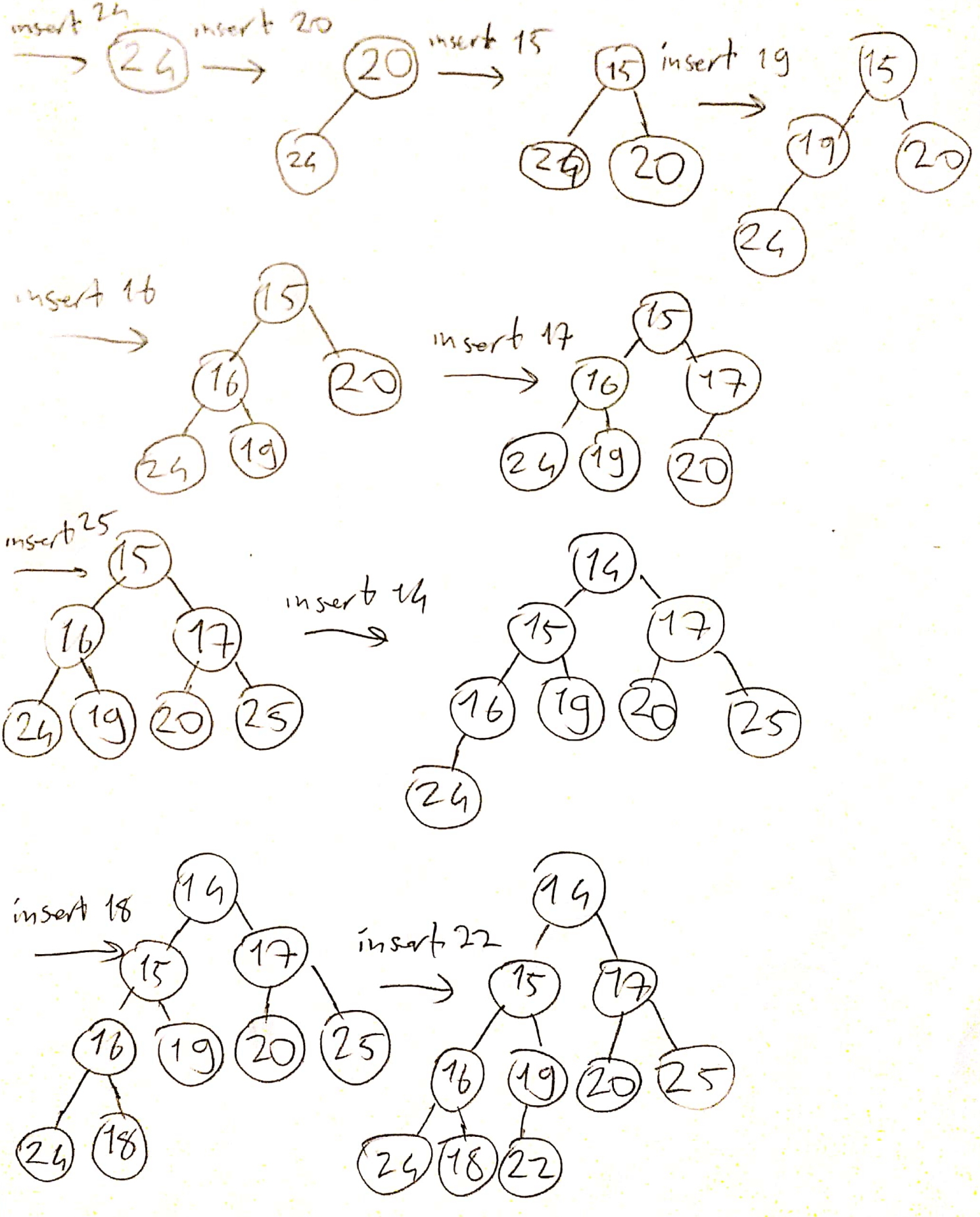
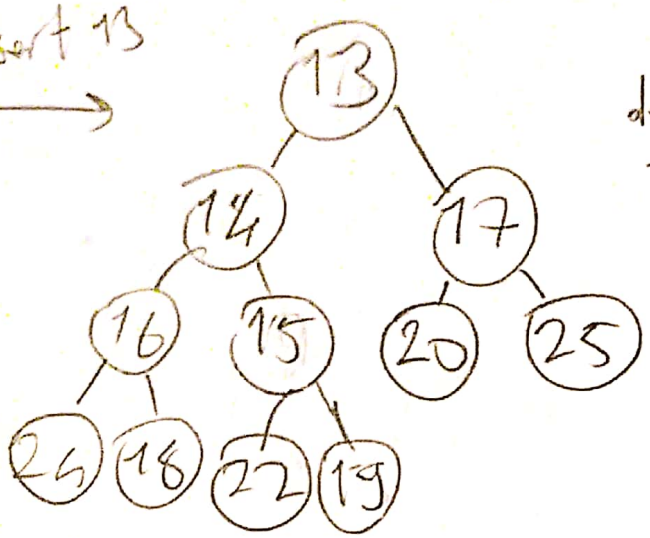
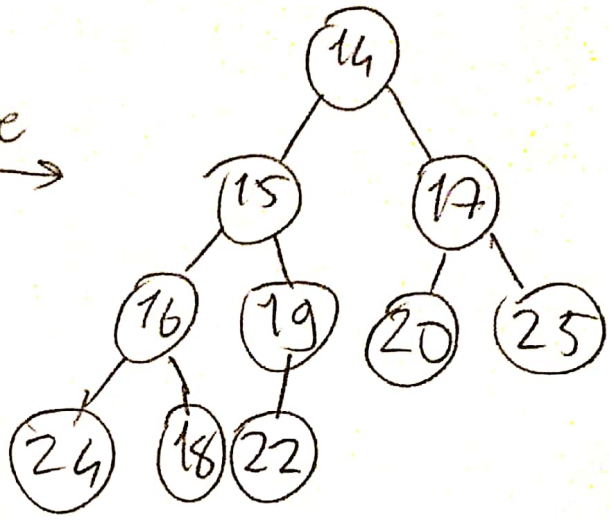
Q1) a)



