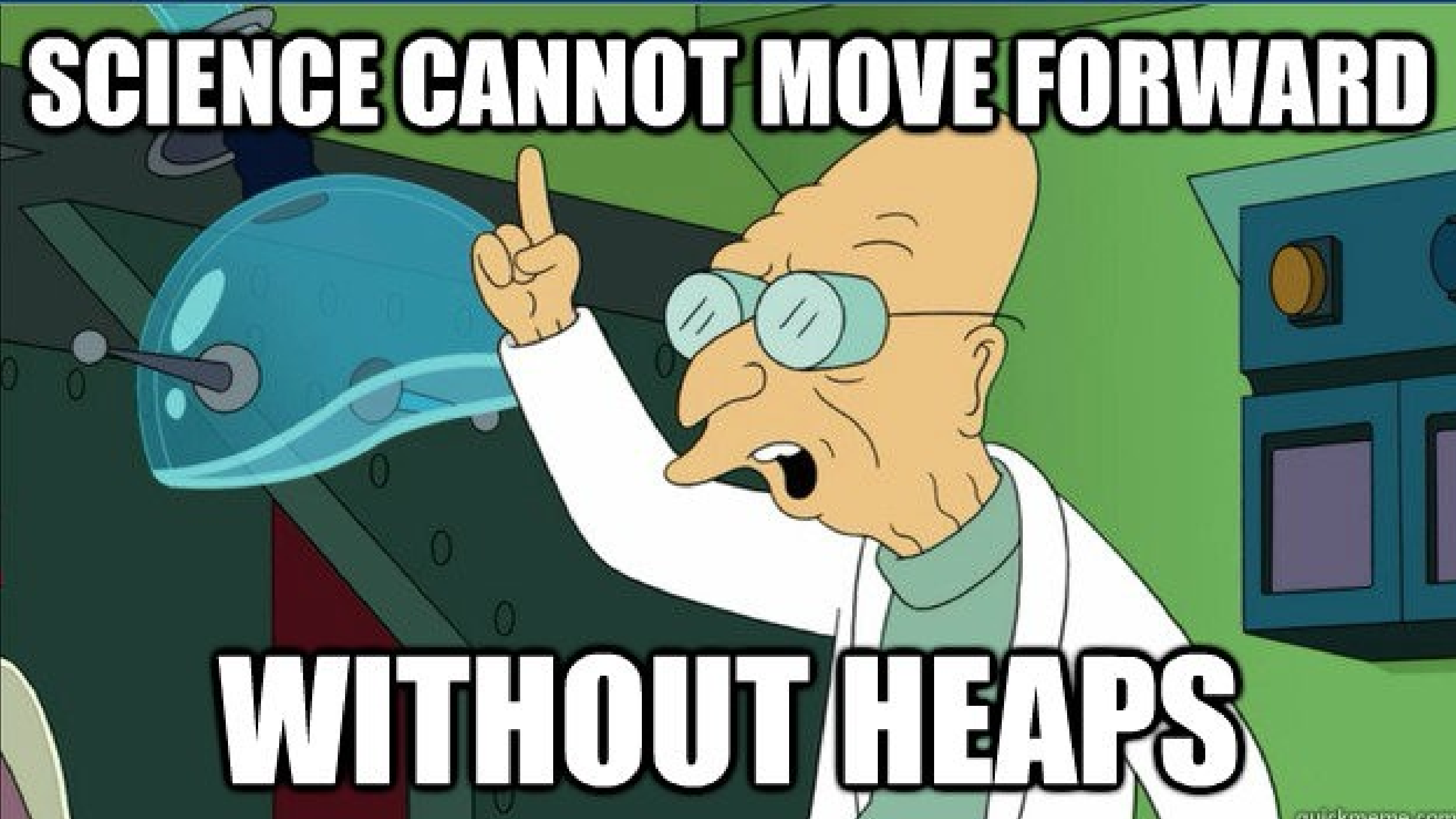
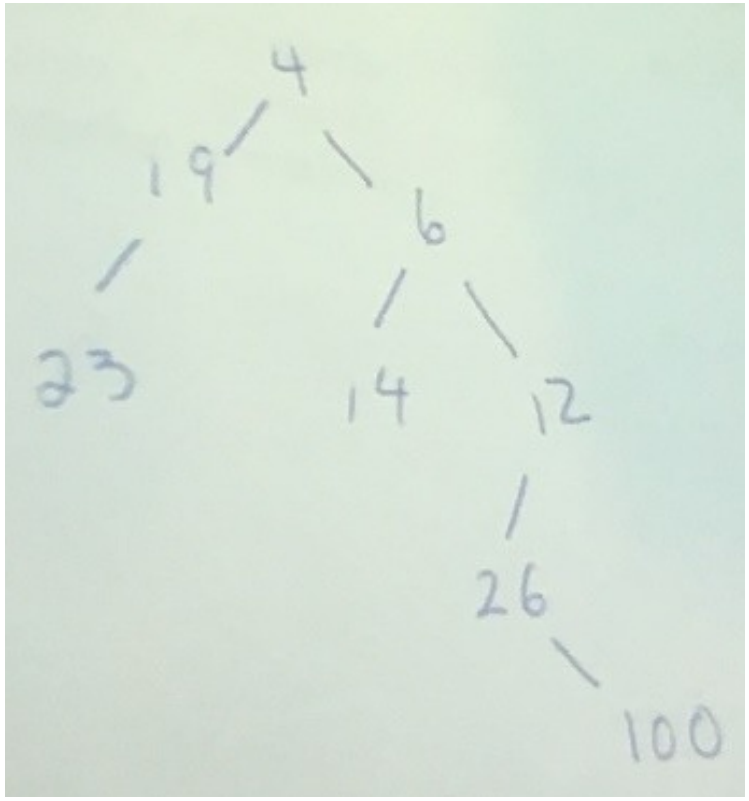

