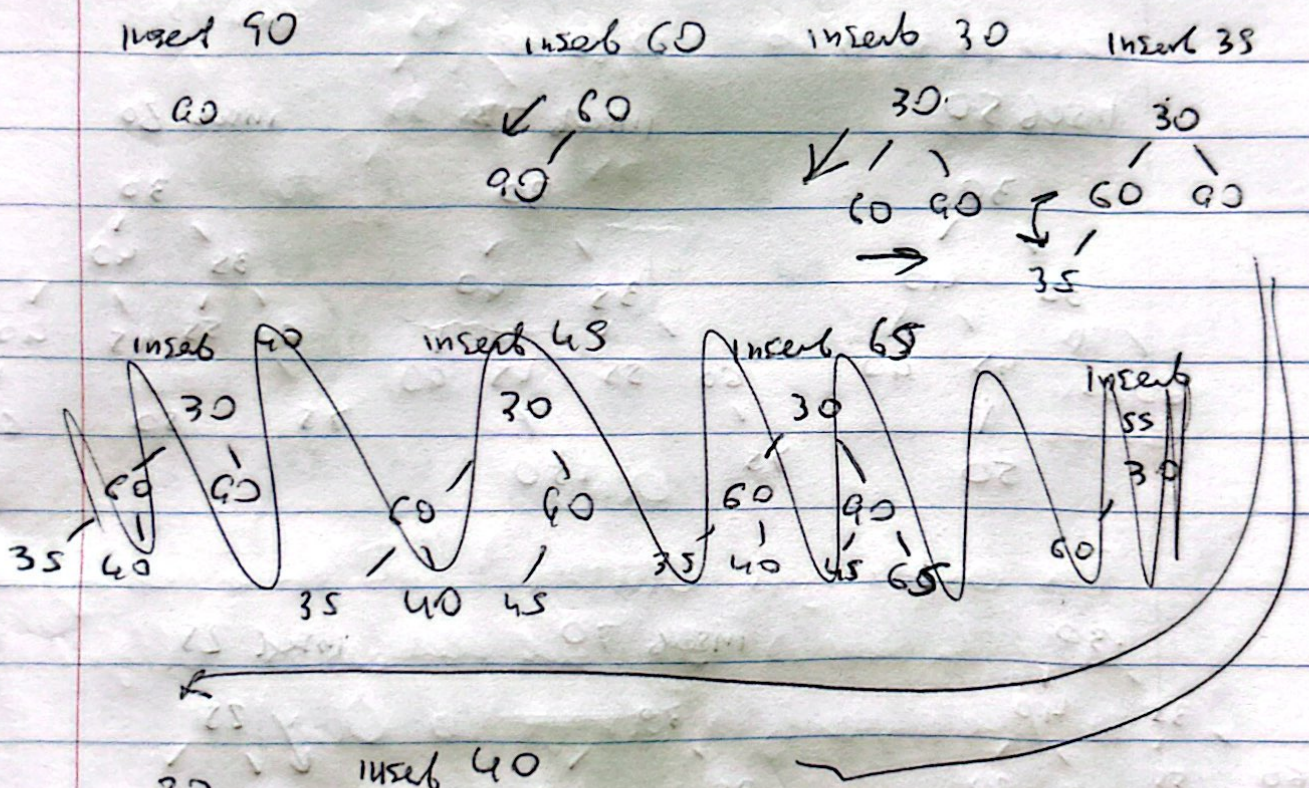
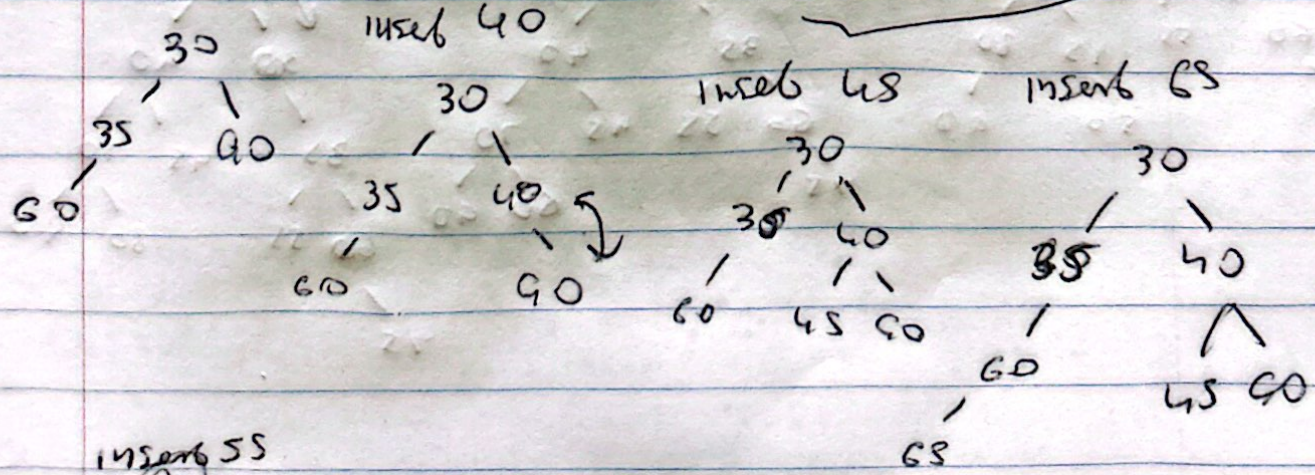
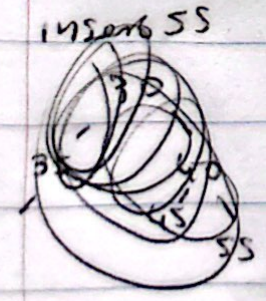
Heap sort

