Topic-05-sorting-overview-and-definitions

Wednesday, June 26, 2024 7:32 AM

SORTING ALGORITHM OVERVIEW!

① Bubble Sort: A very simple S.A. There are variations.

② Selection Sort: (In my mind) the most intuitive.

③ InsertionSort: Again, simple.

④ Merge Sort: Divide-and-Conquer. Sorting w/ a twist (DAC)

⑤ Heap Sort: Treat the list as a complete binary tree.
We also learn about max heaps.

⑥ Quick Sort: Very common D.A.C. sorting algorithm.

⑦ Comment: All of ↑ are comparison-based.
Work by comparing elements.
We'll prove that there is a limit on how fast a CBSA can be!

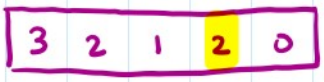
⑧ Counting Sort: A simple non-CBSA
which, under certain circumstances.
Can be faster than ↑

⑨ RadixSort: Another non-CBSA.

⑩ Associated Defns

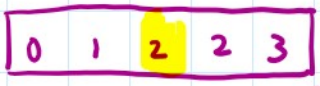
(A) Stability: Imagine a list w/ more than one of a value:

(A) Stability: Imagine a list w/ more than one of a value.

ex 

When sorted it could be 

← same pos. rel. to one another
A

or 

← nope!
B

depending on the alg!

defn: A sorting alg is stable if, whenever we have duplicate elements, they keep the same relative position.
A is stable (S.A. yielding A is stable)
B is not (" " B is not)

(B) Aux. Space: Not counting the memory storing the list, how much extra memory is used?
- loop variables?
- new list?
- measured as O or Θ

(C) In-Place:
Does our S.A. sort "on top of" the original list, say by swapping elements, or perhaps it constructs an entirely new list.
A S.A. is in-place if it sorts the list on top of itself.