SCIENCE CANNOT MOVE FORWARD
WITHOUT HEAPS

Heaps

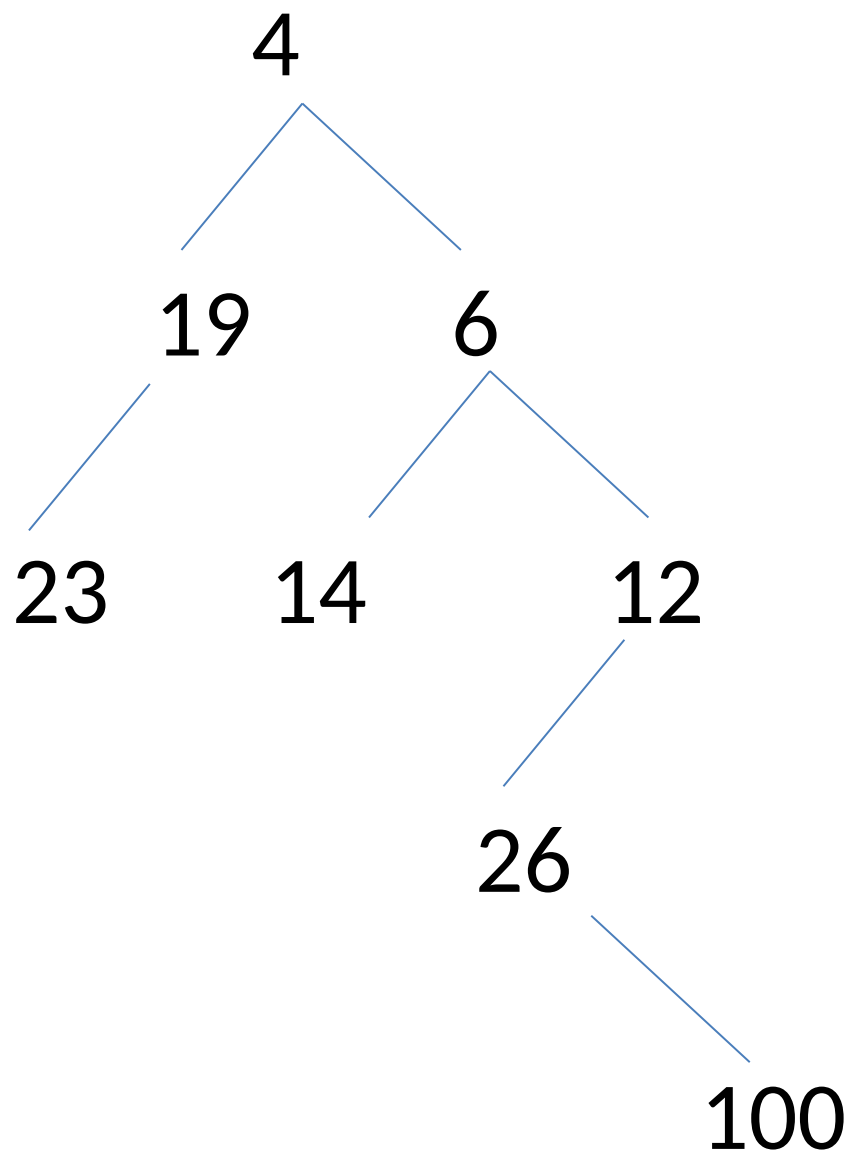
A heap is a binary tree (NOT a BST) such that the smallest item is the root of the tree and the left and right subtrees are heaps.

