UNIVERSITY OF LONDON

BSc EXAMINATION 2019

For Internal Students of
Royal Holloway

DO NOT TURN OVER UNTIL TOLD TO BEGIN

CS2860: Algorithms and Complexity
CS2860R: Algorithms and Complexity – PAPER FOR FIRST SITS/RESIT CANDIDATES

Time Allowed: **TWO hours**

Answer ALL questions
Calculators are NOT permitted

1. Describe the order of growth of the following functions in standard Θ notation as simply as possible. Sort the resulting Θ expressions by increasing speed of growth. [10 marks]

   (a) $f(n) = 12n + 5n^2$

   (b) $f(n) = 2n + 30 \log n$

   (c) $f(n) = 15 \log n + 17n + 3n \log n$

   (d) $f(n) = 35 + 12n$

2. Consider the functions `fnA` and `fnB` listed below. Compute an estimate of the execution time of each function as a function of n. Give your answer in Big-Oh notation and justify your answer.

```
int fnA(int n) {
    int sum=0;
    for (int i=0; i<n; i++) {
      sum += i;
    }
    return sum;
}
```

```
int fnB(int n) {
    int sum=0;
    int size=n;
    while (size > 1) {
      sum += fnA(size);
      size = size / 2;
    }
    return sum;
}
```

[12 marks]

3. This question is about linked lists. You may assume either single-linked lists or double-linked lists in your answer, as you prefer.

   (a) Show the structure of a linked list containing the values $1, 3, 4, 5, 7$ in this order. [3 marks]

   (b) Show the operations involved when an algorithm searches for the number 3 in the above linked list. Also state the worst-case running time (in Big-Oh notation) for searching for a number in a linked list with n items. [3 marks]

   (c) Show the operations involved when deleting the number 4 from the above linked list. Also state the running time (in Big-Oh notation) for deleting a number from a linked list with n items. [4 marks]

4. (a) State the *best*, *worst*, and *average case* running times for the sorting algorithms *merge sort* and *insertion sort*, when sorting n items. [6 marks]

   (b) Suggest one type of list of size n where insertion sort should be expected to be faster than merge sort, and suggest a different type of list where merge sort should be expected to be faster than insertion sort. Justify your suggestions. [4 marks]

5. (a) Insert the following items into a BST (binary search tree), in the following order: $[10, 5, 13, 3, 11, 1, 4]$. Show the result. [4 marks]

   (b) State the balance condition for AVL trees (i.e., the condition that guarantees that AVL trees are well-balanced). [4 marks]

   (c) At which point in the above insertion sequence does the BST first become non-balanced (i.e., after how many items have been inserted)? Justify your answer. [4 marks]

6. Below is shown Java code which uses an unspecified data structure `DS` to find the k largest items in a list. The data structure supports the following operations: *insert(x)*, to insert a new item into the data structure; *size()*, to report the number of items contained in the data structure; and *deleteMin()*, to delete the smallest item currently contained in the data structure.

   (a) What are the worst case running times for *insert(x)* and *deleteMin()* for a standard array (e.g., ArrayList, with new items added at the end) and a binary heap? [6 marks]

   (b) What is the worst-case running time for n items, as a function of n and k, if `DS` is implemented by storing items without any particular order, in a standard array? [7 marks]

   (c) What is the worst-case running time if `DS` is implemented using a binary heap? [5 marks]

```
DS maxK(List<Integer> items, int k) {
   DS output = new DS();
   for (int x : items) {
      output.insert(x);
      if (output.size() > k) {
         output.deleteMin();
      }
   }
   return output;
}
```

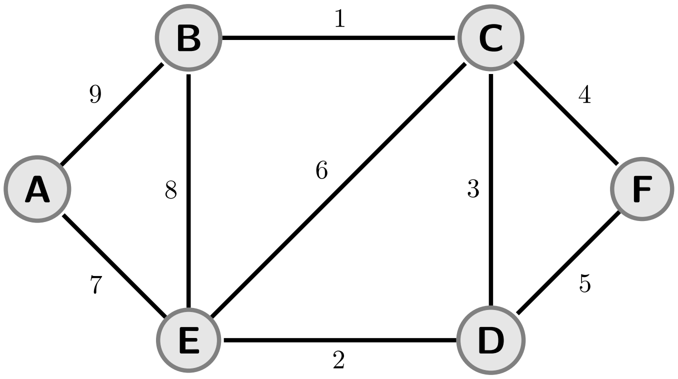

Figure 1: Graph for the Min Cost Spanning Tree question, with edge lengths shown

7. (a) Describe in your own words Prim's and Kruskal's algorithms from the course for finding a Min Cost Spanning Tree in an edge-weighted graph. Illustrate the algorithms by showing their execution on the graph above (in Figure 1). [8 marks]

   (b) Describe how to get an efficient implementation of the algorithms (running time $O(m \log m)$ for a graph with m edges), using data structures from the course. [8 marks]

8. Propose a solution to the following problem: You are writing a simple "events handler," which has the following two functionalities: (i) add an event, to be executed at a specified time, and (ii) find the next closest event in time that is to be executed (and if its time is up, return it and forget it). It is expected that the application will be run in a system with high "load", i.e., a large number of events, with events arriving frequently, so an efficient implementation is important.

   Suggest a possible efficient implementation of this system, and describe both the data structures you will use and a rough outline of the implementation of the functionalities (i) and (ii) as above.

   Also state the running times of operations (i) and (ii) in Big-Oh notation when the system contains n events. [12 marks]

END