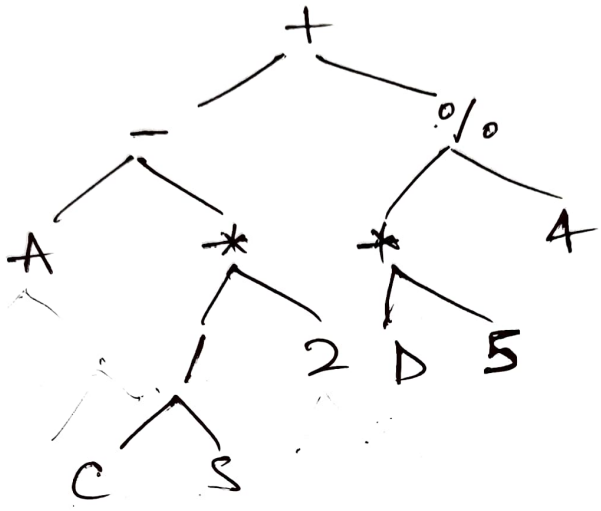
(1) Inorder, Preorder, Postorder.



Inorder - Left, Root, Right

Preorder - Root, Left, Right

Postorder - Left, Right, Root

INORDER : A - C / S * 2 + D * 5 % 4

PREORDER : + - A * / C S 2 % * D 5 4

POSTORDER : A C S / 2 * - D 5 * 4 % +

(2)
- ~~ab~~ - 4
- ah - 8
- bc - 8
- bh - 11
- ci - 2
- cd - 7
- cf - 4
- de - 9
- df - 14
- ef - 10
- fg - 2.
- gi - 6
- gh - 1
- hi - 7

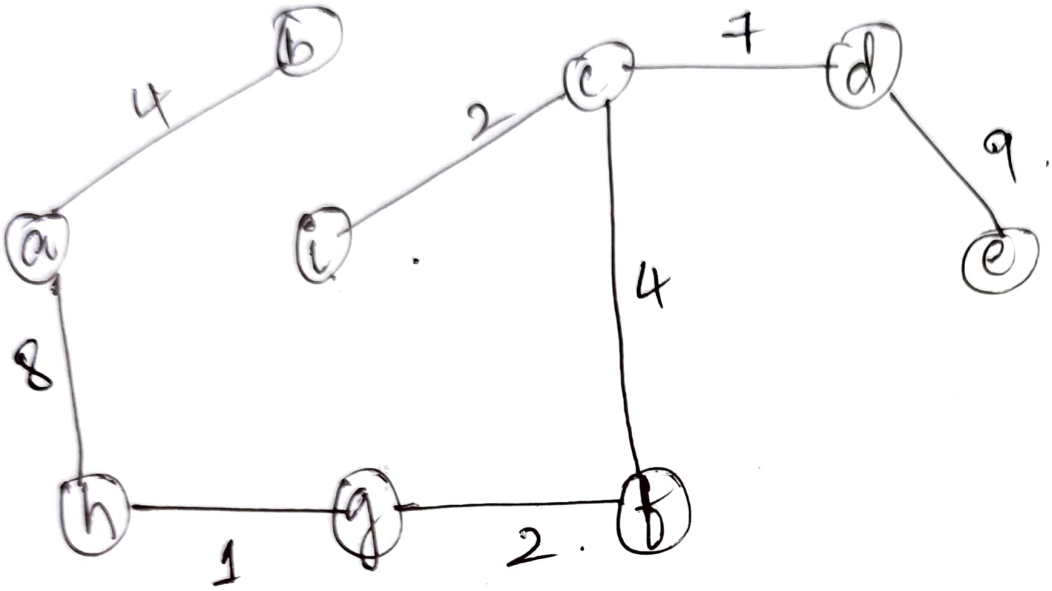
(i) Sort :

