UNIVERSITY OF LONDON

BSc EXAMINATION 2018

For Internal Students of
Royal Holloway

DO NOT TURN OVER UNTIL TOLD TO BEGIN

CS2860: Algorithms and Complexity
CS2860R: Algorithms and Complexity – PAPER FOR 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. [12 marks]

   (a) $f(n) = 19n + n \log n$

   (b) $f(n) = 1000n$

   (c) $f(n) = 2n^2 + 3n(\log n)^{10} + 50n$

   (d) $f(n) = 10 \log n + n$

2. Show the execution of the `quicksort` sorting algorithm on the array `A=[5, 3, 8, 9, 2, 1, 4]`. Assume an implementation that uses the last element as pivot. Illustrate all recursive calls made by the algorithm. [8 marks]

3. Consider the function `fnA` listed below. Compute an estimate of its execution time as a function of n. Give your answer in Big-Oh notation and justify your answer. [10 marks]

```
int fnA(int n) {
  int sum=0;
  for (int i=0; i<n; i++) {
    int j=i;
    int product=1;
    while (j>1) {
      product *= j+1;
      j = j / 2;
    }
    sum += product;
  }
  return sum;
}
```

4. Although merge sort runs in $O(n \log n)$ worst-case time and insertion sort runs in $O(n^2)$ worst-case time, the constant factors in insertion sort make it faster for small n. Thus, it makes sense to use insertion sort within merge sort when sub-problems become sufficiently small. Consider a modification to merge sort in which n/k sublists of length k are sorted using insertion sort and then merged using the standard merging mechanism, where k is a value to be determined.

(a) Show that the n/k sublists, each of length k, can be sorted by insertion sort in $O(nk)$ worst-case time. [4 marks]

(b) Show that the sublists can be merged in $O(n \log(n/k))$ worst-case time. [8 marks]

(c) How should k be chosen in practice? [4 marks]

5. Suppose that a certain binary search tree has keys that are integers between 1 and 20, and we search for 10. Which sequence below *cannot* be the sequence of keys examined? Justify all your answers. [10 marks]

(a) 20, 19, 16, 15, 12, 9, 10

(b) 7, 18, 15, 12, 10

(c) 1, 12, 7, 16, 4, 10

6. Which of the following are valid heaps? Draw the corresponding trees. Justify your answers.

$$A = [2, 4, 12, 14, 11, 18] \qquad B = [1, 8, 6, 9, 7, 11]$$

[6 marks]

7. Propose data structures to use for the following situations. Justify your choices. You do not need to give complete code, but it should be clear how your solution is intended to work.

(a) You are given a long list of books (with author names). You want to find the author who has written the largest number of books in the list. What running time do you achieve with your choice? (You do not need to prove this claim.) [10 marks]

(b) You are writing a simple "event manager," which handles events which are to be executed at specified times. New events arrive frequently, each with an associated time when it is to be executed, and it is important to be able to keep track of the next closest event to execute. What structure do you choose to store the event information? What is the time required for the two basic operations (add a new event, and find the next event to execute)? [8 marks]

8. Give definitions of a connected graph and connectivity components. DFS is applied to find connectivity components in the graph G of Figure 1. Assume that in the loops of DFS the vertices of G are considered in the natural order. Give the order in which the vertices of G are visited. [8 marks]

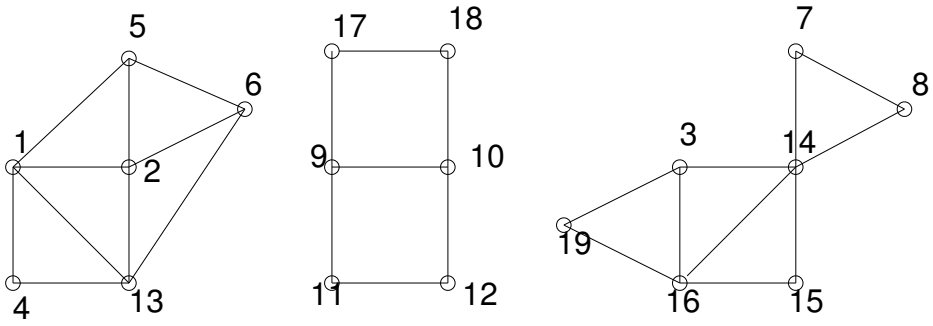

Figure 1: Disconnected graph G

9. Suggest an algorithm for a search engine that lets you plan travel connections between cities. Its input is a list of possible connections, where each connection is given as a combination of a starting time, an ending time, a starting location and a destination location. (For example, one such connection could be "a train departing from London at 08:48, arriving in Manchester 10:55".)

   (a) Describe an algorithm for finding a fastest journey from one city to another, with a given starting time. Note that a journey can consist of many connections. [8 marks]

   (b) Explain how to adjust your algorithm to accommodate minimum *changeover times*, e.g., when switching from one train to another at least a 20-minute margin should be given, and when switching from one plane to another a margin of at least one hour is required. [4 marks]

END