UNIVERSITY OF LONDON

GOLDSMITHS COLLEGE

Department of Computing

B. Sc. Examination 2016

IS52017C(Resit)
Algorithms

Duration: 2 hours and 15 minutes

Date and time:

There are five questions in this paper. You should answer no more than three questions. Full marks will be awarded for complete answers to a total of three questions. Each question carries 25 marks. The marks for each part of a question are indicated at the end of the part in [.] brackets.

There are 75 marks available on this paper.

No calculators should be used.

**THIS PAPER MUST NOT BE REMOVED
FROM THE EXAMINATION ROOM**

Question 1

(a) Consider the two cases below, i.e. Case1 and Case2. In each case, two functions (f and g) are used to represent the time complexities of two algorithms in terms of the input size n: [8]

(Case1) $f = n(n+1)$, and $g = 2000n^2 + \log n$
(Case2) $f = \log_2 2n$, and $g = 100(log_e n) + n \log_2 n$.

For each case,

i. describe the time complexities in big-O notation.
ii. indicate whether function f has the same, bigger or smaller order of growth than function g.

Show all your work.

(b) Construct and demonstrate the *trie* and *compressed trie* for the set of words ("array", "map", "apply", "middle", "method", "apple", "key", "kettle"). [6]

(c) Derive step by step a Huffman code for alphabet {A, B, C, D, E, F}, given that the probabilities that each character occurs in messages are 0.3, 0.2, 0.2, 0.1, 0.1, 0.1 respectively. Explain how a given sequence of 0s and 1s in such a message can be decoded. [11]

Question 2

(a) Demonstrate how the closed hashing algorithm works using the data set (4,2,12,3,9,11,7,8,13,18) as an input example. Assume the length of the hash table is 7 initially. You should demonstrate:

i. how the hash table can be built step by step; [5]

ii. how a search on such a hash table can be achieved in $O(1)$ time in the worst case. [5]

(b) Discuss briefly the time complexity in the *worst* case for the algorithm below. Indicate the basic operations you have counted. [5]

for $i \leftarrow 1; i < 2n; i \leftarrow i + 1$ **do**
    for $j \leftarrow 1; j < i; j \leftarrow j + 1$ **do**
        for $k \leftarrow 1; k < j; k \leftarrow k + 1$ **do**
            $x \leftarrow x + 1$
        end for
    end for
end for

(c) Explain what distinguishes a *binary search tree* from a *binary min-heap*. Draw diagrams to demonstrate, step by step, how each of the data structures can be used to store the data (5, 7, 10, 6, 1, 14, 11, 2). Assume that both the binary search tree and the binary min-heap are empty initially, and the data is added in the order given. [10]

Question 3

(a) Consider the Closest Pair Problem. Suppose that two most similarly-built children are to be selected from a class of n children. Let (w_i, h_i) represent the weight and height of child i, where $i = 1, \cdots, n$. If each of these data is plotted as a point (w_i, h_i) in a x-y plane, the closest pair of these points can be used to represent the two most similarly-built children.

i. Draw a diagram to describe an instance of the problem of 3 children and plot their height (vertical y-coordinate) against weight (horizontal x-coordinate). [4]

ii. Define a suitable data structure to store n weight-height points, i.e. the (w_i, h_i)s of n children, where n is an integer greater than 2. [4]

iii. Describe how to measure the similarity between two children in terms of their weight and height. [5]

iv. Devise an algorithm to find the two most similarly built children, using the data structure proposed in part a.(ii). Outline your algorithm in pseudocode or as a flowchart. [7]

v. Demonstrate how to compute the time complexity of your algorithm in part a.(iv) for the worst case. [5]

Question 4

(a) Explain the two-step approach in a *proof by mathematical induction.* [5]

(b) Prove Cassini's identity using induction: [8]

$$F(n+1)F(n-1) - [F(n)]^2 = (-1)^n$$

where $n \geq 1$, and $F(n)$ is the Fibonacci sequence with $F(0) = 0, F(1) = 1$, and $F(n+1) = F(n) + F(n-1)$.

(c) Fill the missing words/sentences in each gap below: [8]

NP class is the class of ____(1)____ problems which can be solved in ____(2)____ time by a ____(3)____ computer.

NP-complete is the term used to describe ____(4)____ problems that are the ____(5)____ ones in NP class in the sense that, if there were a ____(6)____ algorithm for a NP-complete problem, then there would be a ____(7)____ algorithm for ____(8)____ problem in NP.

To prove a new problem is NP-complete, we first prove that ____(9)____, then we prove that ____(10)____.

(d) Name two well-known NP-complete problems. [4]

Question 5

(a) A sorting algorithm is regarded as *stable* if it maintains the relative order of identical data. Demonstrate that the selection sort can be stable using the data (5, 2, 4, 6, 2, 7) as an input example. [5]

(b) Demonstrate and explain why the number of comparisons is of $O(n^2)$ for the selection sort. [6]

(c) Demonstrate and explain why selection sort is not efficient for a nearly sorted positive integer list using the data (1, 2, 4, 6, 3, 7) as an input example. [7]

(d) Outline, in diagrams, how the Heapsort algorithm works, using the data (1, 2, 4, 6, 3, 7) as an input example. [7]