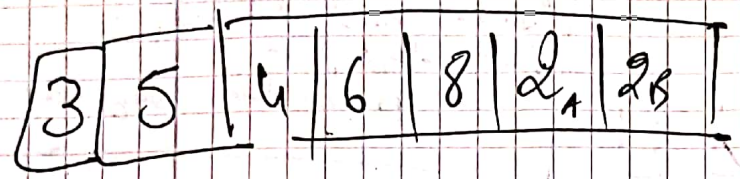
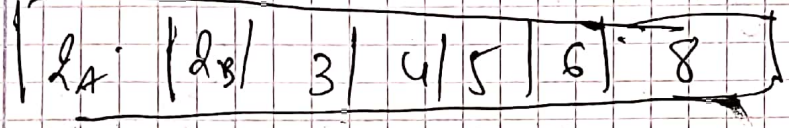
4. Stability of Algorithm. (Insertion, Merge, Heap, Bubble

Insertion Sort:



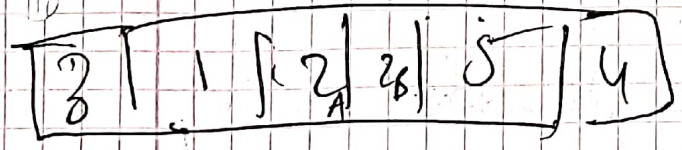
Insertion Sort is stable

After Insertion Sort



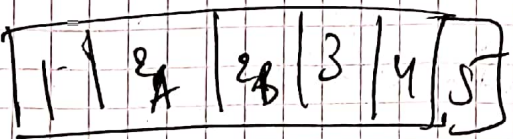
We just pick an element and place it in the right order, there is no swap between 2 elements, so its stable obviously.

Merge Sort



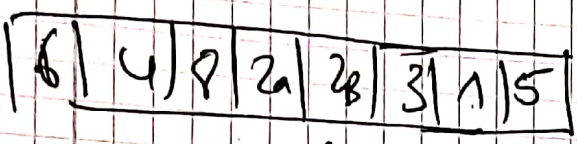
Merge sort is stable

After heap sort

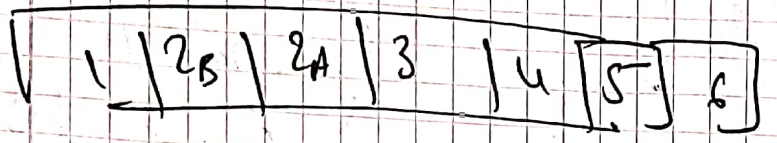


While merging 2 separated halves, there is no need to change the original order (L<=R) guarantee that we favor left hand values.

Heap Sort



After



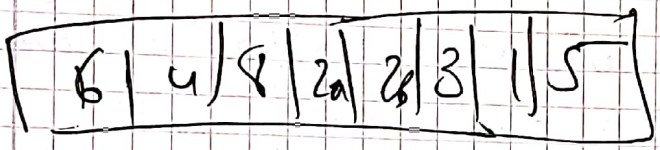
Unstable

heapsort pick the largest element and put it in the last of the list
So the element that was picked first stays first
∴ picked last in stays last

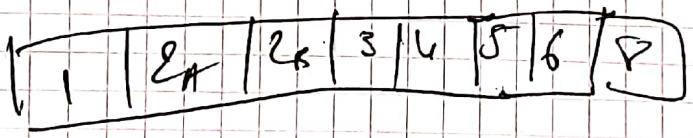
It's not stable because operations on the heap can change the original order.

Bubble Sort

Stable



After



Again, no swaps can be made if the two elements are equal

(it depends on the condition also,
if there is $\leq$ or $\geq$ swap
then, the algorithm is no more stable.

Insertion Sort

Best case complexity : $O(n)$
Worst case complexity : $O(n^2)$

- There is a slight change in the behaviour between the 2 cases.

So, I'll assume it's adaptive.

Merge Sort

Best case complexity : $O(n \log n)$
Worst case complexity : $O(n \log n)$

- there is literally no change in behaviour

$\Rightarrow$ Non-adaptive

Heap Sort

Best case complexity : $O(n \log n)$
Worst case complexity : $O(n \log n)$

No change

$\Rightarrow$ Non-adaptive

Bubble Sort:

Best case Complexity : $O(n)$

Worst case Complexity : $O(n^2)$

Change in behaviour

So, it's adaptive