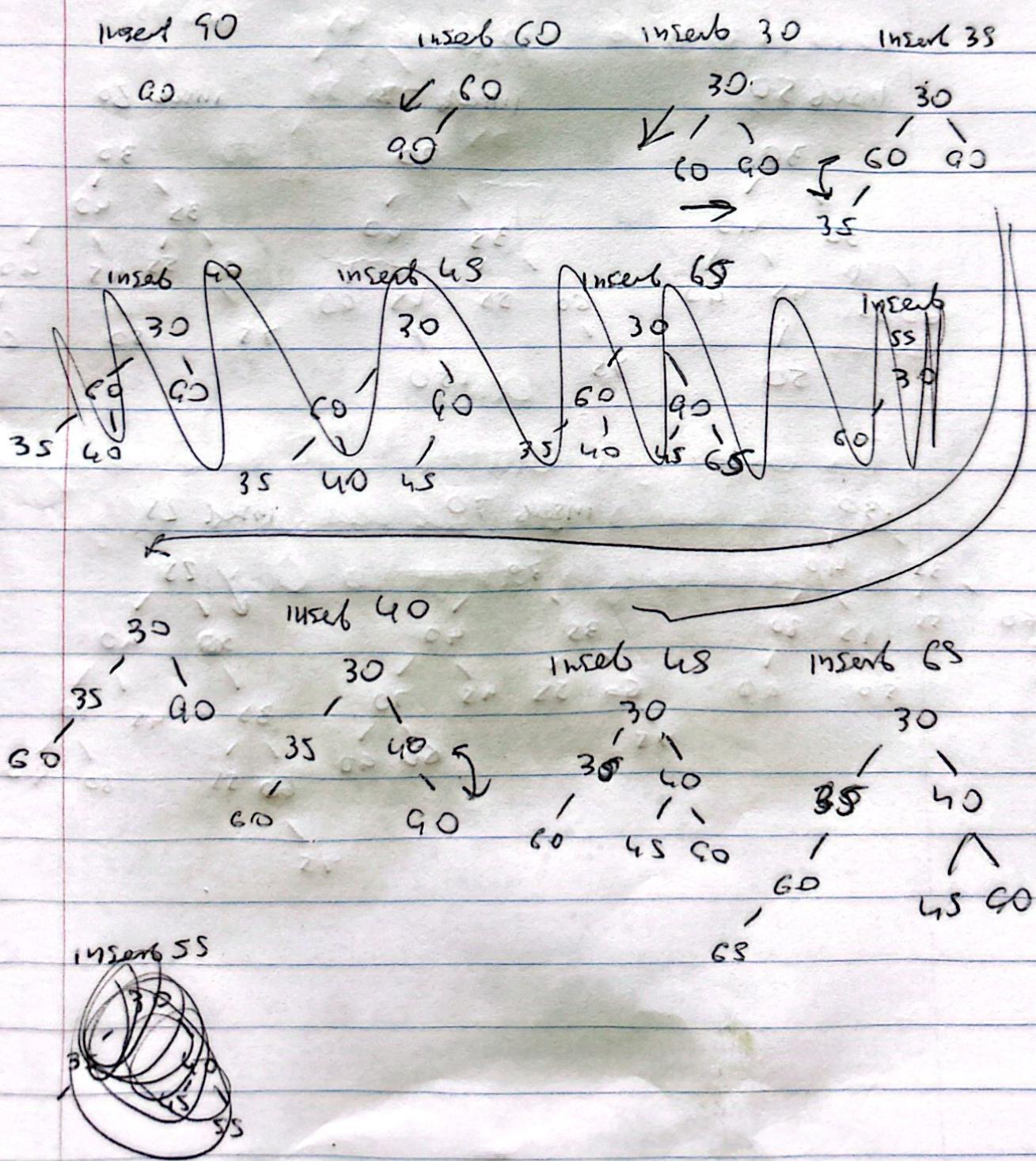


Heap sort

99, 60, 30, 38, 60, 48, 65, 55, 50, 75, 10, 70, 25, 10
85, 10.



