ESO207 Assignment-2

Submission Deadline: Oct. 5, 2020

Instructions

- The assignment is open book and open notes.
- Use of internet or some discussion with other students on how to solve the problem is acceptable.
- The submission however, should be written by you alone and not copied from anywhere. Any instance of copying literally (as opposed to using the same method to solve the problem), if caught, will be penalized severely.
- You are expected to solve all questions.
- In order to conserve our manpower resources, we may decide to grade only a subset of questions. The same subset will be graded for all the students and marks will be scaled appropriately.
- In programming questions, by your preferred language, we mean any one of the four languages: C, C++, Java or python.

Q1(marks 10+20+10) In lectures we saw implementation of a queue using an array. Suppose instead of an array you are given two empty stacks, $S1, S2$. You may operate on $S1, S2$ using stack operations only, you have no access to implementation of $S1, S2$.

(a) Describe a strategy to design a queue using $S1, S2$ and some integer variables.

(b) Write pseudo-code for queue operations using your strategy. What is the complexity of various queue operations, in terms of the number of elements in the queue?

(c) Briefly argue for correctness of your implementation.

Q2(marks 5+15+10) In lecture 9, we saw how to convert recursive procedure for prefix (also called pre-order) traversal of a binary tree to a non-recursive procedure using a stack.

(a) Write pseudo-code for a recursive procedure for inorder traversal of a binary tree.

(b) Now convert the recursive procedure in part (a) to an equivalent non-recursive procedure using a stack. Write the pseudo-code of your non-recursive procedure.

(c) Argue for correctness of your code in part (b).

Q3(marks 20+5+25)

(a) Extend the idea in Q2, to write pseudo-code of non-recursive procedure for merge-sort using stacks.

[Note that the code should be derived from the recursive version of merge sort. Unrelated non-recursive procedure for merge-sort will not get any marks].

(b) Implement a stack in your preferred language.

(c) Using (b), implement the pseudo-code of part (a) in your preferred language.

——————- x–x–x ——————