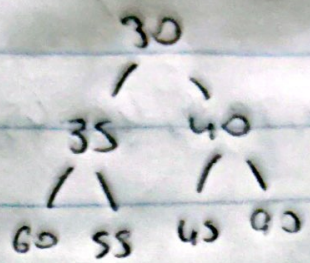
90, 60, 30, 35, 40, 45, 65, 55, 50, 75, , 60, 70, 75, 10, 85, 10.



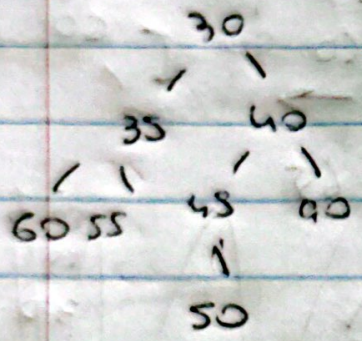




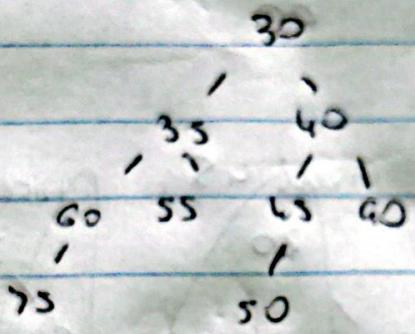
Insert 55



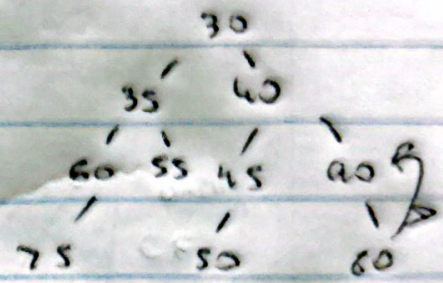
Insert 50

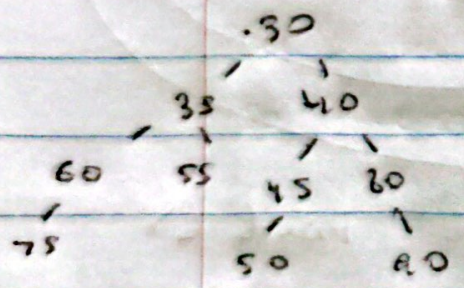


Insert 75

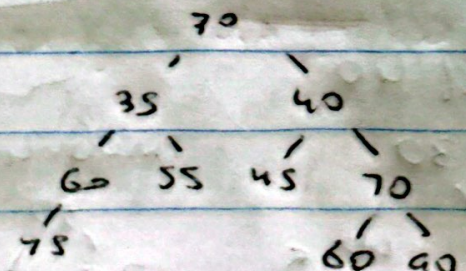


