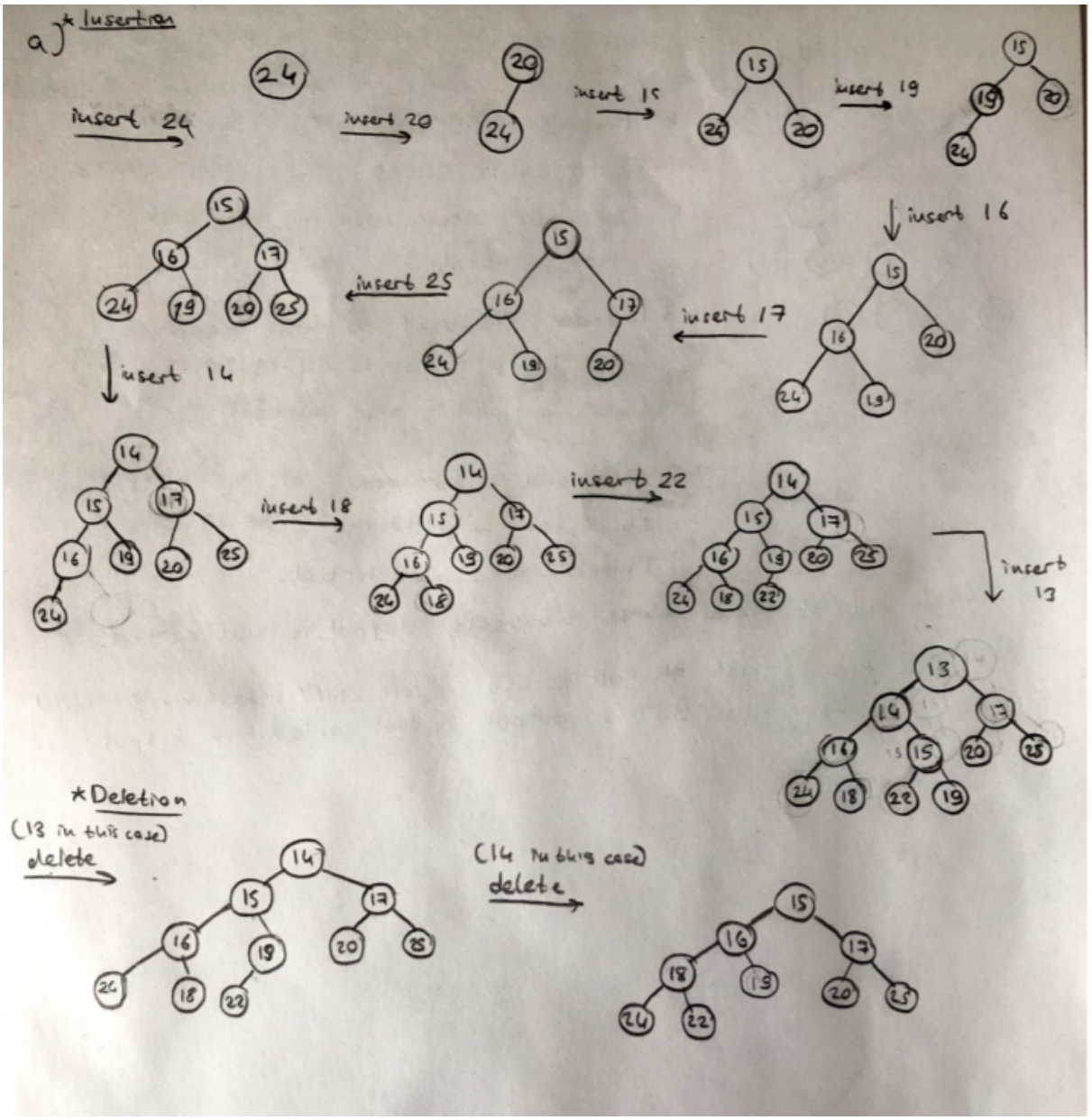
Berk Saltuk Yılmaz - 21903419 - Sec03