inset 55

30
25 27
35 40
60 55 45 95

inset 50

30
35 40
60 55 45 90
50

inset 75

30
35 40
60 55 45 90
75

inset 60

30
35 40
60 55 45 90 95
75 50 60

30

inset 70

35 40
60 55 45 80
75 50 80

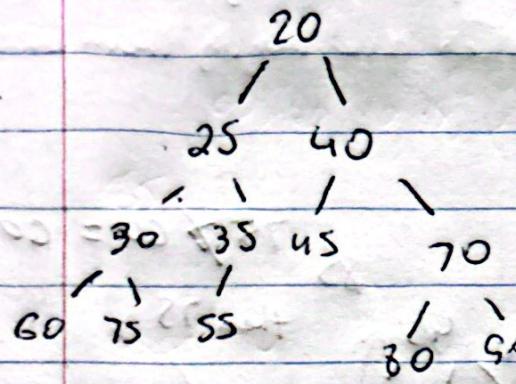
70

35 40
60 55 45 70
75 60 90

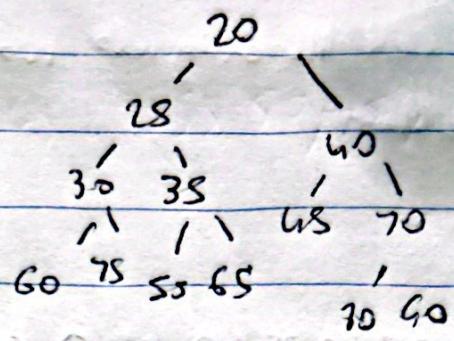
inset 25

25
30 40
35 45 70
60 55 80
75

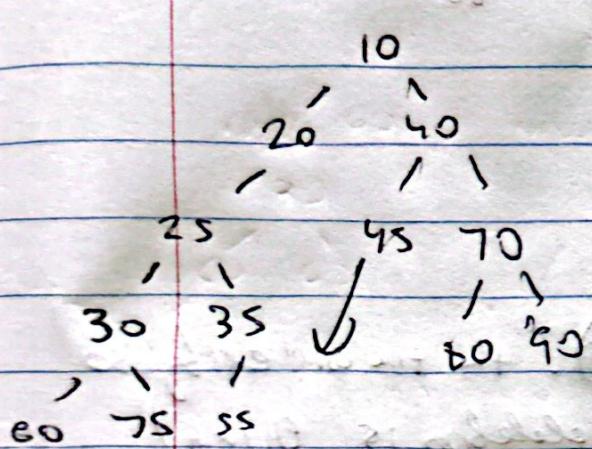
1986 20



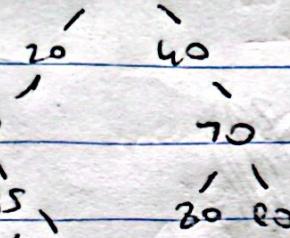
insert GS



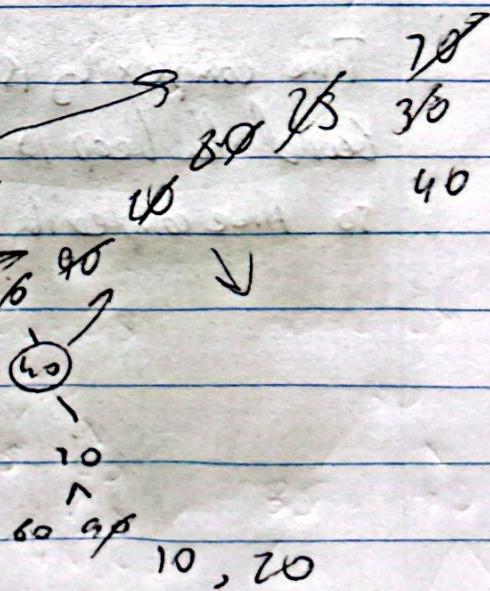
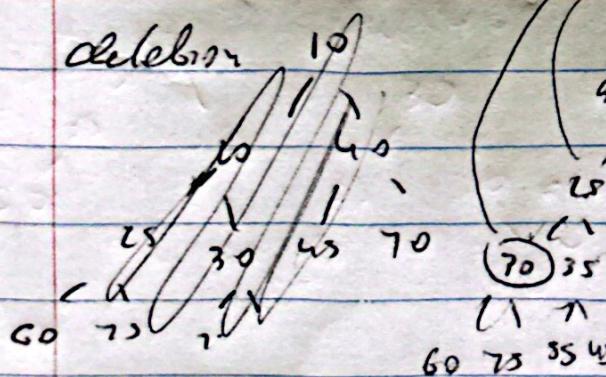
insub 10