1 2 2 4 4 6 7 7 8 8 9

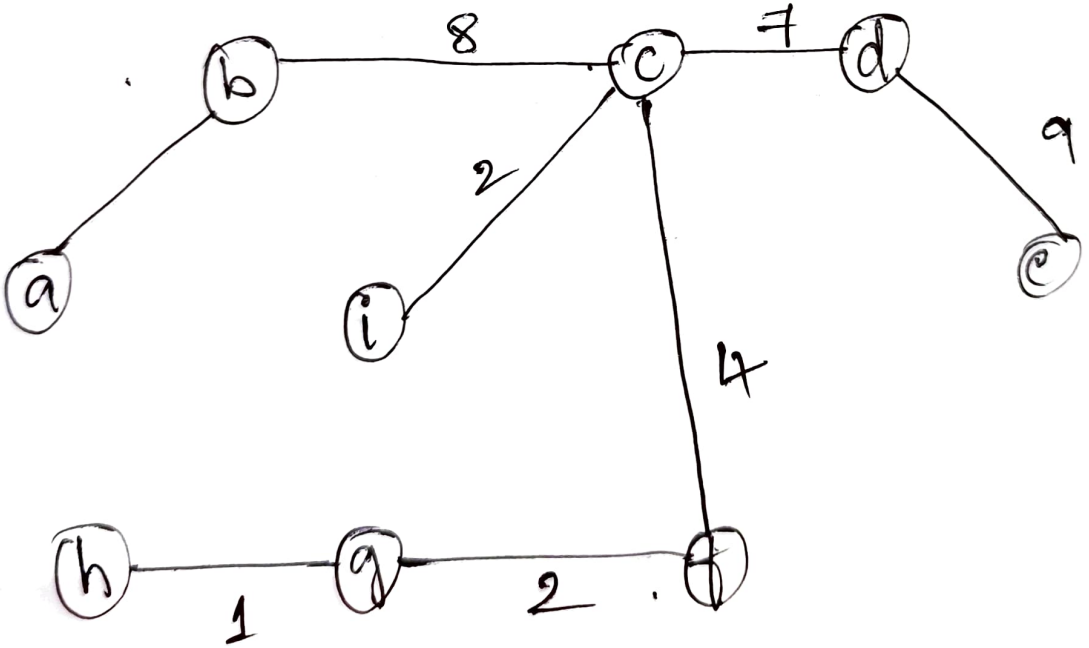
gh, ci, fg, ~~ab~~, cf, gi, hi, ~~cd~~, ~~bh~~, ah, de,

10 11 14

ef, bh, df.



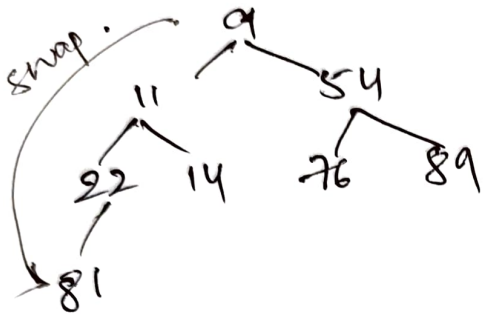
OR



(3) Minimum Heap Sort.

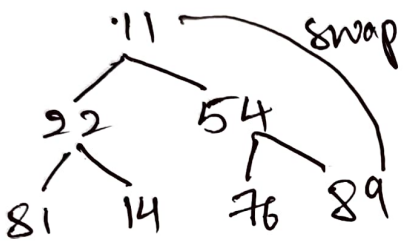
③ · Minimum Heap Sort

Initial Array : [81, 89, 9, 11, 14, 76, 54, 22]



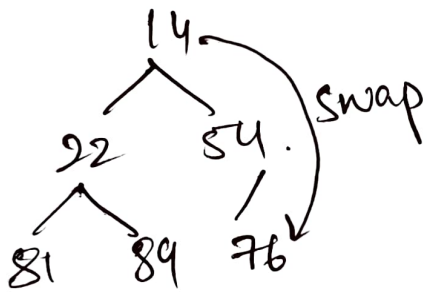
① [81, 11, 54, 22, 14, 76, 89]

heapify([81, 11, 54, 22, 14, 76, 89])



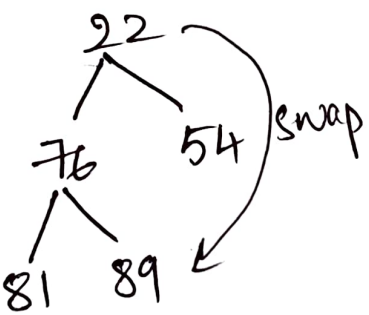
② [89, 22, 54, 81, 14, 76, 11, 9]

heapify([89, 22, 54, 81, 14, 76])



③ [76, 22, 54, 81, 89, 14, 11, 9]

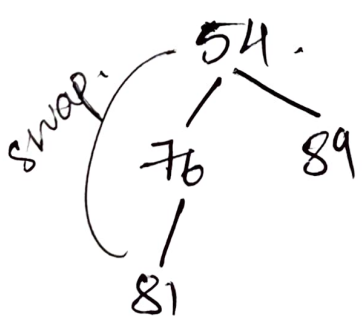
heapify([76, 22, 54, 81, 89]).



④ [89, 76, 54, 81, 22, 14, 11, 9]

([89, 76, 54, 81])

heapify.

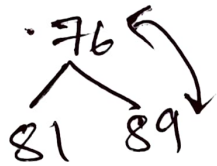


⑤ [81, 76, 89, 54, 22, 14, 11, 9].

([81, 76, 89]).

heapify.

(6)



[89, 81, 76, 54, 22, 14, 11, 9]

([89, 81])

heapify.

(7). 81 / 89. swap. [89, 81, 76, 54, 22, 14, 11, 9]

Now entire array is swapped.

Final Sorted Array:

[9, 11, 14, 22, 54, 76, 81, 89].