Insert 60

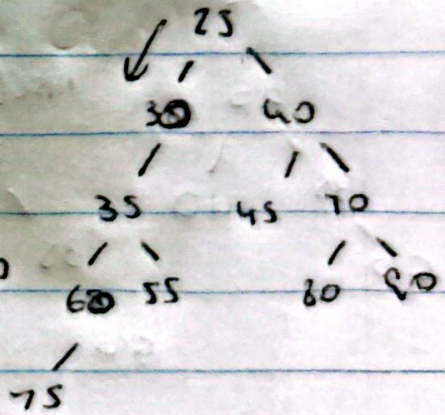




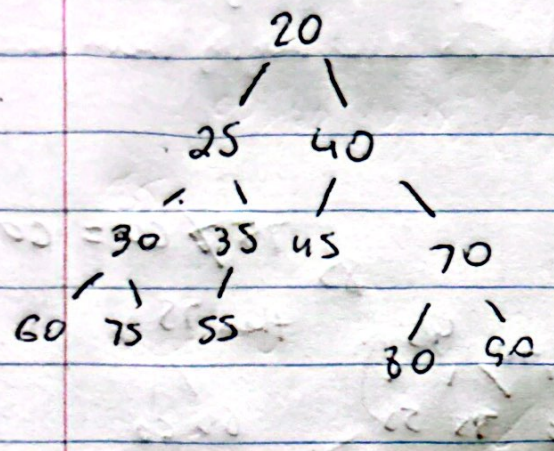
Insert 70



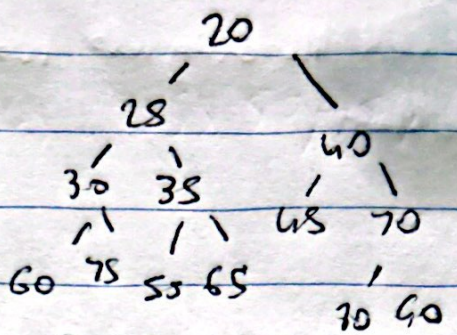
Insert 25



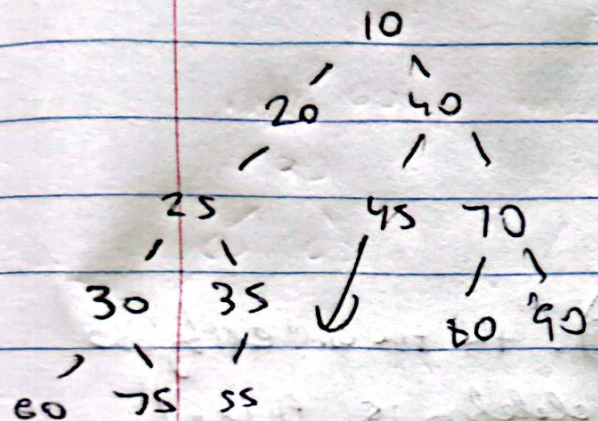
insert 20

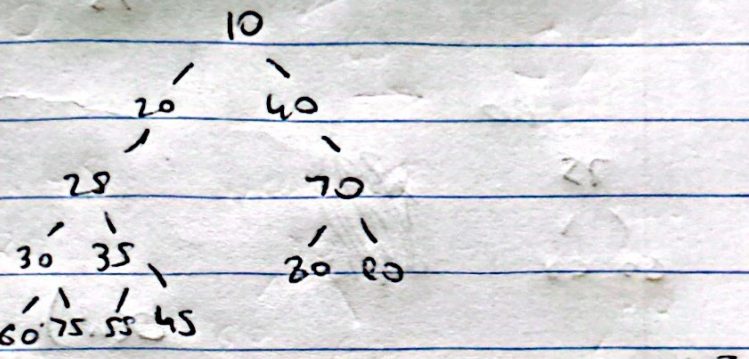


insert 65

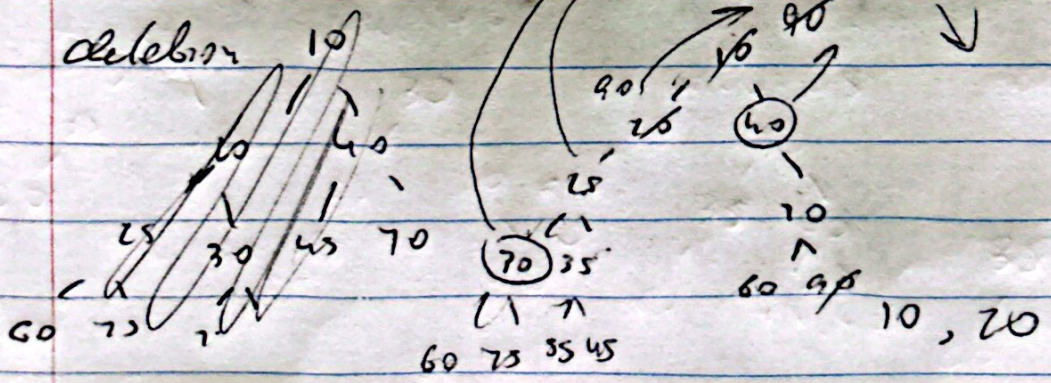


insert 10

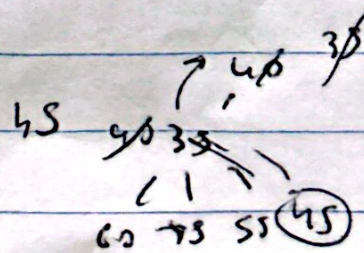