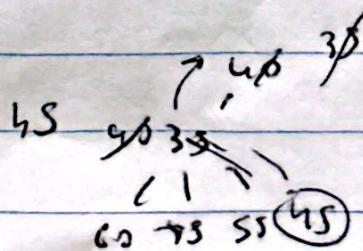
10

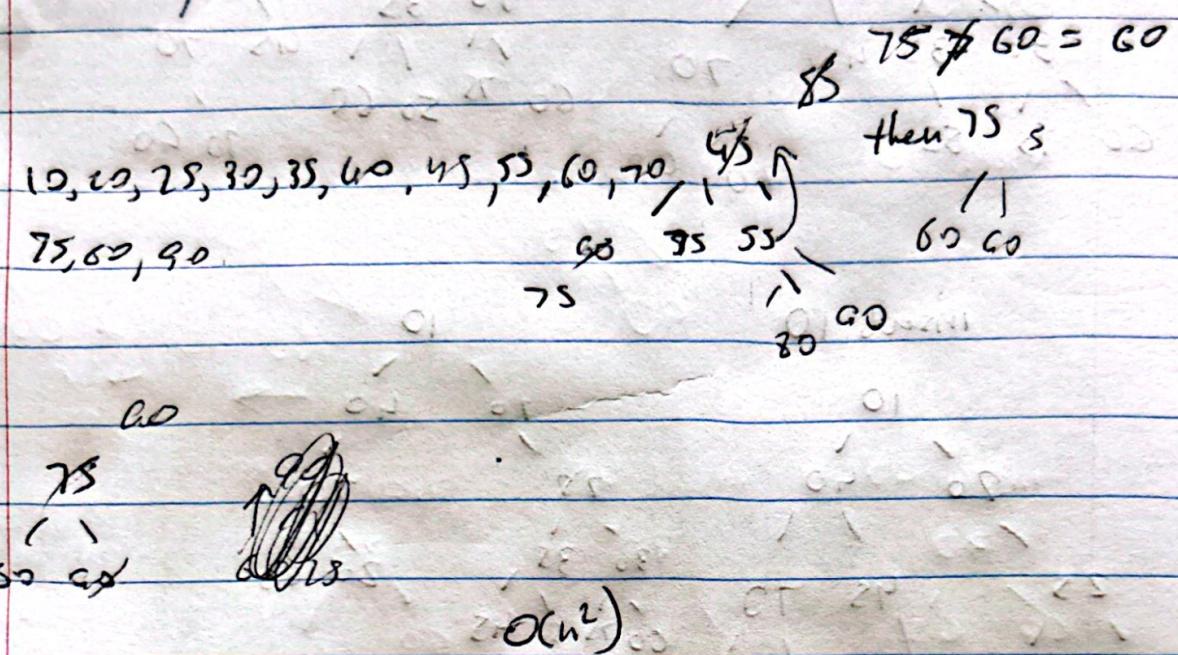


delebris



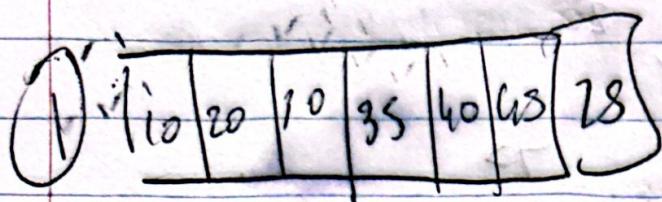
10, 20, 28, 30, 35, 40



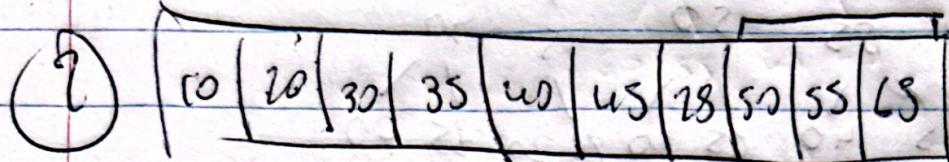


Time complexity is $\Theta(n \lg n)$. The time complexity to search the heap is $O(n)$ but deletion is $O(\lg n)$ so time complexity is $\Theta(n \lg n) \otimes O(n^2)$