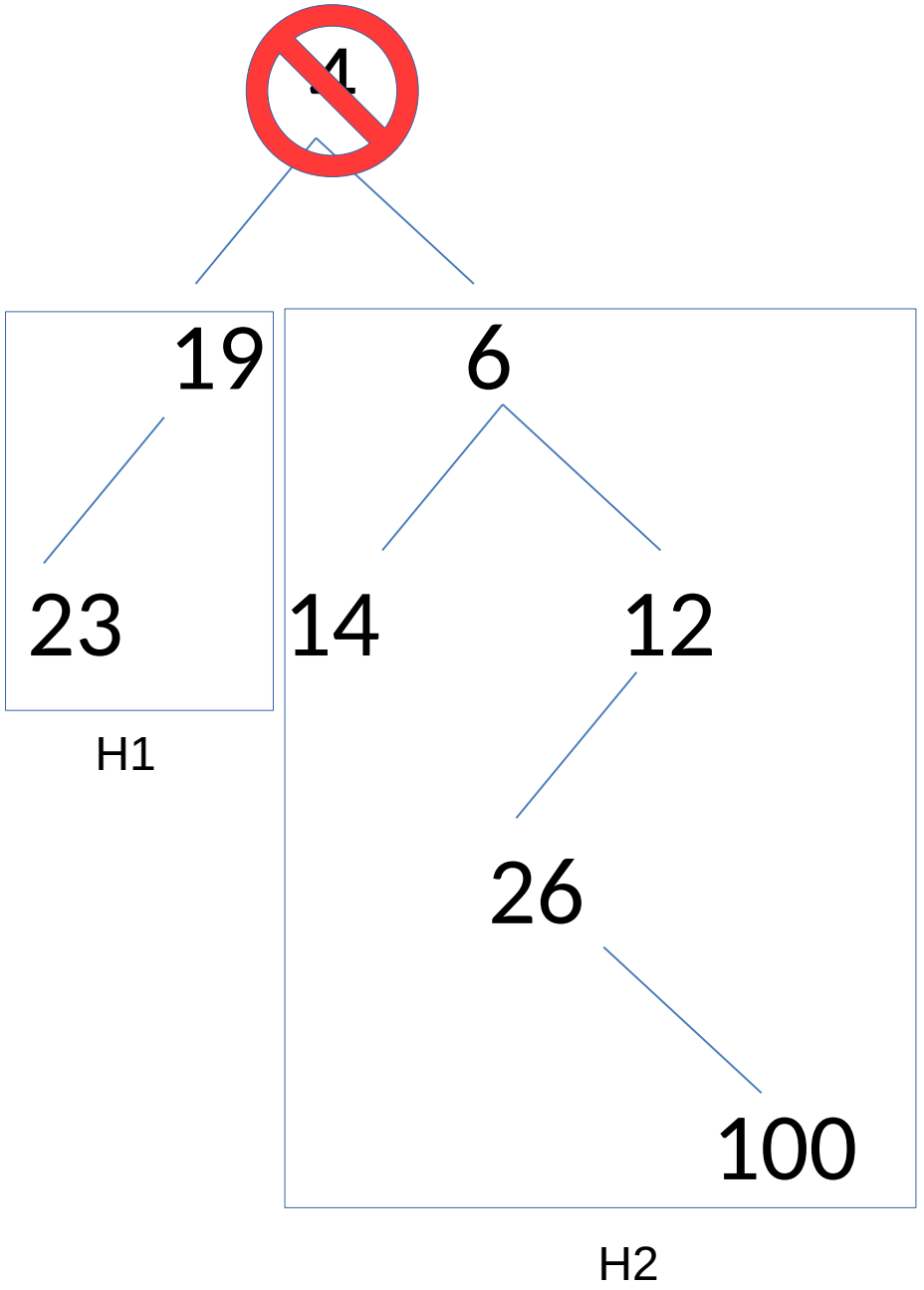


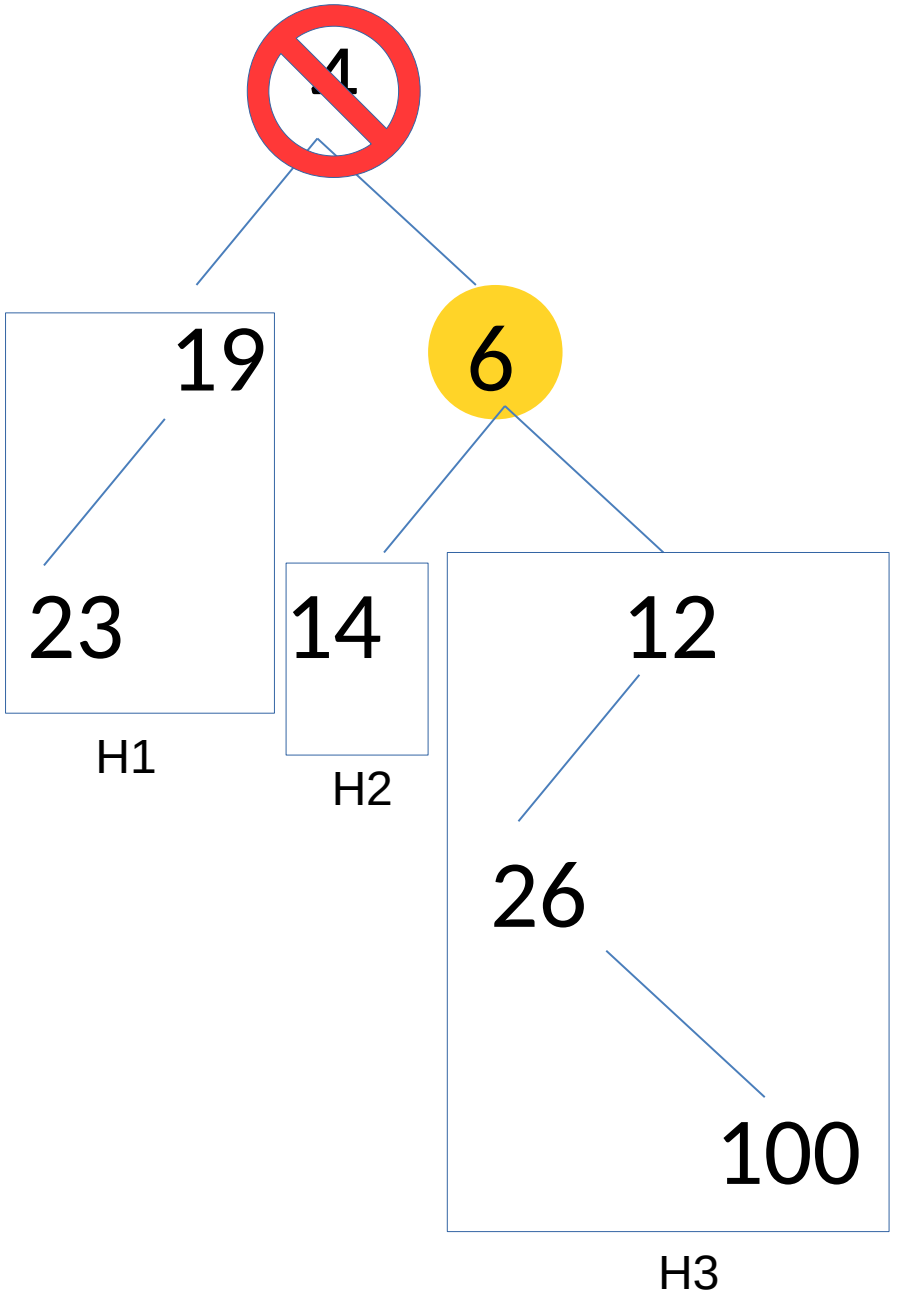
ADT Priority Queue

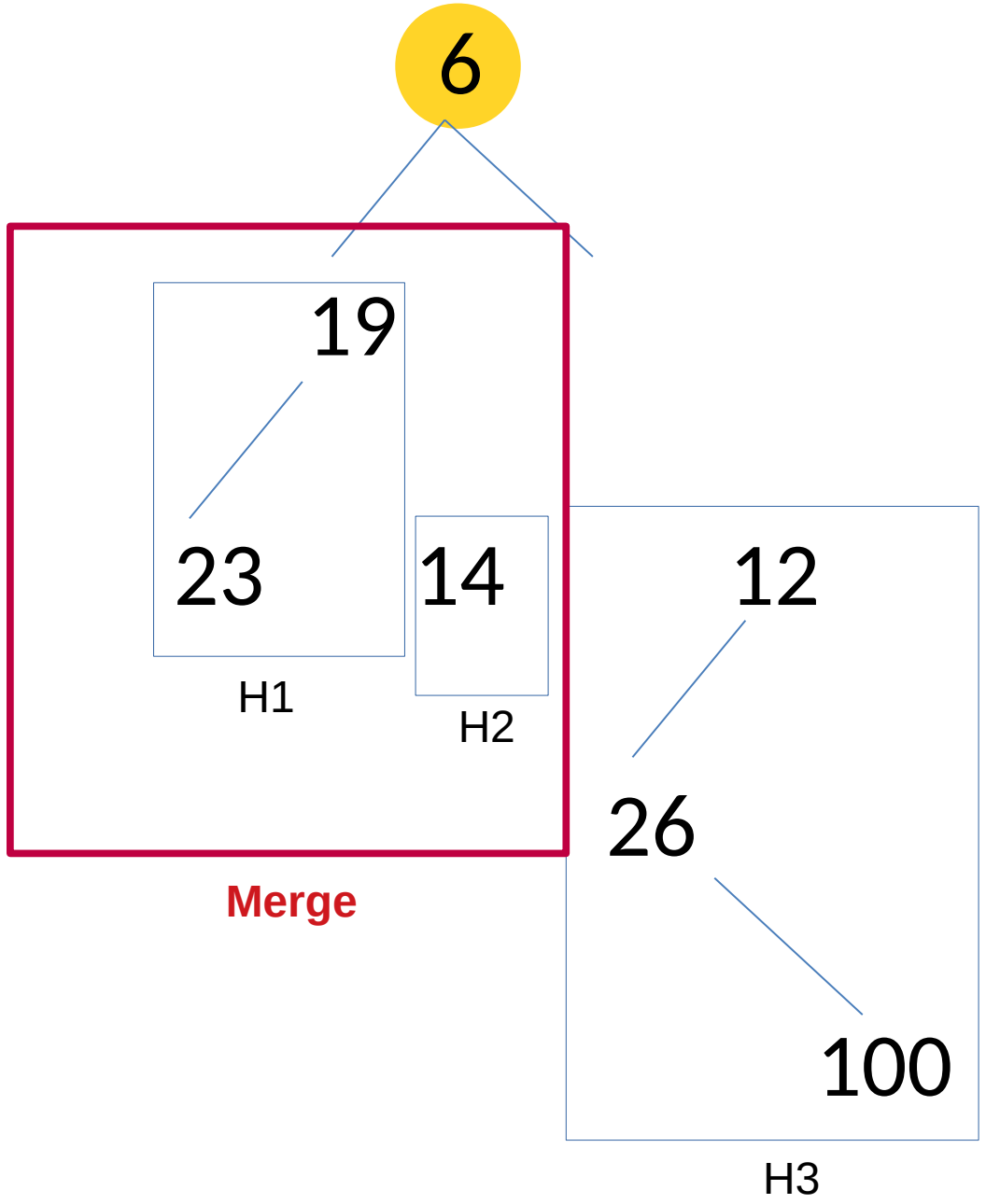
- The ADT Priority Queue maintains a collection of elements with an accessing function that produces the highest priority elemenet
  - addElt: PQ elt -> PQ
  - minElt: PQ -> elt
  - remMinElt: PQ -> PQ

