EXERCISE

Sorting is one of the categories many computer scientists have been working to optimize algorithm complexity. Quick sort is one of the fastest, therefore, the most popular, sorting algorithms used by many application programs today. Quick sort works in the following way:

Assumptions:

(a) There is an array of $N$ elements
(b) Each element in the array is an integer (any integer, including both negative and positive integers)

Quick Sort Algorithm:

Step ①: Pick up the first (the right end) element in the array (it’s “3” in the example below) and call it “pivot”
Step ②: Prepare three new empty arrays (left, center, and right)
Step ③: Move the pivot to the first slot in the center array.
Step ④: Scan each element in the array with $N$ elements from the right to the left from the next number in the left of the pivot and move each element that is smaller than the pivot (i.e., “3”) to the first empty array (the left array) and move each element that is larger than or equal to the pivot to the third empty array (the right array).
Step ⑤: Repeat ③ and ④ until each array has only one entry (or all elements in an array are the same).
Step ⑥: Concatenate all arrays from the left to the right, to make one array of $N$ elements.

Algorithm Complexity of Quick Sort Algorithm using SISD:
For each round, a SISD processor performs comparisons \((N-1)\) times. In the worst case, the first element can be the smallest number, resulting in \((N-1)\) rounds. Thus, the complexity for the worst case will be \(O((N-1)^2) = O(N^2)\).

Questions: If we used a SIMD computer with as many processing units as you would need:

(a) Is it possible to improve the algorithm complexity of quick sort? If, yes, what is the complexity order and explain how it is possible. If not, explain why not.

(b) Is it possible to improve the (actual) execution time of quick sort? If, yes, explain how it is possible. If not, explain why not.

Assume the followings:

- You always have infinite capacity of memory (for both SISD and SIMD architecture computers)
- You always have an infinitely large number of processing units in a SIMD architecture computer
- All processing units in a SIMD architecture computer can read from a memory address physically at the same time
- Preparing whatever size of an array requires a constant time (i.e., \(O(1)\))

Note 1: Most (85%) of the credit goes to a correct explanation (not to correct YES or NO).

Note 2: It is assumed that providing your solutions (conclusions and explanations) in such a way that it is readable (reasonably neat handwriting) and understandable (reasonably well organized) to Dr. Fujinoki. Any solution that is not readable or understandable will be subject to some minor or major penalties.

Note 3: “Nothing wrong” solution will not earn much credit, if important idea(s) is (are) not explained.