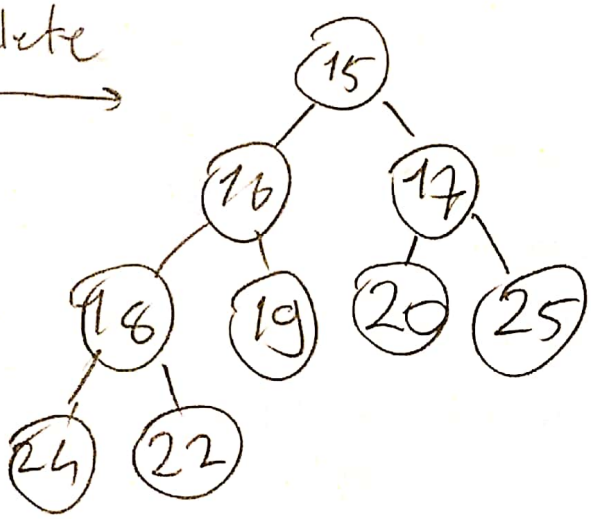
insert 13



delete



delete



b)

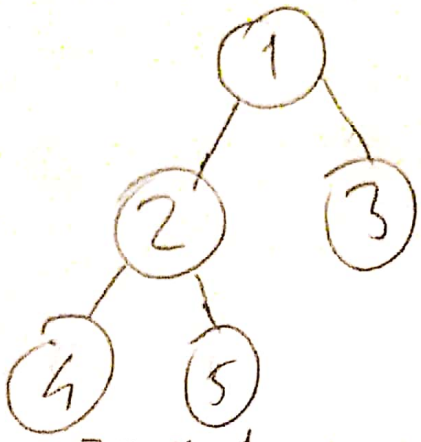


Figure 1

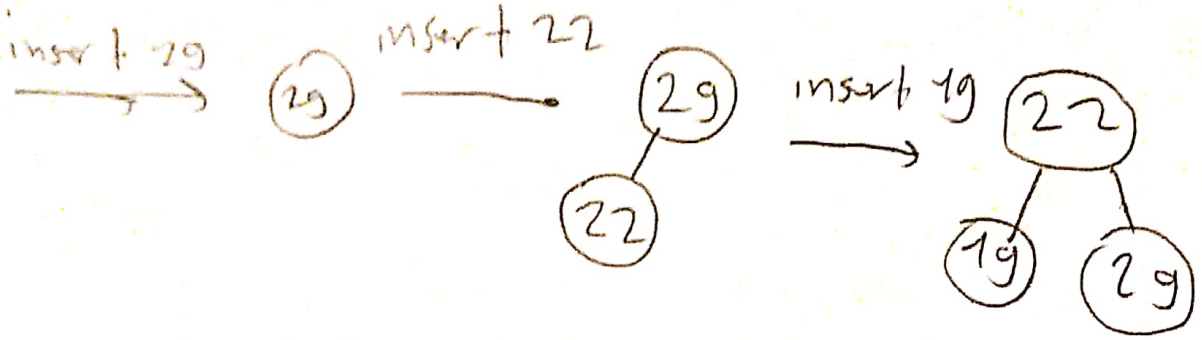
Preorder traversal: 1, 2, 4, 5, 3

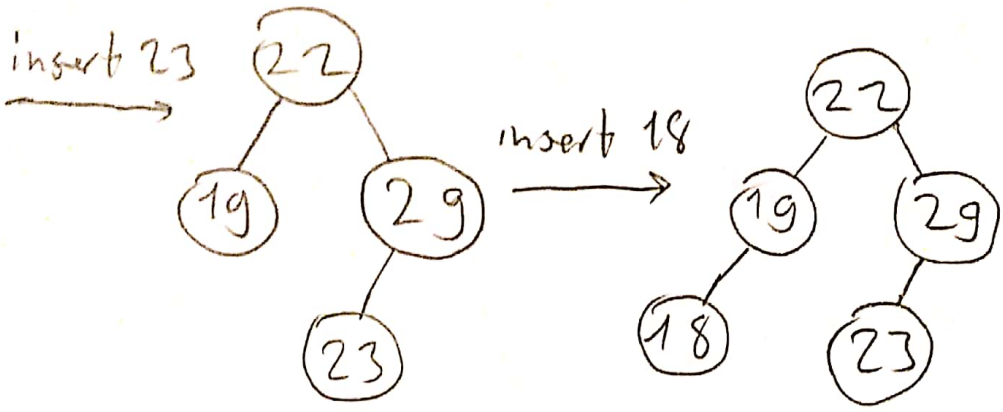
Inorder traversal: 4, 2, 5, 1, 3

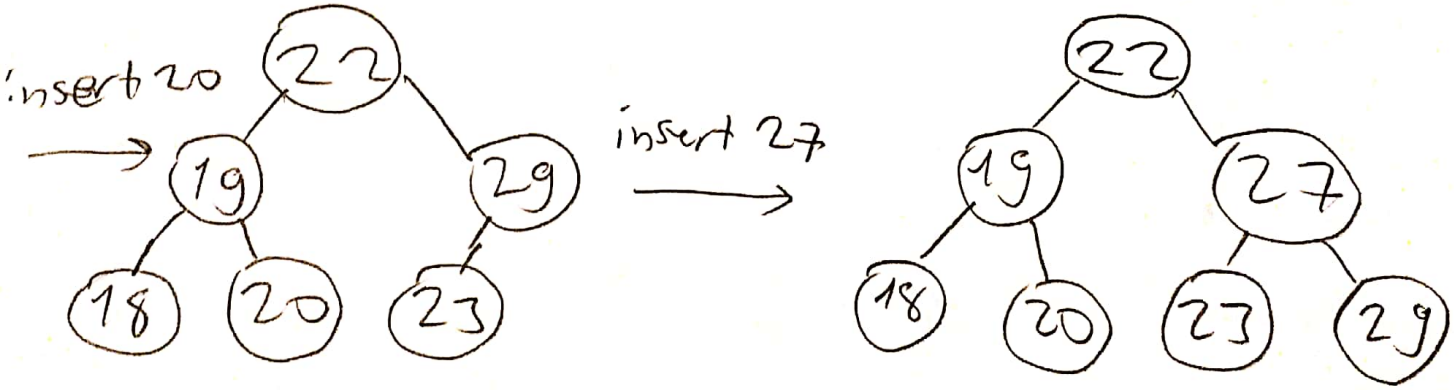
Postorder traversal: 4, 5, 2, 3, 1

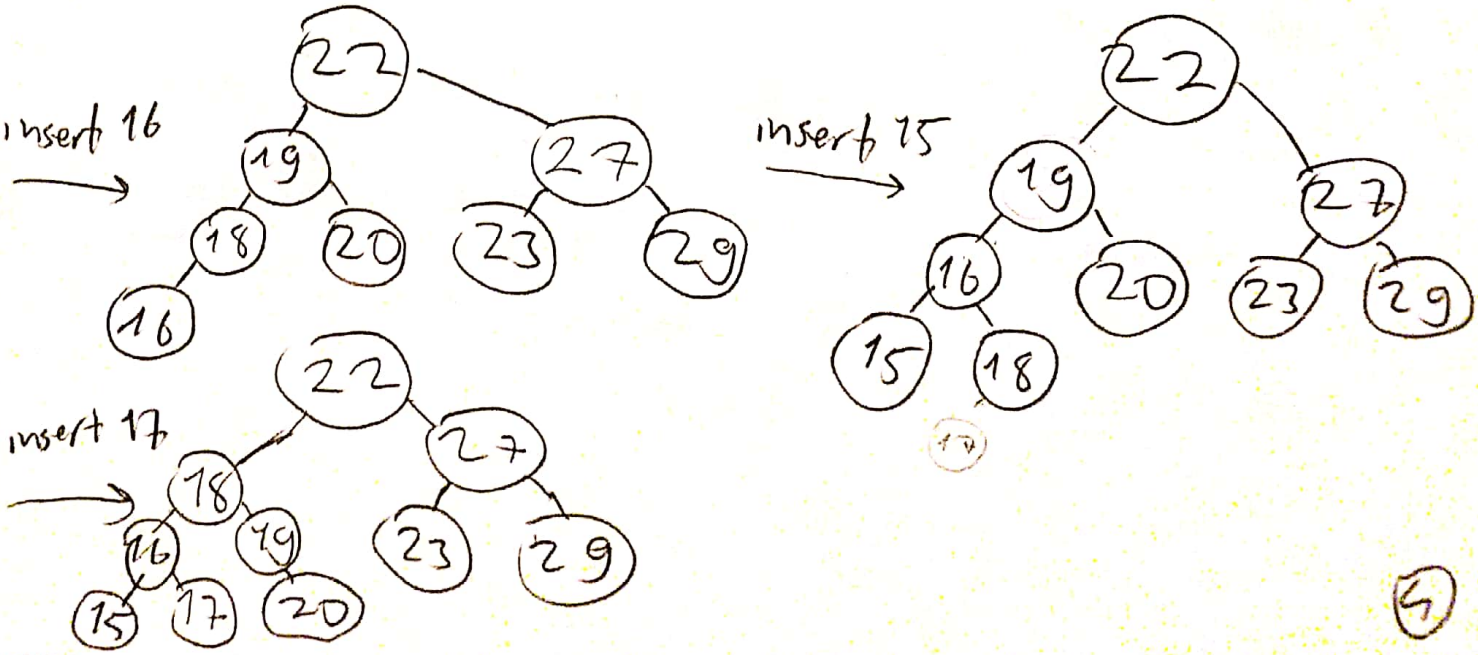
The result of any traversals is not sorted for the given min heap. In a min heap, a parent is always $\leq$ its children but there is no relationship between children. Left can be smaller in a subtree and greater in another one. So this makes impossible to guarantee to obtain a sorted results.

c)









Q3)

Since this can be considered as searching, starting from 1 and increasing t by 1 is similar to linear search. In this case, instead we can use a binary search approach. We can run simulation for $N/2$ printers. If it does not meet requirements, we could do it by increasing N and running for $3N/6$ printers. If $N/2$ satisfies the requirement, we can run the simulation with $N/6$ to find the minimum number that satisfies the requirement.

Also, a further improvement could be considering the difference between the result and requirement. For example, if the result is close to expected, we can run again with closer values and the result is too far from expected, we could increase the difference.