UNIVERSITY OF LONDON

GOLDSMITHS COLLEGE

Department of Computing

B. Sc. Examination 2017

IS52038A/B
Algorithms and Data Structures

Duration: 2 hours 15 minutes

Date and time:

This paper is in two parts: part A and part B. You should answer ALL questions from part A and TWO questions from part B. Part A carries 40 marks, and each question from part B carries 30 marks. The marks for each part of a question are indicated at the end of the part in [.] brackets.

There are 100 marks available on this paper.

Electronic calculators must not be programmed prior to the examination. Calculators which display graphics, text or algebraic equations are not allowed.

THIS PAPER MUST NOT BE REMOVED FROM THE EXAMINATION ROOM

Part A

Question 1

(a) The terms *abstract data structure* and *abstract data type* are often used interchangeably. Explain briefly when the term *abstract data structure* is more appropriate and when *abstract data type* is more appropriate to use. [4]

(b) Draw a flowchart or write in pseudocode a recursive algorithm `fibonacciTerm(n)` that returns the nth term of the Fibonacci sequence $1, 1, 2, 3, 5, 8, 13, 21, \cdots$. For example, the first term `fibonacciTerm(1)`$= 1$, and the fourth term `fibonacciTerm(4)`$= 3$. [5]

(c) Consider below the adjacency matrix of a graph. Write the immediate neighbours for each vertex. Draw the graph represented by the matrix. [5]

	A	B	C	D	E
A	0	0	0	0	1
B	1	0	0	0	0
C	0	0	0	0	0
D	0	0	0	0	0
E	0	1	1	1	0

(d) Draw the *trie* and *compressed trie* for the set of words (“array”, “map”, “apply”, “middle”, “method”, “apple”, “key”, “kettle”). [6]

(e) Consider the algorithm below, to determine whether a number k is present in an array A:

Require: A :: array
Require: k :: number
  result ← false
  start ← 0
  end ← LENGTH(A)
  while start < end **do**
    mid ← $\left\lfloor \frac{\text{start+end}}{2} \right\rfloor$
    if A[mid] = k **then**
      result ← true
    else if A[mid] < k **then**
      start ← mid + 1
    else
      end ← mid
    end if
  end while
  return result

i. What conditions are required for this algorithm to operate correctly – that is, to return true if the number is present in the array, and false otherwise? [2]

ii. What are the best- and worst-case additional space complexities of this algorithm in terms of L, the length of the array? [2]

iii. What are the best- and worst-case time complexities of this algorithm in terms of L, the length of the array? [2]

iv. Suggest an improvement to this algorithm. [2]

(f) A divide-and-conquer algorithm for multiplication of numbers of size N does five multiplications of size $\frac{N}{3}$, along with a constant number of additions and subtractions.

i. Write the recurrence relationship expressing this statement. [2]

ii. Draw the recursion tree to illustrate your answer to part f.(i). [3]

iii. What is the complexity of this divide-and-conquer multiplication algorithm in terms of the size N? [3]

(g) Copy and complete the following table of values of three functions:

x	$3x^2$	$4x^2 + 7x + 6$	$5x^2$
1	3	17	5
2			
4			
8			
16			

and hence, or otherwise, argue that $4x^2 + 7x + 6 \in \Theta(x^2)$. [4]

Part B

Question 2

(a) Let X be an array of n elements. Algorithm A chooses $\log n$ elements in X at random and executes an $O(n)$ time calculation for each. Outline algorithm A in pseudocode and compute the worst-case running time of Algorithm A step by step. [4]

(b) 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. Under what condition a search on such a hash table can be achieved in $O(1)$ time and how. [5]

(c) Draw a sequence of diagrams to demonstrate, step by step, how a (i) binary search tree and (ii) binary max-heap may be constructed to store (4, 5, 6, 2, 3, 7). Assume that both data structures are empty initially and the data is input in the order given. [10]

(d) Consider a children game 'Tic-Tac-Toe' for two players X and O. The Tic-Tac-Toe (also referred to as 'wick wack woe' or 'noughts and crosses') is a pencil-and-paper game where players X and O take turns marking the spaces in a 3×3 grid that is initially blank. Assume that X goes first. The player who succeeds in placing three respective marks in a horizontal, vertical, or diagonal row wins the game.

i. Draw a partial game tree for the Tic-Tac-Toe. The partial tree should start from the (X's turn) state as shown below, and contain (i) the state where *X wins*, (ii) the state where *O wins* and (iii) the state where *Draw*. [3]

X:

X		X
	X	O
O		O

ii. Devise a heuristic function $h(s)$ that is based on each new state. Demonstrate how to compute values of the heuristic function using the above given state in (d)i. as an example. [3]

Question 3

(a) Suppose that, given a sorted list L of n elements, we want to determine whether or not there is a pair of elements a and b in the list such that $a + b = x$ for some given value of x.

For example, given a list $L = (2, 5, 6, 7, 8, 9)$ and $x = 11$, there is a pair (5,6) in the list such that $5 + 6 = 11$, where $a = 5, b = 6$ (or (2,9) such that $2 + 9 = 11$). However, given the same list L and $x = 2$, there is no such a pair.

i. Design an $O(n^2)$ algorithm to determine whether such a pair (a, b) exists. Show all your work. [4]

ii. Design an $O(nlogn)$ algorithm to determine whether such a pair (a, b) exists. Show all your work. [5]

iii. Discuss the correctness of your algorithms and analyse its complexity. You may use a small example to ease your discussion and analysis. [6]

(b) How many edges does an undirected simple graph have if it has 5 vertices of degrees 4, 3, 3, 3 and 1 for (A, B, C, D and E), respectively? Devise an adjacency matrix for representation of your graph. [5]

(c) The distances between pairs of vertices is given by the following table, **including** between pairs for which there is no edge in the graph:

	A	B	C	D	E
A		5	7	2	1
B	5		2	2	3
C	7	2		6	4
D	2	2	6		7
E	1	3	4	7	

Showing your working, use Dijkstra's algorithm to compute the length of the shortest path (using **only** edges in the graph from your answer to part (b)) from vertex E to vertex C. [6]

(d) Briefly describe how Dijkstra's algorithm can be improved on if information about straight-line distances between vertices is given. [4]

Question 4

The following algorithm performs string search for the first occurrence of a pattern (needle) within a text (haystack).

```
function SEARCH(needle,haystack)
    hpos ← 0
    while hpos ≤ LENGTH(haystack) - LENGTH(needle) do
        found ← true
        for 0 ≤ i < LENGTH(needle) do
            if needle[i] ≠ haystack[hpos+i] then
                found ← false
                break
            end if
        end for
        if found then
            return hpos
        else
            hpos ← hpos + 1
        end if
    end while
    return false
end function
```

(a) In terms of the length of the needle m and the length of the haystack n, what are the best- and worst-time computational complexities of this algorithm? Explain your answers, giving an example of each of the situations leading to best- and worst-time for this algorithms. [6]

(b) Describe in detail how string search can be improved by a suitable preprocessing of the pattern (needle). Give in your answer a description of the preprocessing, the revised SEARCH algorithm, and its worst-case time complexity. [8]

(c) Describe in detail how string search can be improved by constructing a suitable data structure to represent the text (haystack). Give in your answer a description of the data structure, the revised SEARCH algorithm, and its worst-case time complexity. [8]

(d) Of the above modifications in parts (b) and (c), identify which you would apply for each of: a spelling checker; search and replace within a text document. Explain your answers. [8]