## Question 1 — 0 / 1 point

Please select all algorithms and their tight running times for the sorting problem discussed in our course.

- [x] insertion sort algorithm with running time O(n^2)
- [x] merge-sort algorithm with running time O(n log(n))
- [x] Quick-sort algorithm with running time O(n log(n))
- [ ] Quick-sort algorithm with running time O(n)

## Question 2 — 0 / 1 point

Which of the following statement(s) is/are correct?

- [ ] Merge-sort algorithm described in the class is an in-place algorithm.
- [ ] If there exist optimal solutions that are not achieved by greedy algorithm, then the greedy algorithm does not work.
- [x] Quick-sort algorithm described in the class is an in-place algorithm.
- [x] In quick-sort algorithm, partition function divides the array into two parts based on the pivot element.

## Question 3 — 0 / 1 point

Which of the following algorithm is (are) a stable sorting algorithm?

- ( ) Merge-sort algorithm
- ( ) Quick-sort algorithm with in-placement implementation

## Question 3 — 0 / 1 point

Which of the following algorithm is (are) a stable sorting algorithm?

- [x] Merge-sort algorithm
- [ ] Quick-sort algorithm with in-placement implementation
- [ ] Quick-sort algorithm
- [ ] All of them
- [ ] None of them

## Question 4 — 0 / 1 point

What does the correct intermediate array(s) look like when it is being sorted with the merge-sort algorithm on [4,3,5,6,8,9,1,2]?

- [ ] [4,3,5,6,8,9,1,2]->[3,4,5,6,1,2,8,9]->[1,2,3,4,5,6,8,9]
- [ ] [3,4,5,6,8,9,1,2]->[3,4,1,2,5,6,8,9]->[1,2,3,4,5,6,8,9]
- [x] [3,4|5,6|8,9|1,2]->[3,4,5,6|1,2,8,9]->[1,2,3,4,5,6,8,9]
- [ ] [3,4,5,6,8,9,1,2]->[1,2,3,4,5,6,8,9]->[1,2,3,4,5,6,8,9]
- [ ] None of them is correct.

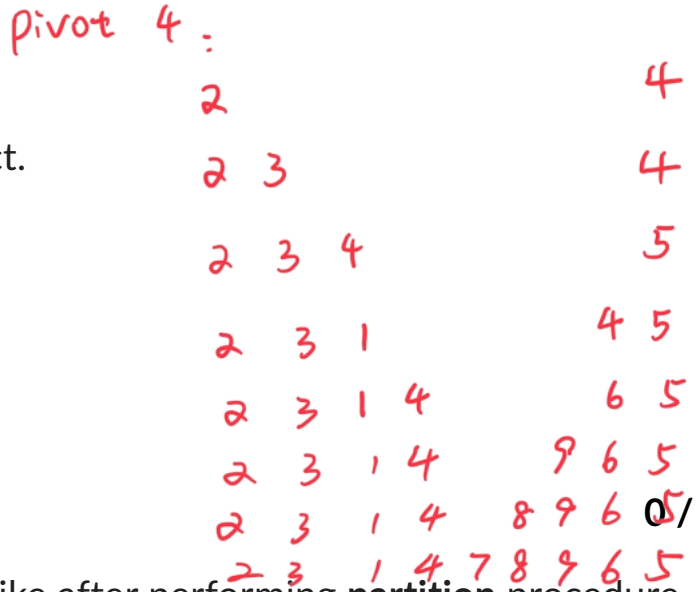
## Question 5 — 0 / 1 point

What does the array look like after performing **partition** procedure (following the in-place implementation) of quick sort **once** on

## Question 5 — 0 / 1 point

What does the array look like after performing **partition** procedure (following the in-place implementation) of quick sort **once** on [4,3,5,6,7,8,9,1,2] (assume we take the first element as pivot)?

- ✓ [2,3,1,4,7,8,9,6,5]
- None of them is correct.
- [3,1,2,4,5,6,7,8,9]
- [1,2,3,4,5,6,7,8,9]
- [2,5,1,4,7,8,9,6,3]




## Question 6 — 0 / 1 point

What does the array look like after performing **partition** procedure (following the in-plac implementation) of quick sort **Twice** on [4,3,5,6,7,8,9,1,2] (assume we always take the first element as pivot)?

- None of them is correct.
- [1,2,3,4,5,6,7,8,9]
- [1,2,3,4,5,6,8,9,7]
- [2,3,1,4,7,8,9,6,5]
- ✓ [1,2,3,4,5,6,7,9,8]