Q1 Instructions

0 Points

To receive full credit on this quiz, you must score at least 50%.

The Github repo for Lecture 13 is at:
https://github.com/ucsd-cse12-f20/ucsd-cse12-f20.github.io/tree/master/lectures/lecture-13

Q2 QuickSort

1 Point

Which of the following statements about QuickSort are true?

- [ ] QuickSort always uses the high - 1 index as the pivot index
- [x] The elements in the array before the pivot index must be smaller than the pivot value (Not always true)
- [x] QuickSort is a recursive algorithm, like MergeSort
- [ ] The partition() method always returns the middle index
- [x] Based on the picture of QuickSort, it will have a similar run-time to MergeSort

Not true

Q3 Partition

2 Points

Consider the following code and the implementation of **partition()** discussed in lecture.

```
String[] b = {"b", "f", "a", "e", "c", "d" };
System.out.println(partition(b, 0, 6));
System.out.println(Arrays.deepToString(b));
```

Q3.1 Return Value

1 Point

What return value would **partition()** method print out for the above array, low and high?

- ○ 0
- ○ 1
- ○ 2
- ◉ 3
- ○ 4
- ○ 5
- ○ 6

Q3.2 Array

1 Point

What would the array look like after the above call to **partition()**?

- ○ a, b, c, d, e, f
- ○ a, b, c, d, f, f
- ○ b, a, c, d, e, f
- ○ b, a, c, d, f, e
- ◉ b, c, a, d, f, e
- ○ b, c, a, e, d, f
- ○ c, a, b, f, e, e