

32.5 Exercises

Factual

1. Suppose we have the array `[17, 15, 19, 32, 2, 26, 41, 17, 17]`, and we partition it using 3-scan partitioning using `17` as the pivot. What array do we end up with?
2. Suppose we have the array `[17, 15, 19, 32, 2, 26, 41, 17, 17]`, and we partition it using Tony Hoare partitioning using 17 as the pivot. What array do we end up with?

› Problem 1

› Problem 2

Conceptual

1. Which of the following arrays will result in N^2 behavior if we use quicksort which always uses the leftmost item as pivot, and which uses a stable partitioning strategy?
 - ☒ `[1, 1, 1, 1, 1, 1, ..., 1]`
 - ☒ `[1, 2, 3, 4, 5, 6, ..., N]`
 - ☒ `[N, N - 1, N - 2, N - 3, ..., 2, 1]`
2. Repeat the exercise above, but assume that we shuffle the array before partitioning.
3. Suppose we try to use Quick Select to find the median of `[17, 15, 19, 32, 2, 26, 41, 5, 9]`. How many total partitioning operations will we need to complete to find the median?
4. Which of the following sorts are stable?
 - ☒ Insertion sort
 - ☒ Quicksort L3 - use leftmost item as pivot, use 3scan partitioning

- ☒ Quicksort L3S - use leftmost item as pivot, use 3scan partitioning, shuffle before starting
- ☒ Quicksort LTH - use leftmost item as pivot, use tony hoare partitioning
- ☒ Quicksort PickTH - use median (computed with PICK) as pivot, use tony hoare partitioning

› Problem 1

› Problem 2

› Problem 3

› Problem 4

Metacognitive

1. Why does Java's built-in `Array.sort` method use Quicksort for `int`, `long`, `char`, or other primitive arrays, but Mergesort for all `Object` arrays?
2. Given that quicksort runs fastest if we can always somehow pick the median item as the pivot, why don't we use quickselect to find the median to optimize our pivot selection (as opposed to using the leftmost item)?

› Problem 1

› Problem 2

Previous
32.4 Summary

Next
33. Software Engineering III

Last updated 1 year ago

