UNIVERSITY OF LONDON

BSc EXAMINATION 2017

For Internal Students of
Royal Holloway

DO NOT TURN OVER UNTIL TOLD TO BEGIN

CS2860: Algorithms and Complexity

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) $f(n) = \log n + 7n$

   (b) $f(n) = 5n + 2n \log n$

   (c) $f(n) = 2n^2 + 3n \log n + 1$

   (d) $f(n) = 3n + 4$

2. Describe what it means that a function $f(n)$ (for example, the running time of an algorithm) is $O(g(n))$. What do the terms *constant factors* and *lower-order terms* refer to in this context, and in which sense does the Big-Oh notation ignore them? [4 marks]

3. 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. [12 marks]

```
int fnA(int n) {
  int sum=0;
  for (int i=0; i<n; i++) {
    sum += i;
  }
  for (int i=1; i<=n; i++) {
    sum *= i;
  }
  return sum;
}


int fnB(int n) {
  int j=1;
  int result=0;
  while (j <= n) {
    result += fnA(j);
    j *= 2;
  }
  return result;
}
```

4. Recall the mergesort sorting algorithm.

   (a) Illustrate the execution of mergesort when sorting the array $[2, 5, 3, 7, 4, 1, 9, 6]$. Draw the tree of recursive calls, and write in each node of the tree the arguments of the call to the `merge` procedure made in that node. [6 marks]

   (b) The mergesort algorithm is an example of the *divide and conquer* algorithm design strategy. Describe the divide-and-conquer design strategy and its different parts, and describe how mergesort is an example of it. [5 marks]

   (c) Formulate a recurrence relation describing the asymptotic running time of mergesort when sorting an array on n elements. Explain the connection between the parts of the recurrence relation and the mergesort algorithm. [4 marks]

   (d) State the asymptotic worst-case running time of mergesort for n elements in Big-Oh notation, and justify your answer (for example, using the recurrence relation from question 4c). [4 marks]

5. The following questions are about binary search trees (BSTs).

   (a) Insert the following numbers into an initially empty binary search tree, in the given order, and show the resulting tree: 4, 2, 9, 7, 3, 8. [6 marks]

   (b) Perform the following operations, in the given order, and show the shape of the tree after each one: Delete 3. Delete 7. Delete 4. [4 marks]

   (c) Suggest an order of insertion of the numbers from 1 to 7 such that the resulting BST is *not* balanced, according to the balance condition used in AVL trees. Justify your claims. [3 marks]

6. Consider the problem `countCommonItems`: The input is two arrays A and B of integers, and the task is to count the number of integers that occur in both lists. For example, on inputs $[1, 6, 3, 7, 5]$ and $[2, 9, 7, 4, 3]$ the answer is 2, because 3 and 7 occur in both lists but no other number occurs in both lists.

   (a) Describe an algorithm for the problem. For full marks, your algorithm should be as efficient as possible (ideally average case running time $O(n)$). [8 marks]

   (b) State the running time of your algorithm, assuming that both input arrays have length n. Justify your answer. [4 marks]

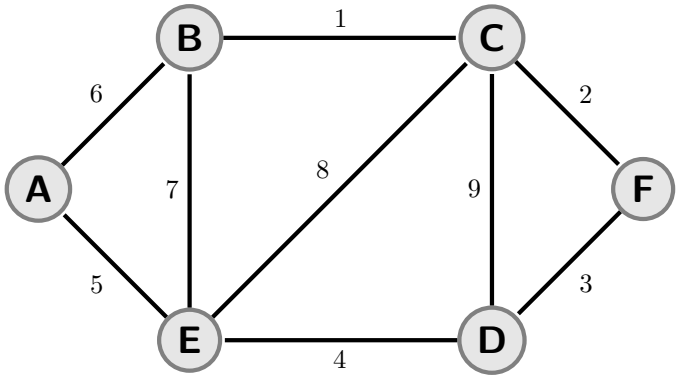

Figure 1: Graph for Dijkstra's algorithm with edge lengths shown

7. (a) Describe Dijkstra's algorithm for finding single-source shortest paths in a graph, and illustrate it on the graph of Figure 1, by finding shortest paths to all destinations starting from vertex C. [6 marks]

   (b) Draw the resulting shortest paths tree. [4 marks]

   (c) Explain how to get an efficient implementation of Dijkstra's algorithm, i.e., an implementation that runs in time $O(m \log n)$ or better for a graph with m edges and n vertices. Justify your claim about the running time. [4 marks]

   (d) Name a potentially faster algorithm for the single-source shortest paths problem for the case that all edges have the same length, and name an algorithm that can solve the same problem on directed graphs if some edges have negative lengths. [4 marks]

8. Consider the following problem: You have a large collection of music files (e.g., MP3 files), and for each file you know the artist name (as a string) and the song title (also as a string). You wish to find the name of the artist who has the largest number of songs featured in the collection.

   (a) Describe an algorithm for this problem. For full marks, the algorithm should be as efficient as possible. [8 marks]

   (b) State the running time of your algorithm, assuming that there are in total n songs, across t different artists. [4 marks]

END