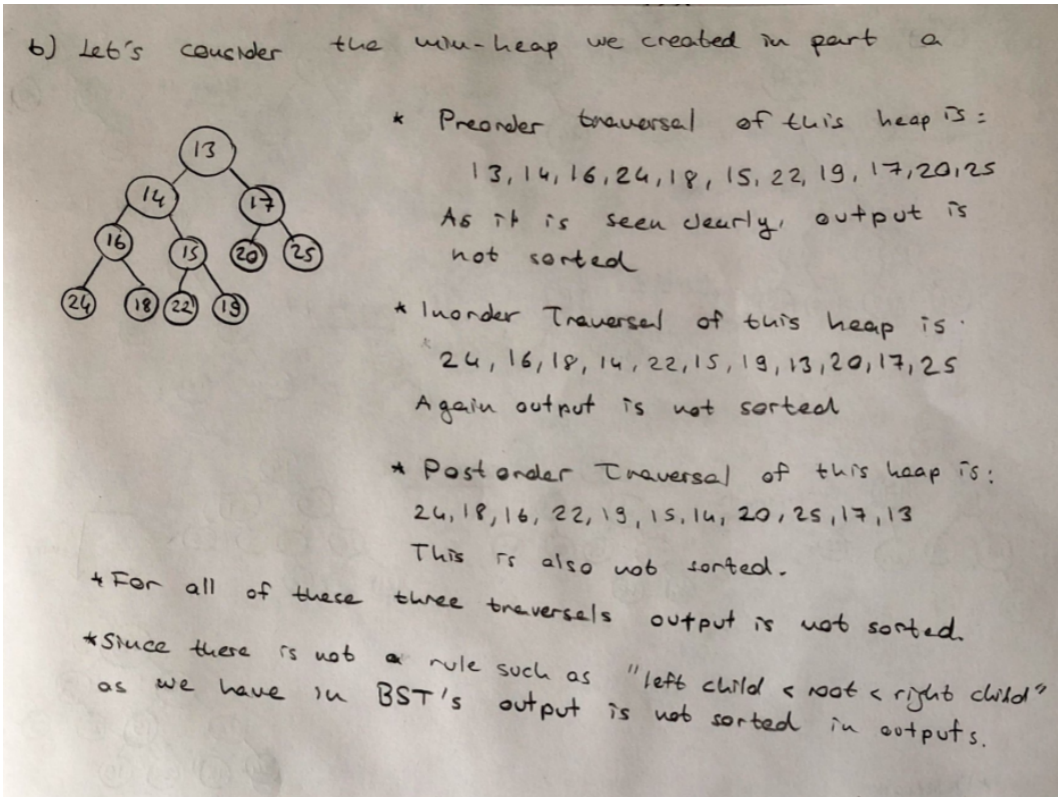
Homework 3: Heaps, Priority Queues, and AVL Trees