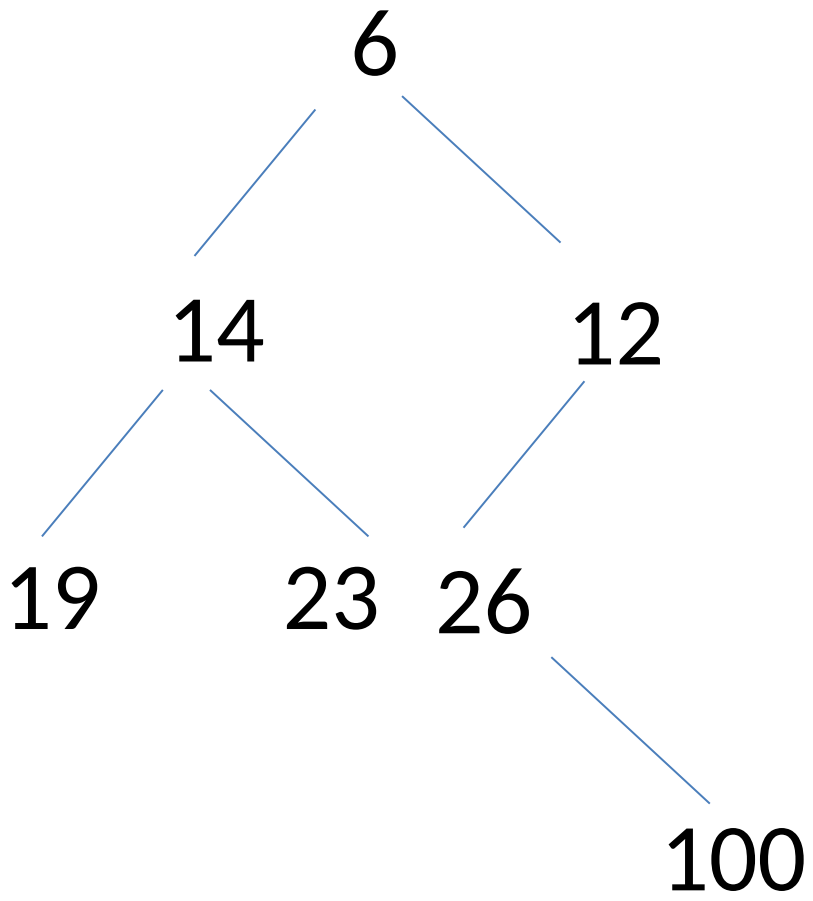
addElt	O(n)
minElt	O(1)


4
19
6
23
14
12
26
100


1
19
23
H1
6
14
12
26
100
H2

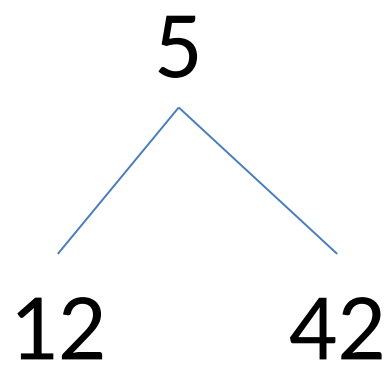

1
19
6
23
14
12
26
100
H1
H2
H3


6
19
23
14
12
26
100
H1
H2
H3
Merge


6
14
12
19
23
26
100

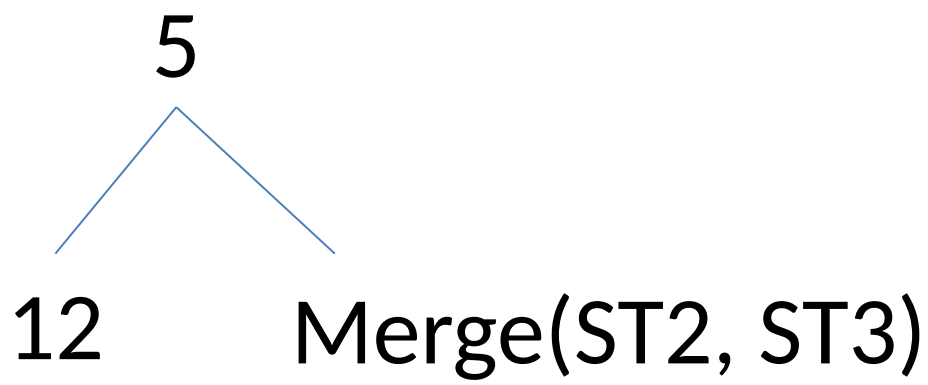
Draw the heap that would result after adding 9 to the given heap H1

H1:



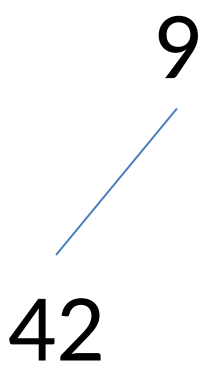
H2: 9

New root	ST1	ST2	ST3
5	12	42	9



Merge(ST2, ST3)