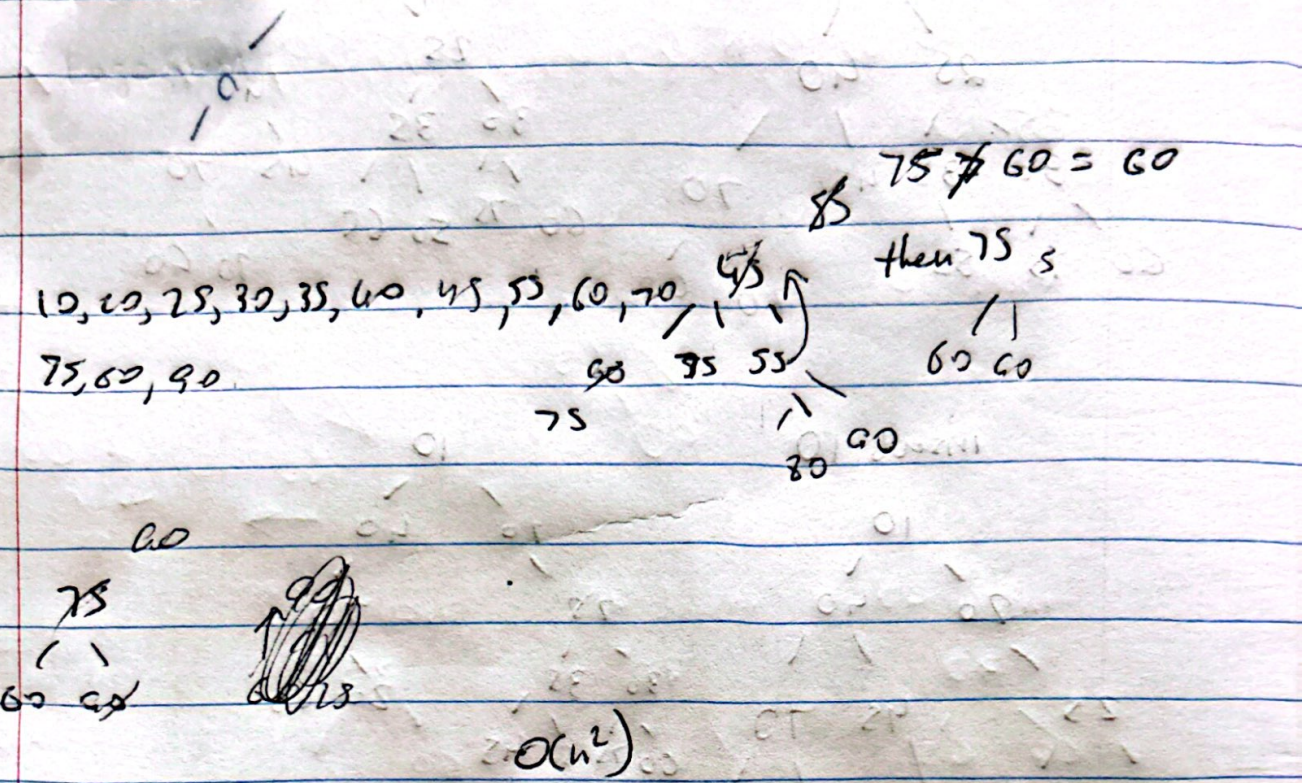


deletion



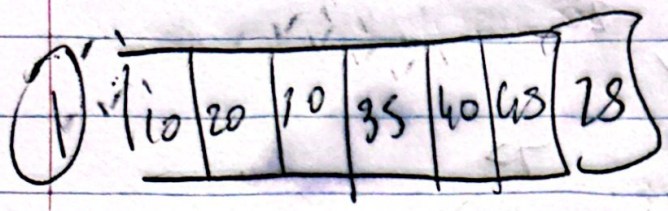
10, 20, 28, 30, 35, 40

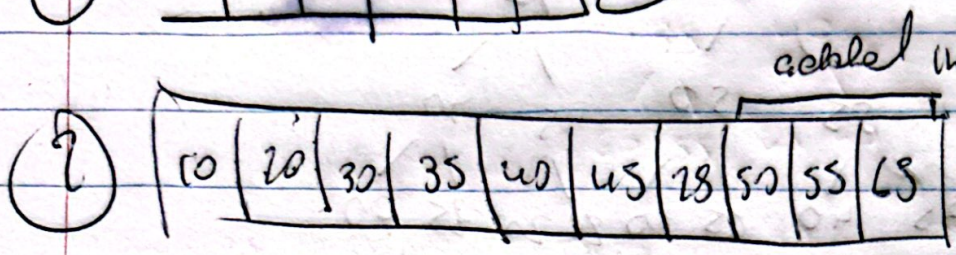


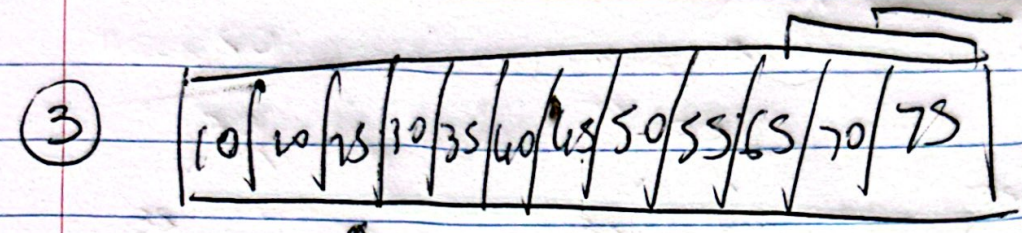


Time complexity is ~~O(nlogn)~~. the time complexity to heapify the heap is $O(n)$ but deletion is $O(\lg n)$ so time complexity is ~~O(nlogn)~~ @@ $O(n^2)$