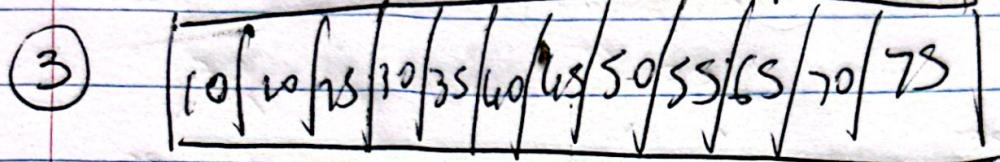
Quicksort



added into function

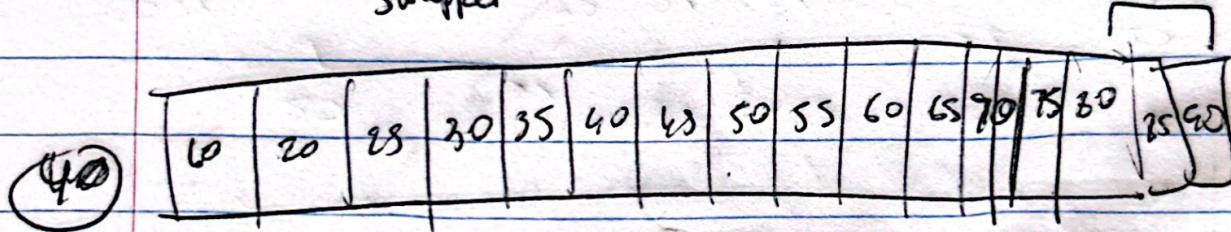


added in



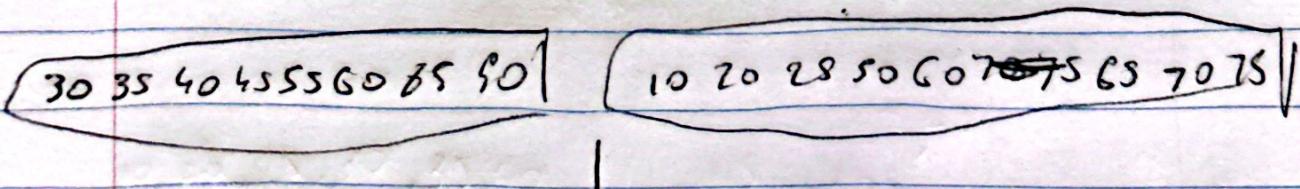
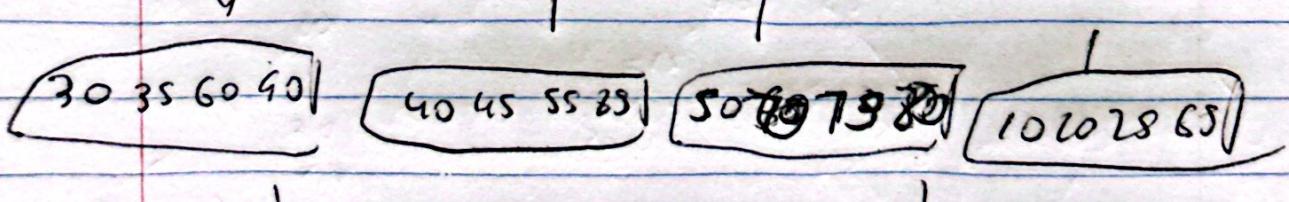
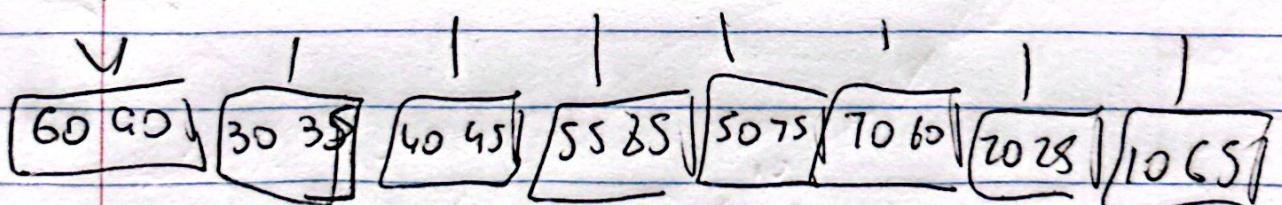
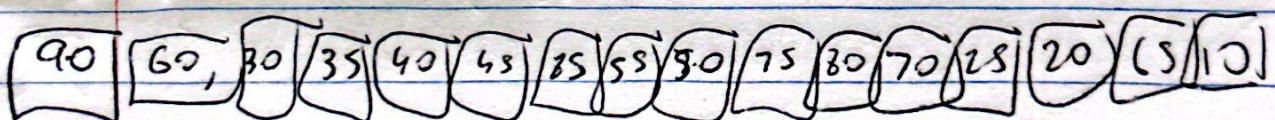