New root	ST1	ST2	ST3
9	empty	empty	42

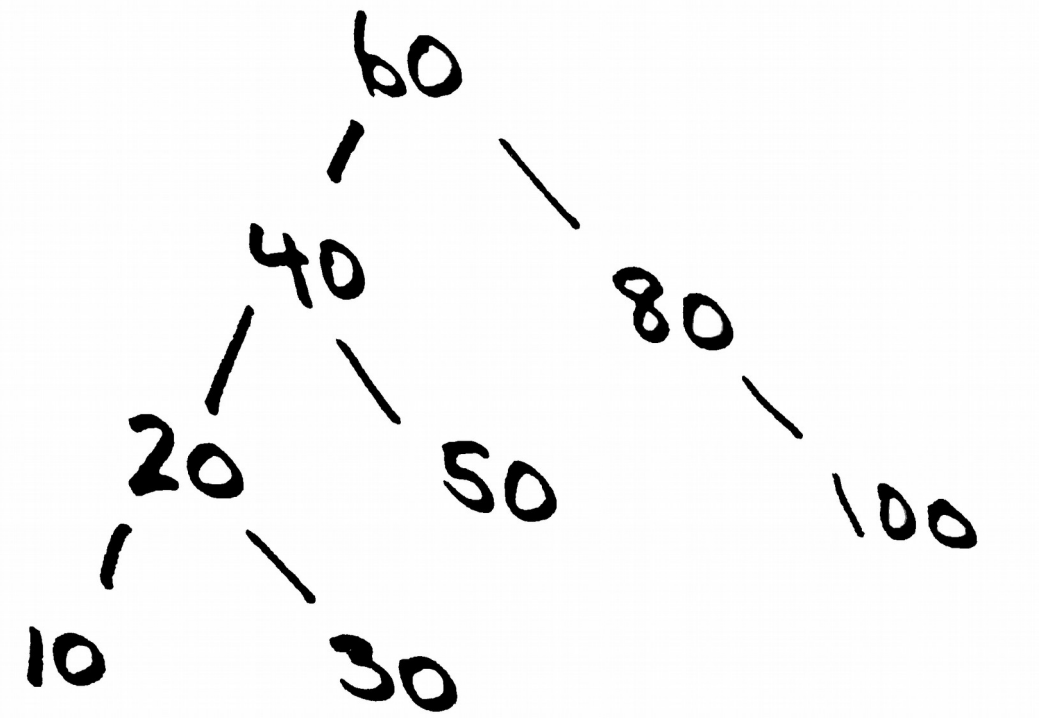


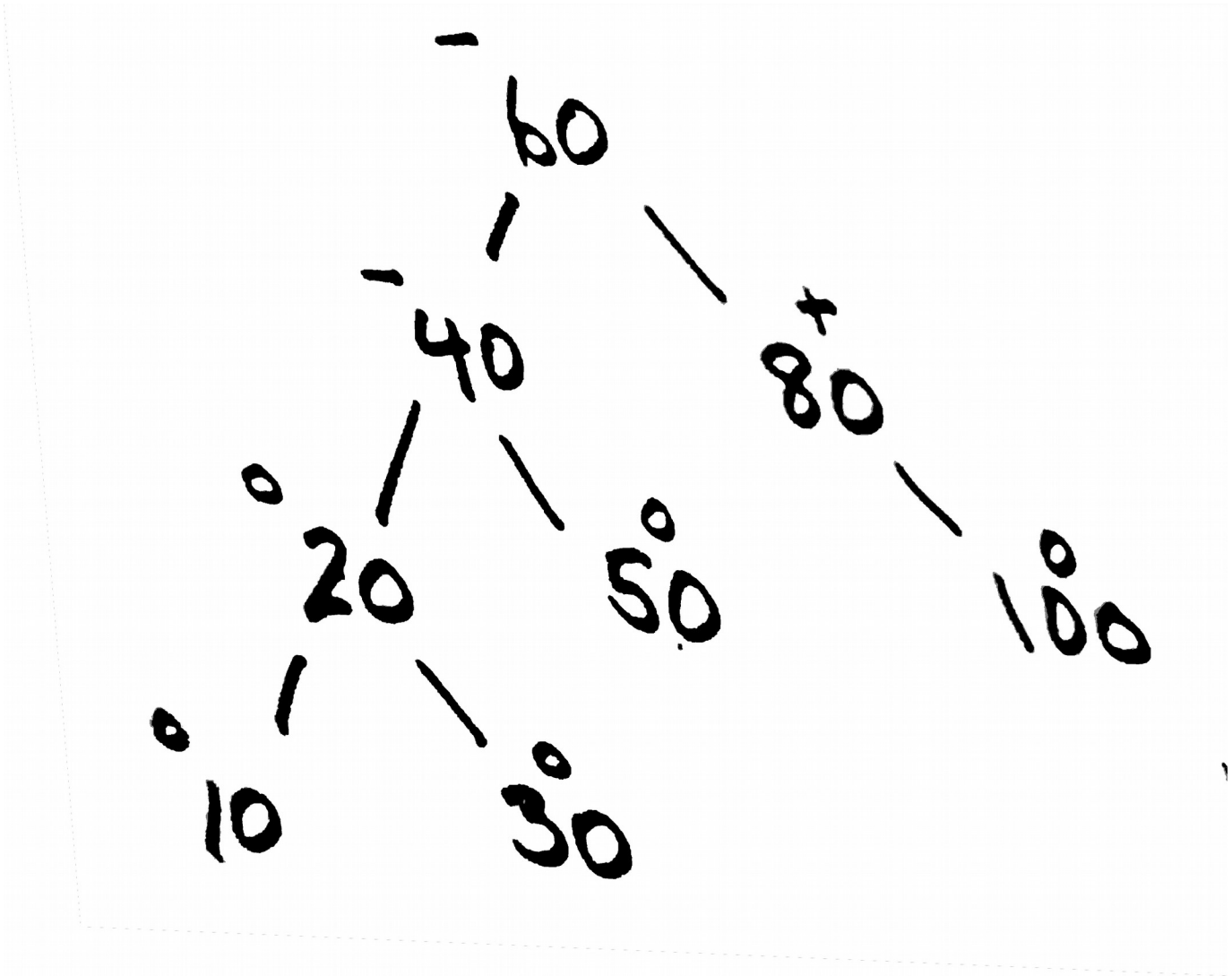
How to test that the answer returned is a valid answer?