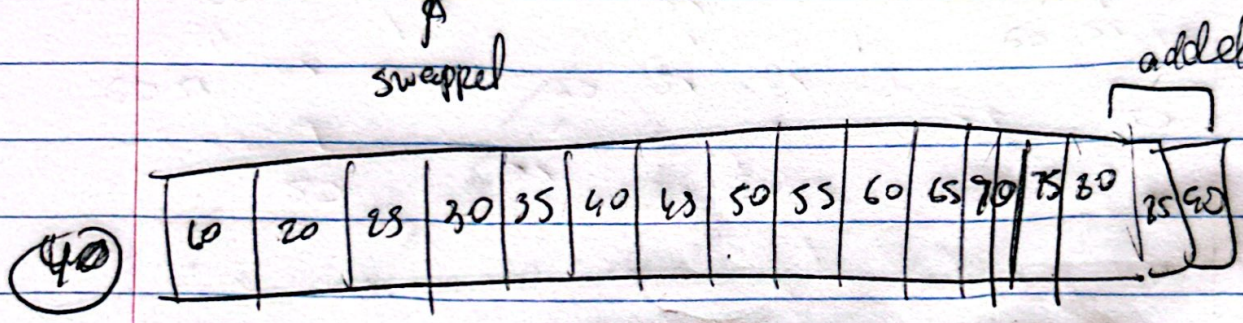
Quick sort





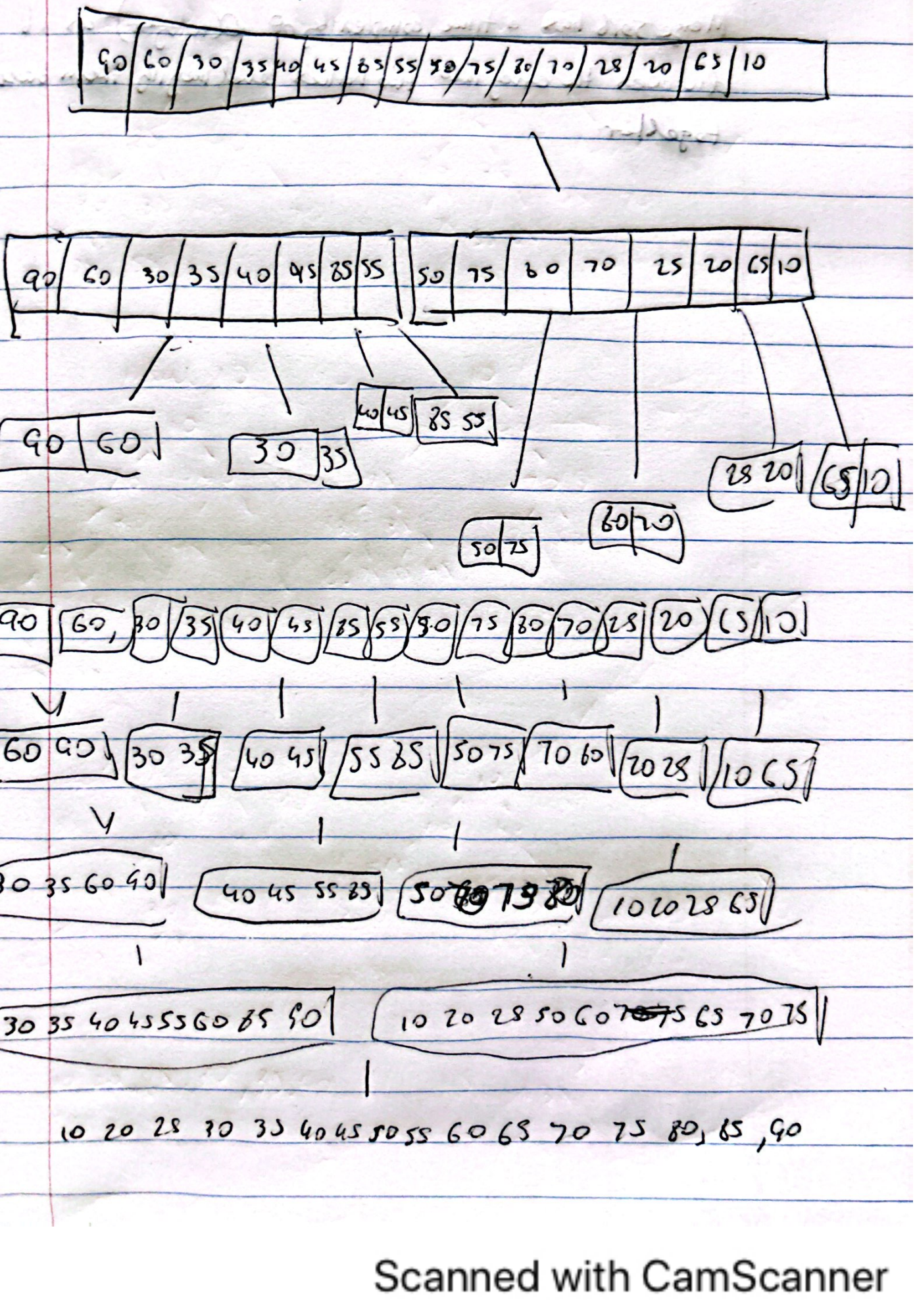


swapped



Quick sort is $O(n \log n)$. Time complexity depends on choice of pivot. The pivot divides the array into roughly two equal halves. leading to $O(n \log n)$

Merge sort



$O(n^2)$

Merge sort has a time complexity of ~~$O(n \log n)$~~ as it divides the array into two halves and merging them back together.