Question-1

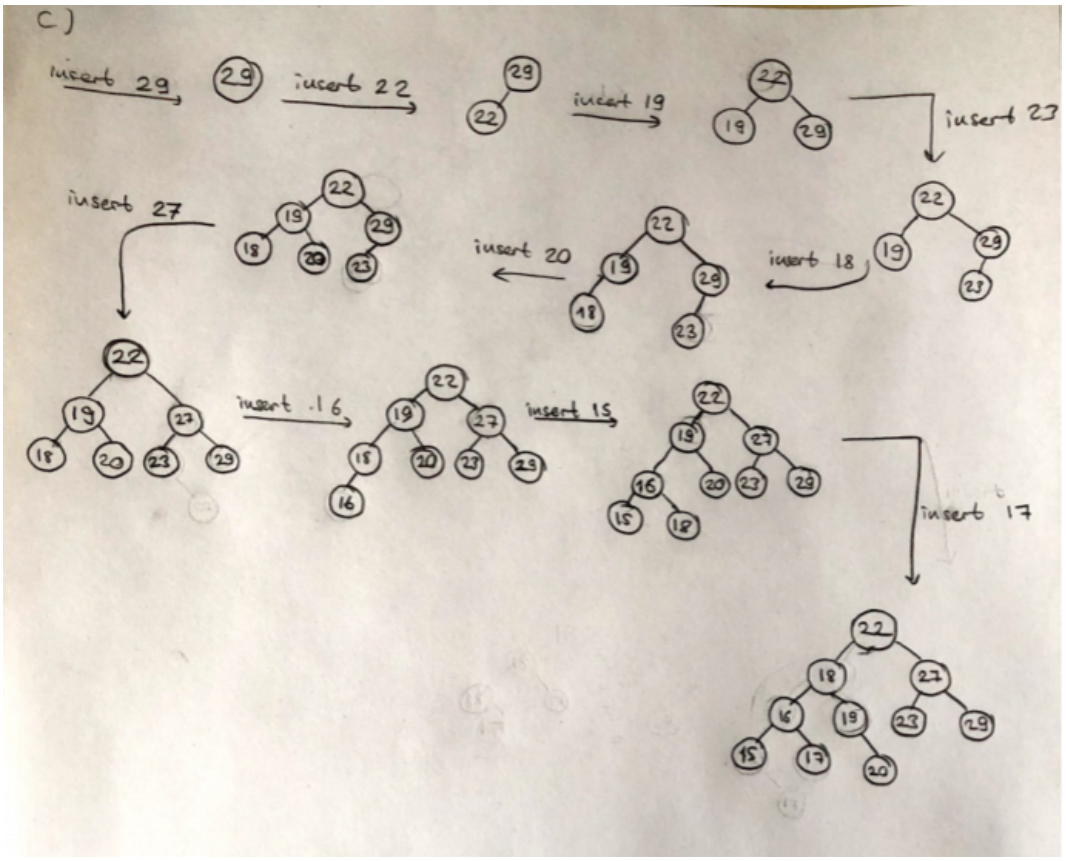
(a):



(b):


b) Let's consider the min-heap we created in part a
13
14
17
16
15
20
25
24
18
22
19
* Preorder traversal of this heap is:
13, 14, 16, 24, 18, 15, 22, 19, 17, 20, 25
As it is seen clearly, output is not sorted
* Inorder Traversal of this heap is:
24, 16, 18, 14, 22, 15, 19, 13, 20, 17, 25
Again output is not sorted
* Postorder Traversal of this heap is:
24, 18, 16, 22, 19, 15, 14, 20, 25, 17, 13
This is also not sorted.
* For all of these three traversals output is not sorted.
* Since there is not a rule such as "left child < root < right child" as we have in BST's output is not sorted in outputs.

(c):


C)
insert 29
insert 22
insert 19
insert 23
insert 27
insert 20
insert 18
insert 16
insert 15
insert 17

Question-3

Now suppose that your simulation was to be run for a very large library with many potential printers (N) and many, many more print requests. Would it still be a good idea to try the simulation starting from 1 printer and increasing until you found the right number $K \leq N$? What is a better strategy for finding the optimum number of printers in such a case?

Answer:

In such a case, it is not the brightest idea to start the printer number from one and increase it by one each time. It is better to make the increasing amount greater; for example, in the example log file, we just have ten print requests and four printers needed. If the request were 1,000,000 for the same average time, the printer number would be 400,000 by very rough basic proportionality, it would take lots of time to find the correct number of printers for the desired average time if we would go one by one. If the amount of increase was 1000 in this situation, we would find the correct number of printers 1000 times faster. However, choosing an arbitrary amount of increase is not a bright idea either. By collecting the number of printers needed for several scenarios, we may find a strategy to decide on the amount of increase such as "... requests needs … printers for … minutes average time". As a penalty of greater steps, we might exceed the optimum printer number mistakenly. So, when we first find an average time lower than desired, we should go back by an amount lesser than the original (for example half of it). This would make us lose time; however, it would probably take less than increasing the printer number one by one.