FAST- National University of Computer & Emerging Sciences, Karachi.
School of Computing,
Mid Term II Examination, Spring 2021
20th April, 2021, 11:15 am – 12:15 pm

Course Code: CS-218	**Course Name:** Data Structures
Instructors: Anam Qureshi, Syed Zain ul Hassan, Basit Ali, Farah Sadia	
Student Roll No:	**Section:**

Instructions:

- Except your Roll No and Section, DO NOT WRITE anything on this paper.
- Return the question paper with your answer sheet.
- Read each question completely before answering it. There are 4 **questions on 2 pages.**
- In case of any ambiguity, you may make assumption but your assumption must not contradict any statement in the question paper.
- All the answers must be solved according to the SEQUENCE given in the question paper, otherwise points will be deducted.

Time Allowed: 60 minutes **Maximum Points**: 50

Short Q/A	
Question No. 1	**[Time: 15 minutes] [Points: 10]**

a) Why implementation of a queue using fixed-size arrays is inefficient in terms of space utilization? Is there a way to make a fixed-size array implementation of queues more space efficient? If yes, then how?

Solution: Because empty spaces are left in an array if we perform dequeue operation. In order to make it efficient we need to use circular queue.

b) Suppose you are asked to choose a sorting algorithm for very large number of records in a gigantic database. Each record consists of first name, last name and telephone number. The sorting is to be done according to first name. You have the option of choosing either A1, which is an in-place, unstable algorithm or A2, which is a stable but not in-place algorithm, or none of these two algorithms. Would you choose one of algorithms A1, A2 or neither? Justify your answer with a reason.

Solution: A2 can be used. It should have been stable. As same name may have different last name and telephone number.

c) Consider a binary tree with a height of 13 and size of 16382. Is this a perfect binary tree? Is it a full tree? Justify your answer.

Solution: Full Binary Tree (Formula 2*(2^h -1))

d) What would happen if we replace stacks with queues in function call frames/activation records?

Solution: Multiple answers

e) Why does binary search works better than linear search?

Solution: Because in each iteration half of the search space is reduced.

Elementary Sorting Algorithms	
Question No. 2	**[Time: 10 minutes] [Points: 10]**

You are running a library catalog. You know that the books in your collection are almost in sorted ascending order by title, with the exception of one book which is in the wrong place. You want the catalog to be completely sorted in ascending order.

a. Which sorting algorithm do you intend to use and why?
b. Write the code for that particular algorithm.

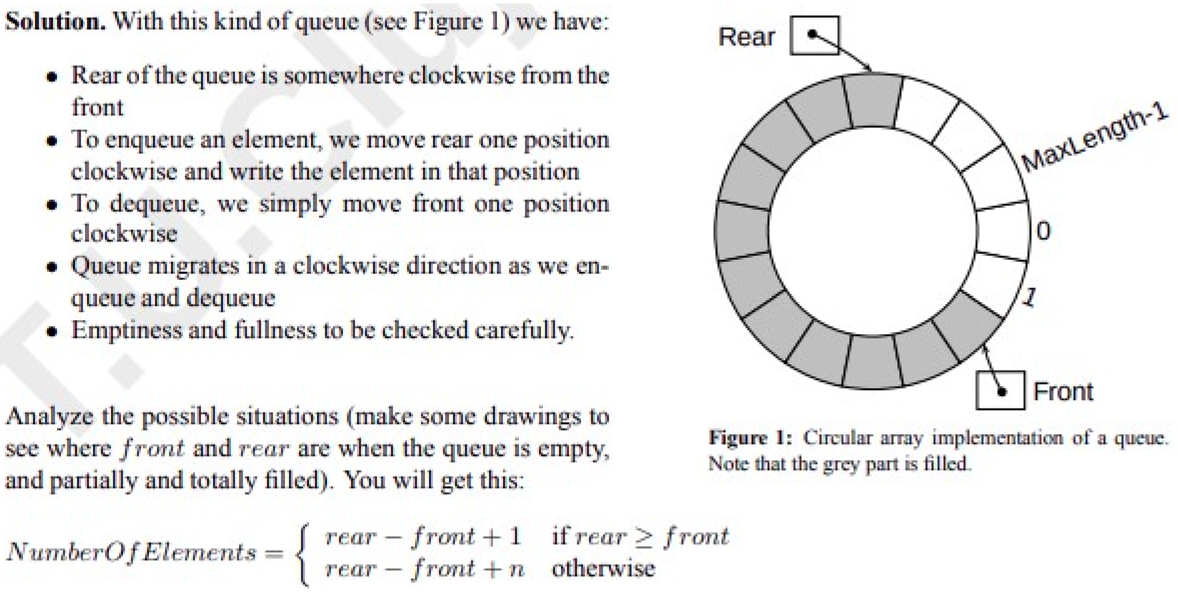
Solution. Use selection sort because it minimizes the number of exchanges.

Stack and Queues	
Question No. 3	**[Time: 25 minutes] [Points: 20]**

a) Consider a empty stack of integers. Let the numbers 1,2,3,4,5,6 be pushed on to this stack only in the order they appeared from left to right.Let S indicates a push and X indicate a pop operation. Can they be permuted in to the order 325641(output) and order 154623(output)? (Hint: SSSSSSXXXXXX outputs 654321)

Solution: SSSXXSSXSXXX outputs 325641. 154623 cannot be output as 2 is pushed much before 3 so can appear only after 3 is output.

b) A queue is set up in a circular array A[0..n - 1] with front and rear defined as usual. Assume that $n - 1$ locations in the array are available for storing the elements (with the other element being used to detect full/ empty condition). Give a formula for the number of elements in the queue in terms of *rear, front*, and *n*.



Solution. With this kind of queue (see Figure 1) we have:

- Rear of the queue is somewhere clockwise from the front
- To enqueue an element, we move rear one position clockwise and write the element in that position
- To dequeue, we simply move front one position clockwise
- Queue migrates in a clockwise direction as we enqueue and dequeue
- Emptiness and fullness to be checked carefully.

Analyze the possible situations (make some drawings to see where $front$ and $rear$ are when the queue is empty, and partially and totally filled). You will get this:

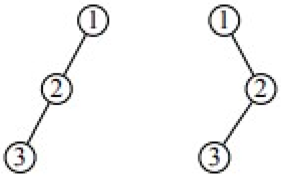
$$NumberOfElements = \begin{cases} rear - front + 1 & \text{if } rear \geq front \\ rear - front + n & \text{otherwise} \end{cases}$$

Figure 1: Circular array implementation of a queue. Note that the grey part is filled.

Binary Trees	
Question No. 4	**[Time: 10 minutes] [Points: 10]**

Construct the two binary trees in such a way that both the trees have pre-order (1,2,3) and post-order (3,2,1).

Solution:



******************** Good Luck *******************