Case Study #2 - Sorting Algorithms: Stack-Based and Queues-Based

In this case study, you will explore your creativity and wits to design a sorting algorithm that will sort/arrange numbers (both in ascending and descending order) following **stack behaviour** – that is first in last out (FILO) approach. Your algorithm here cannot be combined nor use other data structures except additional stacks - This means the algorithm must be purely following stack behaviour in nature and cannot be combined with array/list manipulation, queue manipulation or combination of it. You are free to use algorithms from literature if you are able to find one (i.e., books, journals, conference papers). Proper citation and discussion must be provided as part of your documentation deliverable for this assignment.

Similarly you are to design a sorting algorithm that will sort/arrange numbers (both in ascending and descending order) following **queue behaviour** – that is first in first out (FIFO) approach. The constraints in designing this algorithm follow the constraints stated in the previous paragraph (i.e., the designed algorithm must not be combined with other data structures except in this case, additional queues).

For both proposed algorithms, you are expected to document not only your design but as well as your observations on this type of sorting algorithms. Your discussion should include the following (**but not limited to**):

(a) Discuss how your proposed stack-based and queue-based sorting algorithms work. Run through your algorithm under different cases (i.e, to sort a set of data arranged in ascending order; sort a set of data arranged in descending order; sort a set of data arranged not in any particular order) and provide a step by step trace or illustration on the crucial points of your algorithm as to how sorting process is achieved. It is also important that every time you test &/or trace your algorithm, different data size must be considered (i.e., sort a list having 6 elements; or 10 elements; or 20 elements; or 100 elements; and so on).

(b) Given the different experiments you will do in (a), determine the efficiency of your sorting algorithms. Explain &/or elaborate your analysis.

(c) Perform comparison and contrast type of analysis between your proposed stack-based and queue-based algorithms and classic sorting algorithms discussed in class (i.e., selection sort, bubble sort, among others).

(d) If you are to reflect clearly, what advantages and disadvantages can be seen using stack-based sorting algorithm or queue-based sorting algorithm?

Bonus:
Additional 20 points will be given if correct implementation and visualization of your proposed algorithms (10 points for stack-based sorting algorithm; and 10 points for queue-based sorting algorithm) are made.

General Requirements:

1. You are to submit all the works did in this case study in **both printed document** (only the documentation part) and **electronic form** (documentation in PDF and source codes[1] either in Java or in C) on or before the specified due date. **Submission of both versions must be done on the same day**. In the event that one of the requirements is submitted at the latter date, that date will be considered as the submission date.
2. Late submission will be penalized at the rate of 10% per day. That is, submission is marked out of 90% for 1 day late, 80% for 2 days late, etc. For example: An assignment is due on Friday but handed in on the next Monday morning is marked out of 70%.
3. Students have the option not to work on this case study, thus automatically receive a grade of 0 for the case study task.
4. A signed declaration of work is needed in both printed and electronic versions.
5. For the **<u>electronic version of your assignment</u>**: use the concatenated last names as filename and .c or .java[2] as the extension (as an attachment or place the source codes as part of your email message content). For example if your last names are CRUZ, and TAN, the file name should be cruz_tan.c (for C program) or cruz_ tan.java (for Java program). PDF version on the tables and other needed documents. Other formats apart from your source codes (.c or .java) and PDF will not be accepted and recognized. If you used libraries that are not part of the .C and .Java package, they have to be submitted as well as attachments. **Do not submit or attach your program's executable file version**.
6. Email the electronic version to **projectssubmission@gmail.com** with **subject heading as: DASALGO-CS2-<Lastname of Person 1 (ID#)>-<Lastname of Person 2 (ID#)>** For example: DASALGO-CS2-Cruz(100893454)-Santos (100452323)
7. **Lastly, at the last page of your documentation, you are to list down each of the members' names and their exact contribution to this assignment.**

Submission Date: **February 24 (Tuesday)**, no later than **3.00pm**. This is true for both printed and electronic form. You have the option to forward your printed version to Ms. Maricar, ST secretary of directly to me. ☺

[1] if applicable
[2] if applicable