EXERCISE #1

Suppose that there is a data structure which consists of $N^2$ elements. At the beginning, each element in the data structure is initialized by a constant “$x$” (Figure 1). If we need to update the contents of the data structure as shown in the figure below (Figure 2) using a SIMD computer, answer the following questions:

(1) What is the algorithm complexity if SISD system is used?

(2) Can a SIMD computer improve the calculations?

(a) What is the best achievable algorithm complexity?

(b) How many processors are necessary to achieve the best algorithm complexity?

(a) Describe your algorithm to achieve (a). I assume it is your responsibility to clearly explain how your algorithm can achieve the algorithm complexity you mentioned for (a).
EXERCISE #2

Sorting is one of the categories many computer scientists have been working to optimize algorithm complexity. Bubble sort is one of the classic but popular sorting algorithms used by many application programs today. Bubble 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)

SISD Bubble sort

SIMD Bubble Sort =there is no opportunity for anything to be I parallel

Bubble Sort Algorithm (example: [5, 1, 6, 3, 4])

Step ①: Compare the first two items in the list. If they are in the correct order, leave them. If items are not in the correct order, swap them. In the example above '5' is larger than '1', so they need to be swapped.

Step ②: Repeat the step 1 process for items '2' and '3' in the list. In the example above '5' is smaller than '6', so they do not need to be swapped.

Step ③: Continue applying the step 1 process to the rest of the items in the list.

Step ④: Once one pass is completed, repeat the process again and again until no swaps are required and all the numbers are in the correct order.

An example (visualization) of Bubble Sort (source: bournetocode.com/projects)
Questions: If we use a SIMD computer with as many processing units as you would need (no limit on the number of the PU’s):

(a) Is it possible to improve the algorithm complexity of bubble 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 bubble 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 are your responsibility. 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.