Birla Institute of Technology and Science, Pilani

2nd Semester 2022-2023

CS F211 – Data Structures and Algorithms - Mid Semester Test (Open Book)

===

Date & Time: 14 Mar 2023, 4:00 PM **Max. Marks: 38 Marks** **Max. Time: 90 mins**

===

1. Given the pseudo code of selection sort algorithm for an array of numbers.

```
procedure selection sort
    // list  : array of items
    // n     : size of list
    for i = 1 to n - 1
        min = i
        for j = i+1 to n
            if list[j] < list[min] then
                min = j
            end if
        end for
        if indexMin != i  then
            swap list[min] and list[i]
        end if
    end for
end procedure
```

a. Write the ***pseudo code*** for the recursive version of the above algorithm. **3M**
b. Write the recurrence relation for its time complexity. **1M**
c. Solve the recurrence relation to derive its time complexity. **2M**

2. Devise a tail-recursive procedure **trfib(n)** that computes nth Fibonacci number. ***Write its pseudo code only.*** **4M**

3. Given a string **s** of size **n** containing just the characters **'(', ')', '{', '}', '[' and ']'**.
An input string is valid if:

Open brackets must be closed by the same type of brackets.
Open brackets must be closed in the correct order.
Every close bracket has a corresponding open bracket of the same type.

a. Write an algorithm **isValid(string s)**, that determines if **s** is valid or not. **Your algorithm should be the most efficient in terms of running time**. You can use an appropriate data structure to make it efficient. ***Write pseudo code only*.** **4M**
b. What is the time complexity of the above algorithm? **1M**
c. What is the space complexity of the above algorithm? **1M**

4. Given Queue ADT with operations: ***enqueue*** (insertion at rear), ***dequeue*** (removal from front), ***size*** (size of queue), ***isEmpty*** (checks if queue is empty or not). Use the Queue ADT (as a blackbox) to implement the operations of Stack ADT: ***push*** (insertion at front), ***pop*** (removal from front), ***size*** (size of stack), ***isEmpty*** (checks if stack is empty or not). ***Write your answer as pseudo code (only) for the operations of the stack.*** **6M**

5. Given an unsorted array of integers **arr**, write a **linear time algorithm** that computes the length of the longest consecutive elements sequence in **arr**. ***Write your answer as a pseudo code only.*** Follow the sample example given below: **6M**

Example:
Input: arr = [99,5,250,2,4,3,6]
Output: 5
Explanation: The longest consecutive elements sequence is [2, 3, 4, 5, 6]. Therefore, its length is 5.

6. Given an array **arr** containing n + 1 integers where each integer is in the range [1, n] inclusive. For simplicity assume that ***n <= 50***. There is only one repeated number in **arr**. Design a *linear time algorithm* to find that repeated number. Your algorithm should not use more than O(1) extra space. ***Write your answer in bulleted points only.*** **6M**

7. Given an evenly sized array containing equal number of positive and negative integers. Design a *linear time algorithm* to rearrange the numbers of the array such that the positive and negative numbers occupy alternate positions in the array. Your algorithm should not use more than O(1) extra space. ***Write your answer in bulleted points only.*** **4M**