The University of Nottingham Ningbo China

Centre for English Language Education

PRELIMINARY YEAR, SEMESTER ONE
Sample Paper 2024-25

INTRODUCTION TO ALGORITHMS

Time allowed TWO hours

Candidates must write their ID number on the Answer Booklet and fill-in their attendance card but must NOT write anything else until the start of the exam is announced.

This paper contains FIVE questions. The total number of points is 50.

Answer all questions with necessary steps.

Only general bilingual dictionaries are allowed. Subject-specific dictionaries are not permitted.

No electronic devices can be used in this exam.

Do NOT open the examination paper until told to do so.

All answers must be written in the Answer Booklet.

INFORMATION FOR INVIGILATORS:
A 15-minute warning should be given before the end of the exam.
Please collect both the Question and the Answer Booklet after the exam.
Please return Answer Booklets in ID order.

This page is intentionally left blank

1. A company calculates its end of year staff bonuses according to the following table:

	Experience			
Age	**0-5 (yrs)**	**6-10 (yrs)**	**11-20 (yrs)**	**20-30 (yrs)**
20-30	5000RMB	5000RMB	5000RMB	5000RMB
31-40	5000RMB	7000RMB	10000RMB	10000RMB
41-50	5000RMB	7000RMB	7000RMB	10000RMB
51-65	5000RMB	7000RMB	10000RMB	15000RMB

(i) Write an algorithm called **`bonus(age,exp)`** that takes two positive integers representing the staff age and work experience in years respectively and returns the amount of bonus based on the table given above.

(ii) Trace your algorithm for **`bonus(45,15)`** showing all the intermediate steps.

[10]

2. (a) Write a recursive algorithm called **`length(L)`** that takes a list and returns the length of it. For example: **`length([5,3,1,4])=4, length([ ])=0`**

[2]

(b) Write a recursive algorithm called **`powerten(x)`** that takes a positive integer `x` and returns the value of 10^x. For example: **`powerten(0)=1, powerten(3)=1000`**.

[3]

(c) Write a recursive algorithm **`list2num(L)`** that takes a nonempty list `L` of positive integers and turns the list into a n-digit integer (`n` is the length of the list). For example: **`list2num([5,3])=53, list2num([9,9,9,0])=9990`**.

You may use a helper function **`listHelper()`** and should call the functions you have written in 2(a) and 2(b).

[5]

3. (a) (i) Give an example of a list with 5 elements (integers) that is considered as the *worst-case scenario* for insertion sort algorithm.

(ii) Suppose now that you have a list of numbers with length 100. How many operations would you need to sort this list via the mergesort scheme?

[2]

(b) Consider the list `L=[7,11,13,14,16,18,19,23,27,30,33]`.

(i) Why is this list suitable for a *binary search* algorithm?

(ii) Apply binary search step by step to show that `11` is in list `L`.

[4]

3. Continued ...

(c) Consider the unsorted list `T=[30,7,11,9,20,5,10,40]`. Apply the *divide & conquer* scheme of quicksort algorithm to `T` until you get a sorted list of numbers in asceding order. Draw a clear digaram indicating each step of divide & conquer (partition, sort, merge) carefully.

[4]

4. (a) (i) State the formula for the depth of a minimal binary tree with n nodes.

(ii) Convert the list `Q=[1,2,3,4,5,0,6,9,8,7,10]` to a minimal balanced BST.

(iii) How many more nodes can you add to the tree above before its depth increases?

[3]

(b) Consider the following binary tree (Fig 4.1). Determine the range of integer values that can be assigned to nodes A, B, C and D such that the binary tree in Fig 4.1 is a BST.

[2]

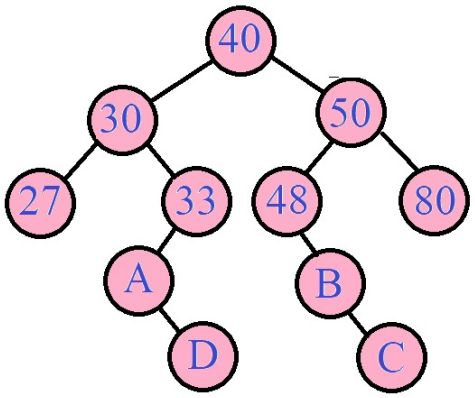

Fig 4.1

(c) Write a recursive algorithm **`inOrder(bst)`** that takes a binary search tree (with integer node values) as input and returns a sorted list of numbers. You can call the **`merge()`** algorithm. Trace your algorithm for the tree given in Fig 4.2 below. Show all intermediate steps carefully.

[5]

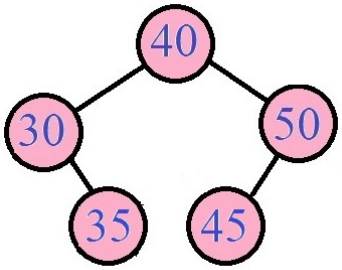

Fig 4.2

5. (a) Consider the graph given in Fig 5.1:

   (i) Is this graph complete? Give a reason why or why not.

   (ii) Redraw the graph by arranging its vertices into two separate sets and then decide whether the graph is bipartite or not.

   (iii) Does this graph have Eulerian path or Eulerian tour? Give a brief reason in each case.

[4]

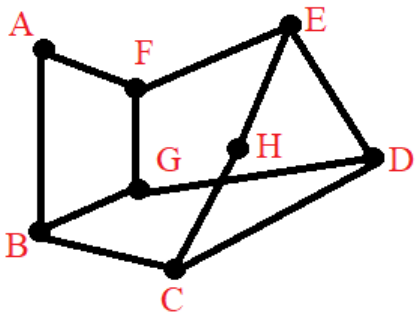

Fig 5.1

(b) Consider the graph given in Fig 5.2:

   (i) Draw one spanning tree of this graph.

   (ii) Compute the total number of spanning trees that this graph can have.

[2]

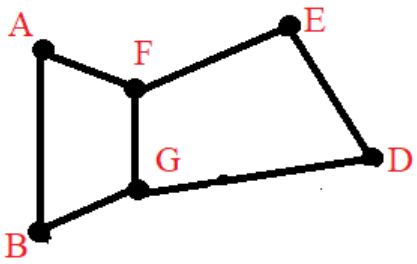

Fig 5.2

(c) (i) Use Prim's algorithm to find the minimum spanning tree for the graph in Fig 5.3. Carefully demonstrate the procedure showing detailed steps.

   (ii) compute the minimum cost of traversal in the spanning tree you obtained above.

[4]

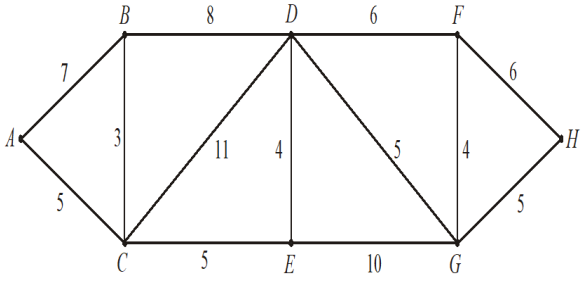

Fig 5.3

Use this page for rough work.

All answers must be written in the Answer Booklet.