UNIVERSITY OF LONDON

GOLDSMITHS COLLEGE

B. Sc. Examination 2011

Computer Science

IS52017A Data Communication and Algorithms

Duration: 3 hours

Date and time:

This paper is in two parts: part A and part B. There are a total of three questions in each part. You should answer two questions from part A and two questions from part B. Your answers to part A and part B should be written in separate answer books.

Full marks will be awarded for complete answers to a total of four questions, two from part A and two from part B. 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 100 marks available on this paper.

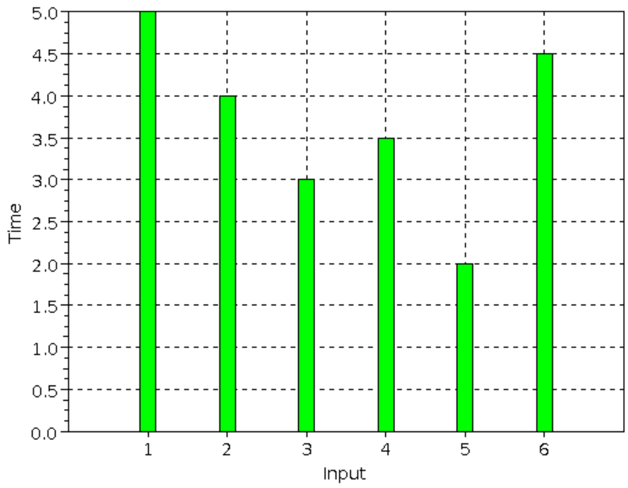
Electronic calculators must not be programmed prior to the examination. Calculators which display graphics, text or algebraic equations are not allowed.

THIS PAPER MUST NOT BE REMOVED FROM THE EXAMINATION ROOM

Part B

Question 1

(a) Consider the bar chart below showing the execution time (vertical axis) of an algorithm against possible inputs 1–6 (horizontal axis):



Referring to the bar chart as an example, explain the differences between the *worst-case performance* and the *average-case performance* of an algorithm. Calculate the execution time of the algorithm in the worst case and the time in the average case. [5]

(b) Consider the two functions below. Indicate whether $f = O(g)$, or $f = \Omega(g)$, or $f = \Theta(g)$. Justify your answer and show all your work. [5]

i. $f(n) = n3^n$
ii. $g(n) = 500n + 10$

(c) Given n objects of sizes $s_1, \cdots, s_n$, where $0 < s_i \leq 1$, the goal of the *Bin-Packing Problem* is to find the smallest number of bins into which the objects can be packed. You may assume, if appropriate, that each bin has a capacity of one in attempt of the following sub-questions: [15]

i. Draw a diagram to show one instance of the problem with 5 objects of sizes $0.4, 0.5, 0.4, 0.2$, and 1.
ii. Explain what is meant by a *greedy approach*.
iii. Devise an approximation algorithm for the bin-packing problem using a greedy approach. Show all your work.
iv. Trace the execution of your algorithm using your own example in part ci.

Question 2

Consider the scenario where eight classroom buildings (labelled $B1, \cdots, B8$) are to be connected by new pedestrian paths. The new paths need to ensure that each building can be reached from any other building via one or more paths. The cost of creating a path between each pair of the buildings is represented in the table below, where a dash '-' indicates 'no path', and the minimum cost on each row is highlighted in bold. For example, the cost of the path between B1 and B2 is 20, which can be found following the row-column index pair (B1, B2) in the table, there is no path possible between B2 and B3, and (B3, B4) is the minimum cost path among paths (B3, B4), (B3, B7) and (B3, B8). Your task is to propose an algorithm that helps decide which paths to build in order to minimise the total construction cost.

	B2	B3	B4	B5	B6	B7	B8
B1	20	**3**	25	51	16	73	18
B2	-	-	**4**	5	6	-	-
B3	-	-	**7**	-	-	9	58
B4	-	-	-	**5**	46	37	-
B5	-	-	-	-	56	17	**8**
B6	-	-	-	-	-	**72**	80
B7	-	-	-	-	-	-	**98**

i. Based on the scenario, formulate an algorithmic problem that would help complete your task. Show any mathematical models used. [4]

ii. Represent the scenario as an instance of the problem in your formulation and in mathematical terms. [4]

iii. Demonstrate how a priority queue can be constructed for the paths by tracing the first 8 execution steps of the construction algorithm. Assume that the queue is empty initially. [6]

iv. Draw the binary search tree for the paths. [5]

v. Demonstrate step by step in a diagram how Kruskal's algorithm can be applied to derive the solution to the problem in part i. [6]

Question 3

(a) Consider the design of an algorithm $mergelists(x, y)$ that merges two linked lists x and y. The data in both x and y are in ascending order. The result after a merge should be one linked list in ascending order, containing all the data in both lists. Assume the node structure is as follows and that you may modify only the *next* field of each node:

data	next	→

i. Assume that the result list is (0, 2, 3, 3, 4, 4, 5, 6, 7, 11, 14). Demonstrate an example including two instances of the problem: one must be a *general case* of the problem where x contains 5 elements, and the other one must be a *special case* of the problem. Explain your definition of the 'special case' if it is different from the one in lectures. [5]

ii. Devise and outline your algorithm $mergelists(x, y)$ in pseudocode or in a flow chart. Assume no predefined ADT is available. [9]

iii. What is the worst case time of your algorithm? Justify your answer. [4]

(b) Outline the *selection sort* algorithm in pseudocode or flowchart. Demonstrate how the *selection sort* algorithm sorts a list of integers to ascending order. Use (5, 2, 4, 6, 2, 7, 8) as an input example and explain whether or not the selection sort algorithm is *stable*. [7]