PAPER CODE NO. COMP108	EXAMINER : S. Ravindran DEPARTMENT : Computer Science Tel. No. 43670

THE UNIVERSITY
of LIVERPOOL

MAY 2004 EXAMINATIONS

Bachelor of Arts : Year 1
Bachelor of Science : Year 1
Master of Mathematics : Year 2

ALGORITHMIC FOUNDATIONS

TIME ALLOWED : Two Hours

INSTRUCTIONS TO CANDIDATES

Candidates will be assessed on their best **four** answers.

If you attempt to answer more than the required number of questions (in any section), the marks awarded for the excess questions will be discarded (starting with your lowest mark).

THE UNIVERSITY
of LIVERPOOL

Question 1

1.A Prove the following statement using mathematical induction.

$$1 + 2 + \ldots + n = n(n+1) / 2$$ [10 Marks]

1.B Give, using *big oh* notation, the running time of the following loop as a function of n.

```
for i = 1  to n do
    for j = 1  to i do
        s := s + 1
```

[5 Marks]

1.C Briefly explain why *binary search* takes at most $O(\log n)$ steps to find an element in a sorted array of length n. Assume that $n = 2^k$, for some integer k. [10 Marks]

THE UNIVERSITY
of LIVERPOOL

Question 2

2.A Assume that n numbers are stored in locations A[1], A[2], …, A[n], where A is an array. We can sort the numbers as follows.

for $i = n$ downto 2 do
  Select the largest among A[i], A[i-1], …, A[1] and swap it with A[i]

In the i^{th} pass, the algorithm selects the largest element by

largenumber := A[i];
for $j = i - 1$ *downto* 1 *do*
    if (A[j] $\geq$ *largenumber*) then *largenumber* := A[j] ;

(i) Prove that this sorting algorithm is not stable.

(ii) How would you modify this algorithm to a stable sorting algorithm.

[10 Marks]

2.B The Quicksort algorithm uses the Divide-and-Conquer technique as follows.

Quicksort (A, p, r)
  If $p \neq r$ then
  begin
    Step 1: Partition the array A[p .. r] into two nonempty sub arrays A[p .. q] and A[q+1 .. r] such that each element of A[p .. q] is less than or equal to each element of A[q+1 .. r].

    Step 2: Recursively sort A[p .. q] (i.e. *Quicksort* (A, p, q))

    Step 3: Recursively sort A[q+1 .. r] (i.e. *Quicksort* (A, q+1, r))
  end.

To sort an entire array A of length n, the initial call is *Quicksort*(A, 1, n).

(i) Assume that Step 1 partitions the array A[1 .. n] into sub arrays A[1 .. i] and A[i+1.. .. n] in O(n) time. Give a recursive equation for the running time of the Quicksort algorithm.

(ii) Using the recursive equation you described in (i), briefly explain the best-case and worst-case behaviour of the Quicksort algorithm. [15 Marks]

THE UNIVERSITY
of LIVERPOOL

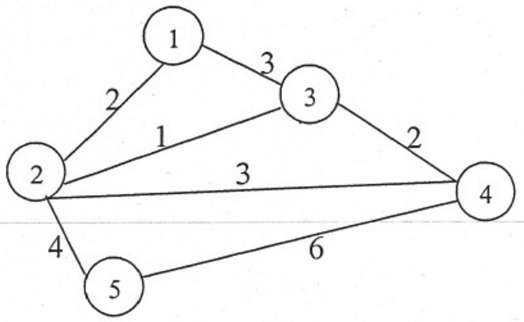
Question 3

3.A Using suitable diagrams, show how a binary search tree would be built for the following sequence of numbers.

5, 8, 10, 7, 3, 9, 1 [5 Marks]

3.B Using a diagram explain that searching for an element may take O(n) steps in an n-node binary search tree. [5 Marks]

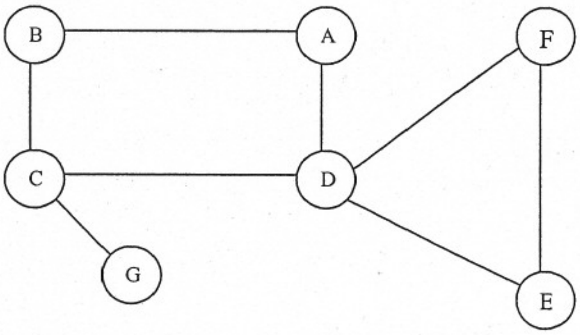
3.C In the following graph the vertices and the edges represent the cities and the direct link between the cities (if any) respectively. A cost of building a direct link is shown as the weight of an edge. We would like to minimize the construction cost and ensure that a path exists between every pair of cities. Removing some links can do this. Using a greedy algorithm, for example Prim's algorithm, to find a minimum weight spanning tree, remove those links.



[10 Marks]

3.D Given a weighted undirected graph $G = (V, E)$, let W be a proper subset of its vertices. Also let e denote the edge of smallest weight with one end in W and the other in $(V - W)$. Show that there exists a minimum weight-spanning tree of G, which contains e. [5 Marks]

Question 4



4.A Give an adjacency matrix representation of the above graph. [5 Marks]

4.B For each vertex v of the above graph compute its position in the order in which the vertices were visited in a Depth First Search ($DFI(v)$), and construct a spanning tree using the edges traversed in the Depth First Search. [10 Marks]

4.C Using the order in which each vertex w was visited in Depth First Search and the spanning tree T, describe an algorithm to find articulation points. [10 Marks]

Question 5

5.A Let D_1 and D_2 denote two decision problems. What does it mean to say that D_1 is polynomially transformable to D_2 (written $D_1 \leq_P D_2$)? [5 Marks]

5.B Prove that, if $D_1 \leq_P D_2$ and D_2 can be solved by a polynomial time algorithm, then D_1 can be solved by a polynomial time algorithm. [5 Marks]

5.C How would you define the set of NP-complete problems? [5 Marks]

5.D (i) Outline the steps for proving that a decision problem D is an NP-complete problem.

(ii)Show that this proof will satisfy the definition you have given in 5.C.
[10 Marks]