Sorting Algorithms

Reset Progress | Reveal Solutions

1 Insertion sort example

Suppose that we want to sort the following array according to the alphabetical order using Insertion Sort.

In the first iteration, Insertion Sort starts moving C. Where does C end up after this iteration?

- (●) Position 1
- ( ) Position 2
- ( ) Position 3

Correct

Now we start moving A. Where does A end up after we are done with this iteration?

- (●) Position 1
- ( ) Position 2
- ( ) Position 3

Correct

In the next iteration, we move B. Where does it end up?

- ( ) Position 1
- (●) Position 2
- ( ) Position 3

Correct

The final array looks as follows.

2 Insertion sort questions

Can you see a pattern? When sorting an array using Insertion Sort, which of the following is correct after having iterated over the first i items.

- ( ) Item i is in its final position and will never move again.
- (●) The first i items are in sorted order.
- ( ) The first i items are in their final positions.
- ( ) All of the above.

Correct

What is the smallest exponent x such that Insertion Sort on an array of size n always takes time $O(n^x)$?

2

Correct

What if we run insertion sort on an already-sorted array. What is the smallest exponent x such that Insertion Sort on a sorted array takes time $O(n^x)$?

1

Correct

Which of the following describes the worst case runtime of Insertion Sort?

- ( ) $O(n^2)$
- ( ) $\Omega(n^2)$
- ( ) $\Omega(n)$
- (●) All of the above

Correct

3 Merge sort

The Merge operation takes two arrays A and B of size n which are already sorted and outputs the union of the two in sorted order. What is the smallest bound on the runtime of the Merge algorithm?

- ( ) $O(n \log n)$
- (●) $O(n)$
- ( ) $O(n^2)$

Correct

In Merge Sort run on array of size n, how many calls (in total across all levels of recursion) are made to the Merge subroutine?

- (●) $\Theta(n)$
- ( ) $\Theta(n \log n)$
- ( ) $\Theta(\log n)$

Correct

Is Merge Sort faster than Insertion Sort on all input arrays?

- ( ) Yes
- (●) No

Correct

Is Merge Sort faster than Insertion Sort on some arrays?

- (●) Yes
- ( ) No

Correct

If algorithm $\mathcal{A}$ is faster than algorithm $\mathcal{B}$ on some inputs, does that mean $\mathcal{A}$'s worst case runtime is better than $\mathcal{B}$'s worst case runtime?

- ( ) Yes
- (●) No

Correct

Is Merge Sort's worst case runtime asymptotically faster than Insertion Sort's worst case runtime?

- (●) Yes
- ( ) No

Correct