the combined tree to node 7 through red pointer.



iv) perform RR rotation to remove consecutive red pointers.

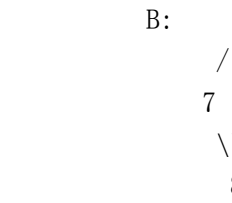


(b) Split(5,S,x,B) for tree S.

i) find node 5 and copy node value to x.
Since node 5 has left and right child,
Init:: tree S is NULL(left subtree) and
tree B is NULL (right subtree).

ii) perform Join(B, 7, NULL).

iii) Join(B, 9, right-subtree of node 9).



iv) Join(left-subtree of node 3, 3, S).

So, the result of split operation: