(2)

Start with a collection of external nodes and Each external nodes defines a different tree.

Step1: Reduce number of trees by 1

Select 2 trees with minimum weight.

and Combine them by making them children of a new root node.

The weight of the new tree is sum of the weight of individual trees

Add new tree to tree collection.

Step2: Repart reduce step until only 1 tree remains.

(you can see an example on lecture7 note)

(more detail)

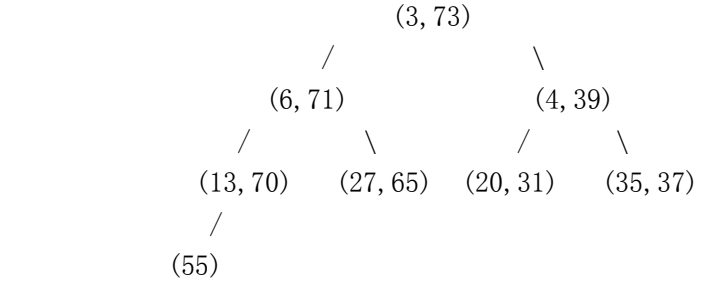
a new tree is created by combining the two trees with minimum weighted external path length which are deleted from the list, and inserted into the list. This new tree must have the smallest external path length since the two trees with minimum weight were used. I.e., no other combination can create a tree with a lower weight.

No matter how many times this process is repeated, the newly created tree has the minimum weight in terms of external path length.

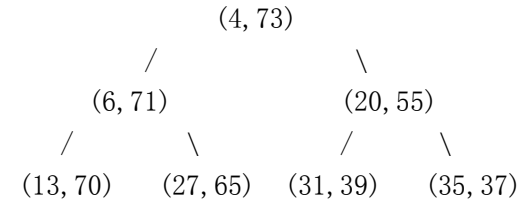
Since the number of trees in the list is decreased by 1 per pass, the combined tree is the one with minimum weighted external path length when the last pass of the for loop combines the remaining two trees.

(3)

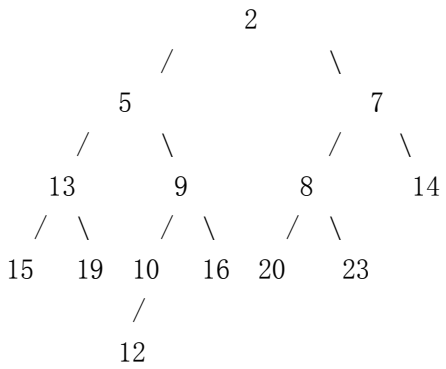
(a)



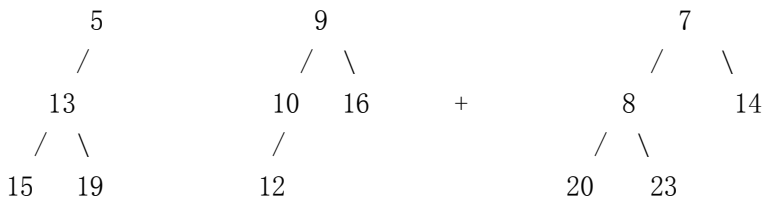
(b)



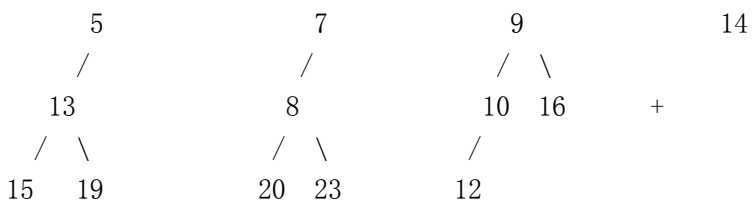
(4) Leftist tree



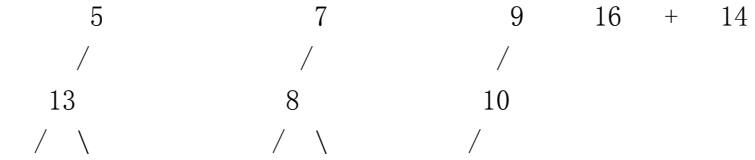
(a) meld right subtree of the tree whose root is minimum



(b) meld right subtree of the tree whose root is minimum

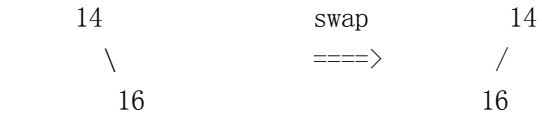


(c) meld right subtree of the tree whose root is minimum

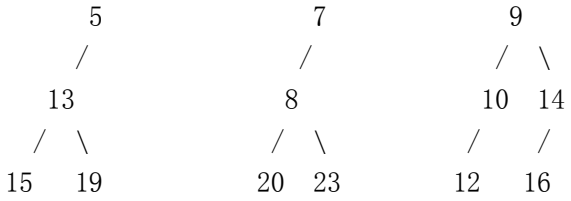


15 19 20 23 12

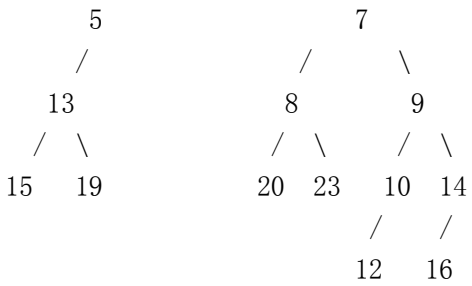
(d) meld two subtrees (do swap if needed)



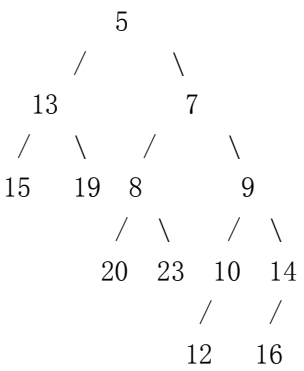
(e) meld two subtrees



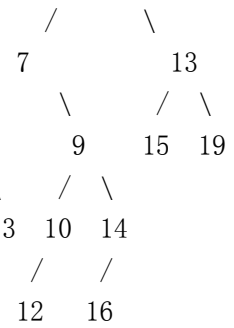
(f) meld two subtrees



(g) meld two subtrees

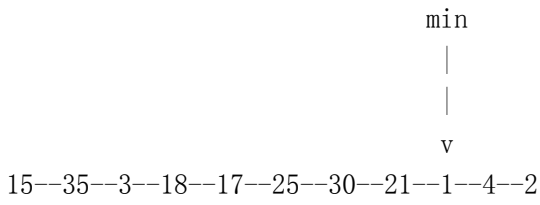


(h) swap

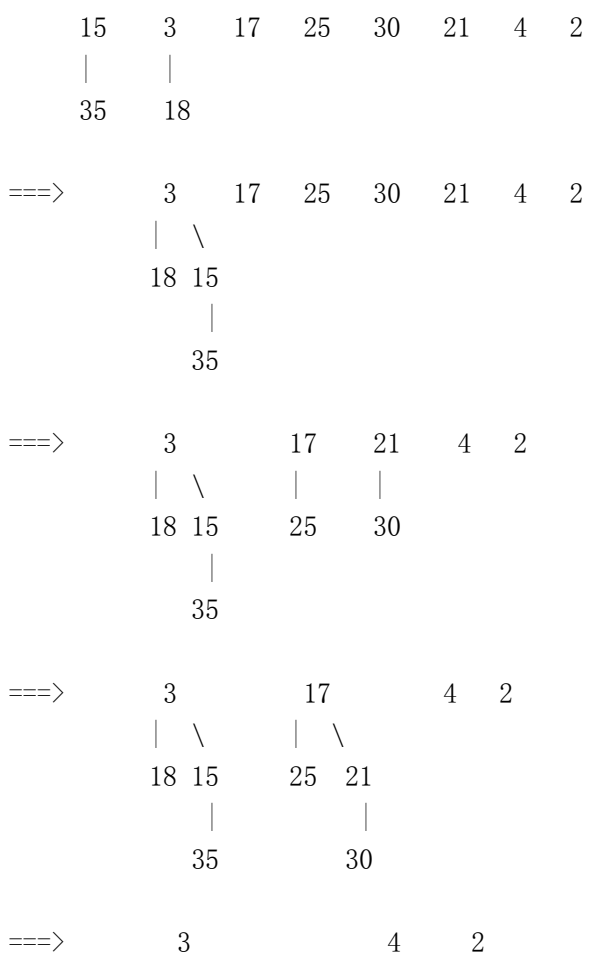


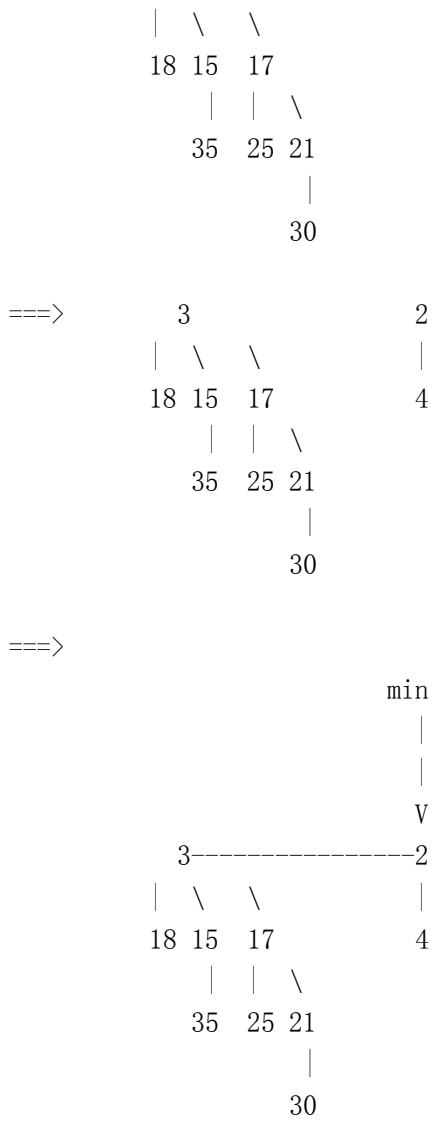
(5)

(a)


min
15--35--3--18--17--25--30--21--1--4--2

(b)


15 3 17 25 30 21 4 2
35 18
===>
3 17 25 30 21 4 2
18 15
35
===>
3 17 21 4 2
18 15 25 30
35
===>
3 17 4 2
18 15 25 21
35 30
===>
3 4 2


18 15 17
35 25 21
30
===>
3
2
18 15 17
4
35 25 21
30
===>
min
3
2
18 15 17
4
35 25 21
30