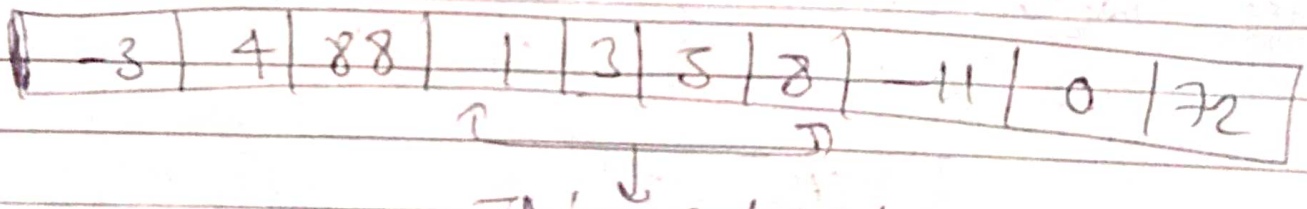
(2.) Adaptive → An adaptive algorithm is an algorithm that changes its behaviour based on information available on runtime.

→ It takes advantage of existing order in its input.

→ It benefits from local order → sometimes an unsorted array contains sequences that are sorted by default → the algorithms will sort faster.

→ Most of the times we just have to modify existing sorting algorithms in order to end up with an adaptive one.



→ Comparison based algorithms have optimal O(N log N) running time complexity.

→ Adaptive sort takes advantage of the existing order of the

input to try to achieve better times: maybe O(N) could be reached.

→ The more pre-sorted the input is, the faster it should be sorted.

→ IMPORTANT: nearly sorted sequences are common in practice!!!

→ Heapsort, merge sort: not adaptive algorithms, do not take advantage of pre-sorted sequences.

→ Shell sort: adaptive algorithm so performs better if the input is partially sorted.

(3.) Stable → Insertion sort is stable due to a no swaps usage.