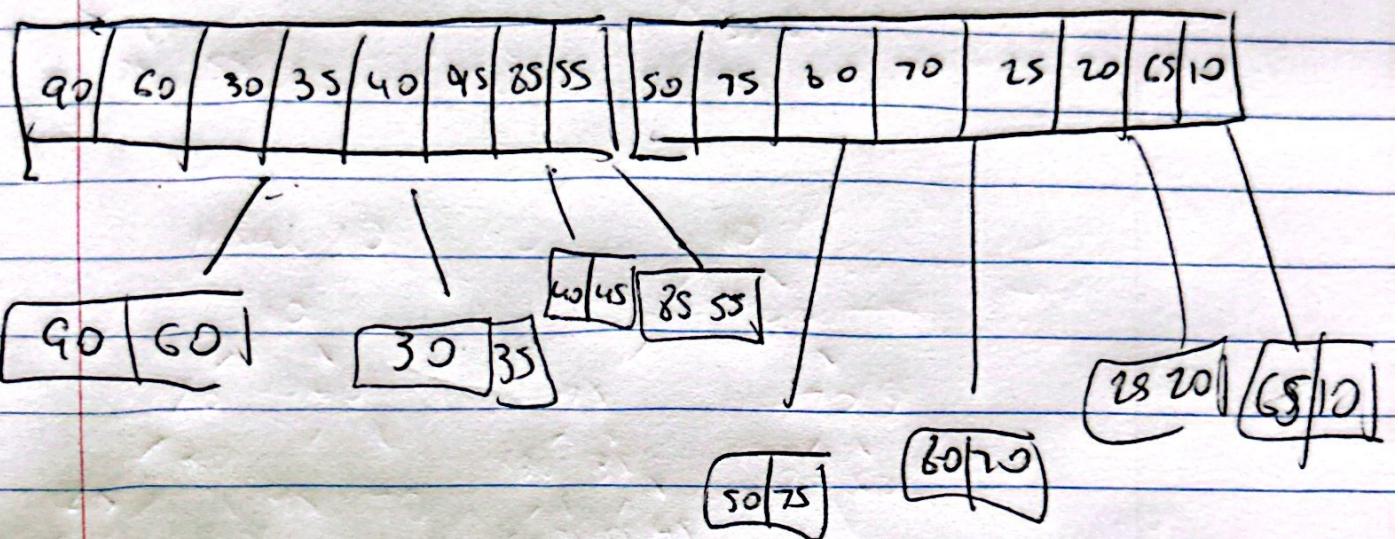
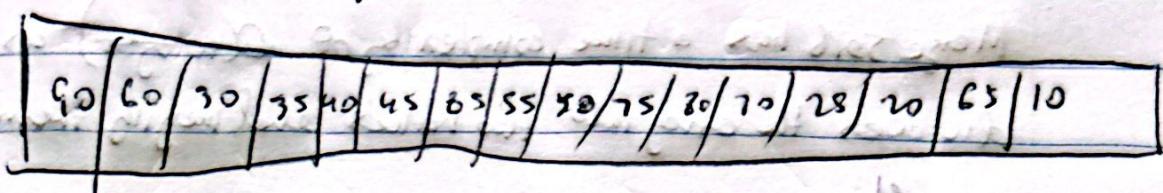
swapped

added in



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

Merge sort



Final sorted array: 10 | 20 | 28 | 30 | 35 | 40 | 45 | 50 | 55 | 60 | 65 | 70 | 75 | 80 | 85 | 90

$O(n^2)$

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