- Is it a heap?
- Does the result contain all the elements from the original heap (the same number of occurrences)
- The new element has been added
- No elements in result that weren’t in original

AVL Trees

A binary search tree where, for each node in the tree, the heights of the left and right subtrees differ by at most 1

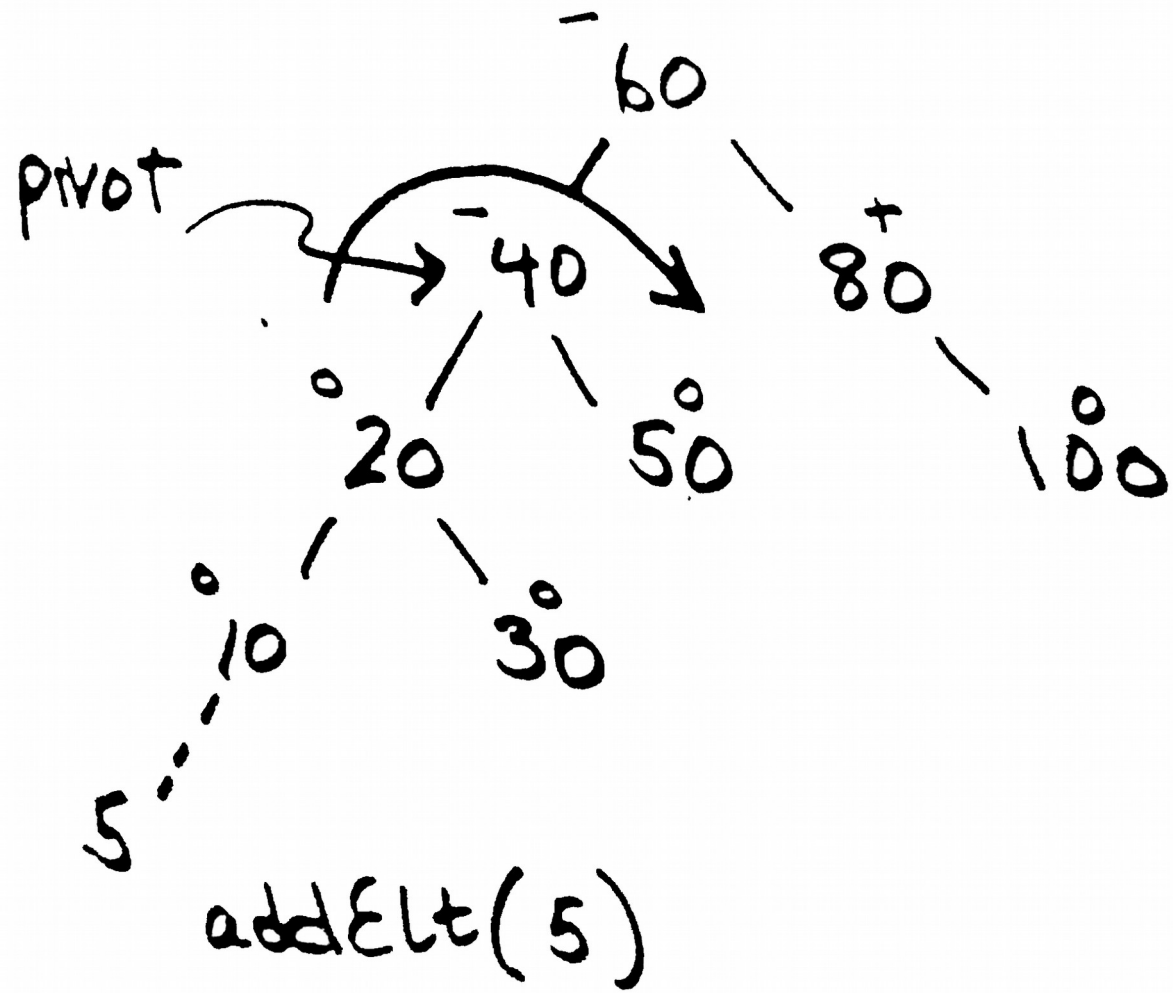


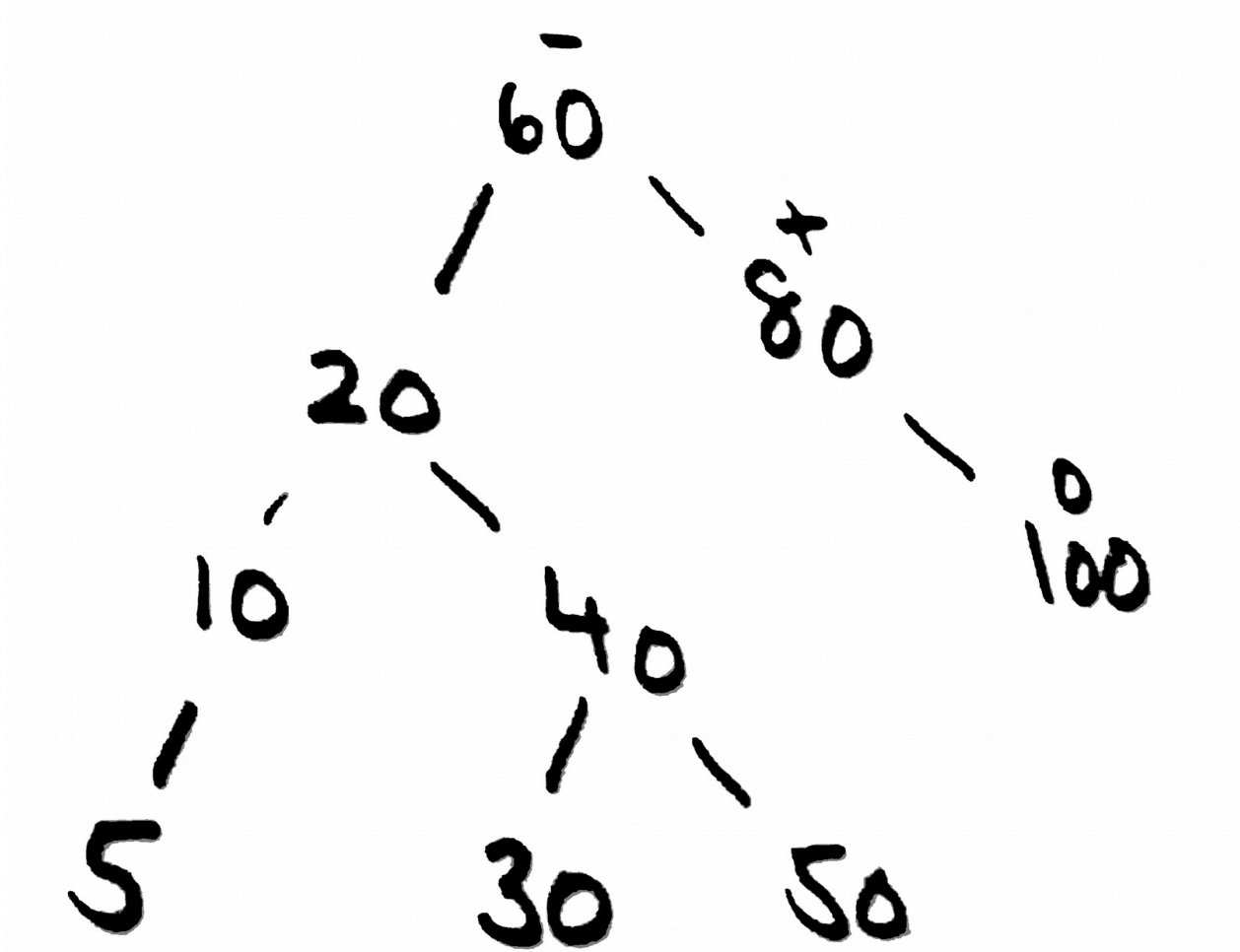


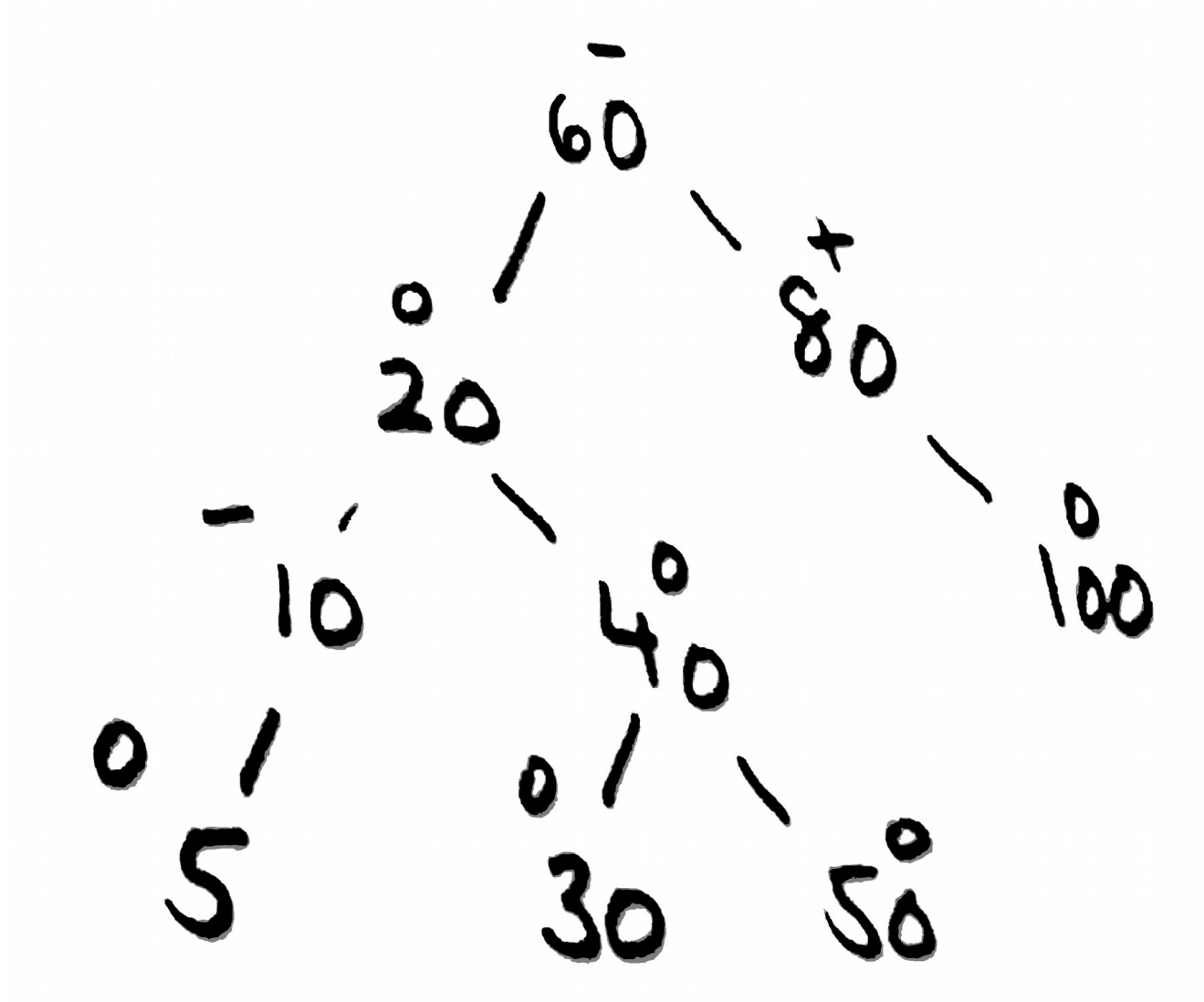
0 – perfectly balanced

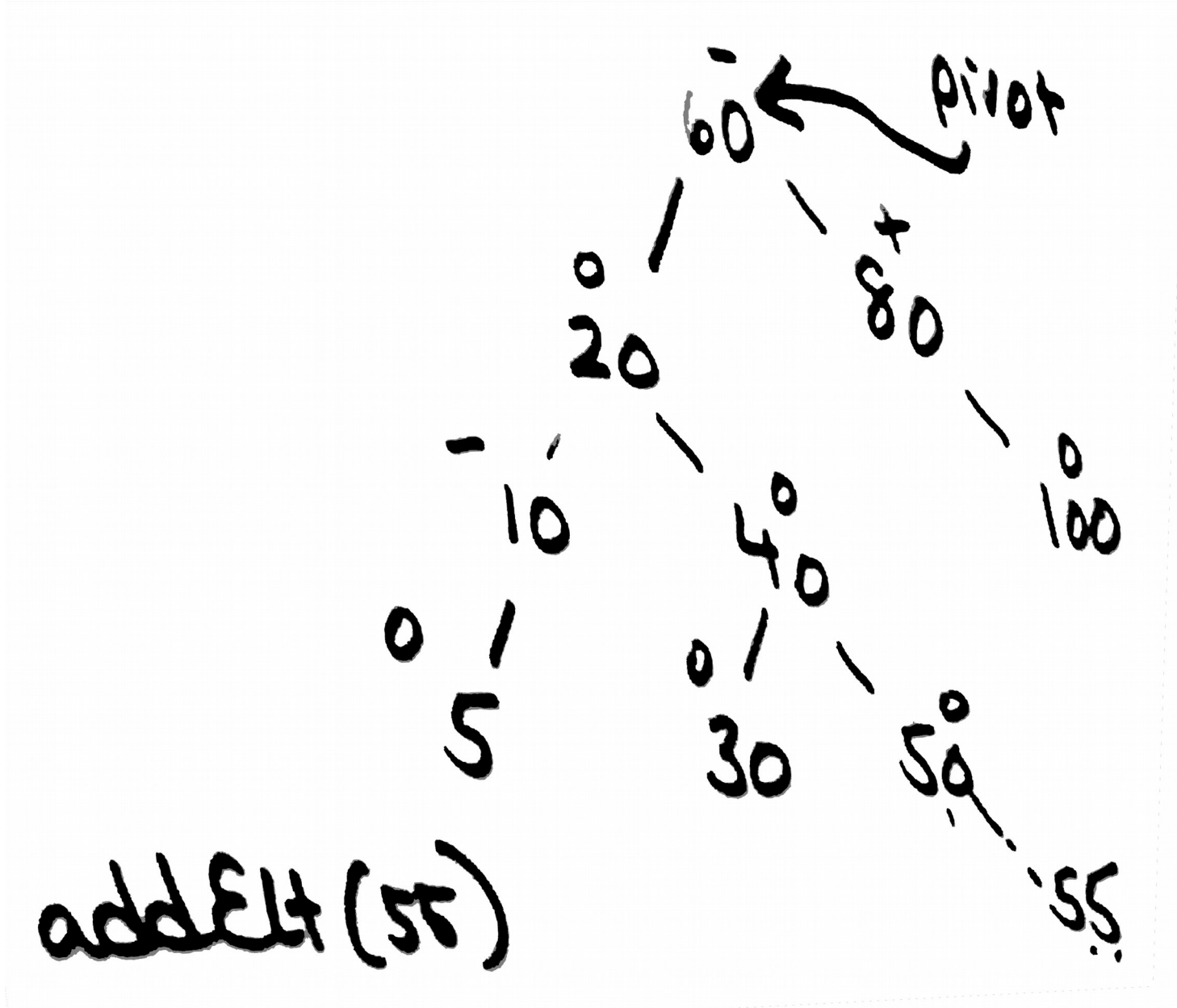
-1 – more on left

+1 – more on right

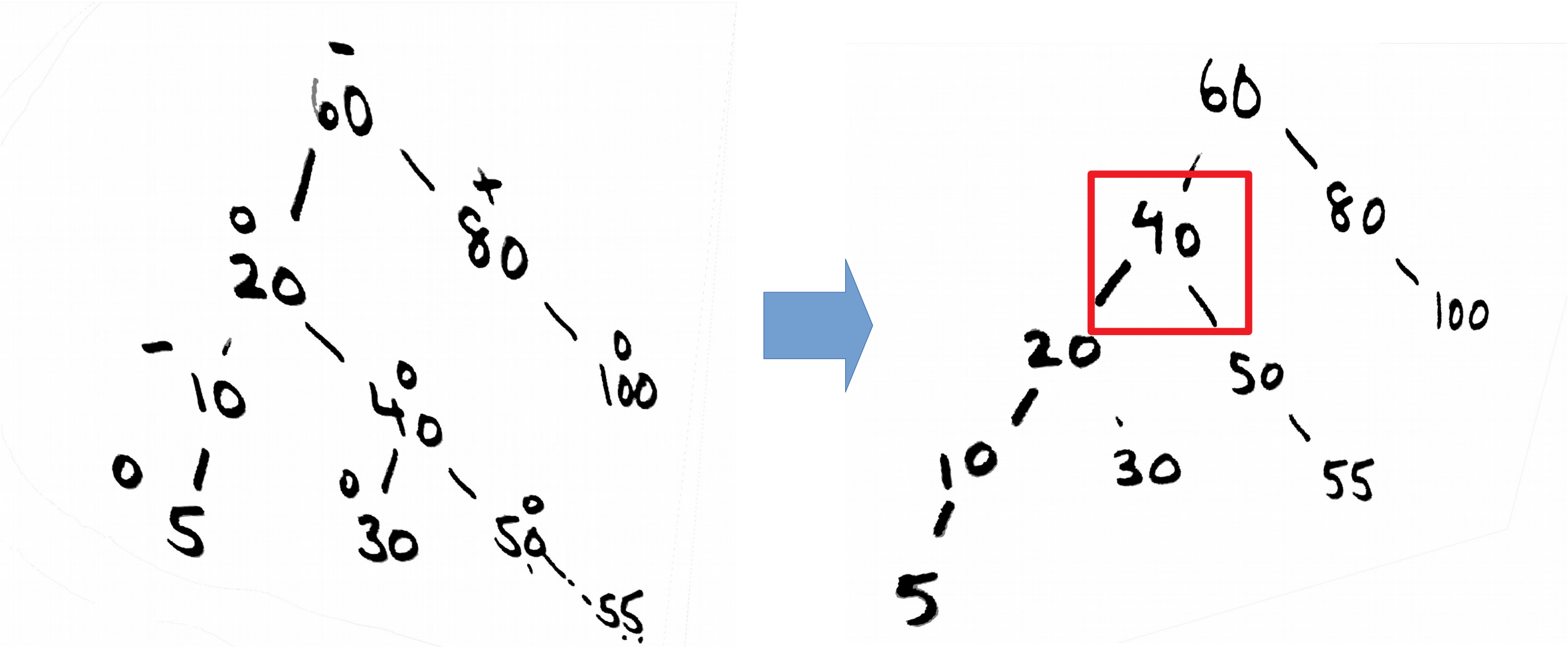

60
pivot
40
80
20
50
100
10
30
5
addElt(5)
single rotation


60
80
20
0
10
40
100
5
30
50


60
20
80
10
40
100
5
30
50


60
pivot
20
80
-10
40
100
5
30
50
55
addElt(55)

First rotation: rotation on child of pivot



Second rotation: rotation on pivot itself