

A27217

No calculator permitted in this examination

UNIVERSITY OF BIRMINGHAM

School of Computer Science

First Year – Degree of BSc with Honours
Artificial Intelligence and Computer Science
Computer Science
Computer Science with Business Studies

First Year – Degree of BEng/MEng with Honours
Computer Science/Software Engineering

First Year – Joint Degree of BSc/MSci with Honours
Mathematics and Computer Science
Pure Mathematics and Computer Science

First Year – Joint Degree of BEng/MEng with Honours
Electronic and Software Engineering

First Year – Combined Degree of BSc with Honours
Natural Sciences with Computer Science

First Year – Combined Degree of BA with Honours
Computer Science/French Studies

06 22754

Foundations of Computer Science

Summer Examinations 2011

Time allowed: 3 hours

[Answer ALL Questions]

[Use a Separate Answer Book for EACH Section]

[Use a Separate Answer Book for THIS Section]]

SECTION A

1. Question on logarithms in base 2 and binary notations.

- (a) Calculate the floor and the ceiling of logarithms of the following numbers. Show work. [5%]
- (i) 1000
 - (ii) 1024
 - (iii) 10^6
 - (iv) 2^n
 - (v) $1+2^n$
- (b) If you want to represent unsigned integers in the range 0 to 2047 in binary, what is the minimum numbers of bits needed? Justify your answer. [2%]
- (c) If you want to represent integers in the range -64 to 63 in two's complement binary notation, how many bits are needed? Justify your answer. [2%]
- (d) If you use 8 bits to represent integers in binary with two's complement, what is the range of representable integers? [2%]
- (e) Using 6 bits and two's complement, convert the numbers -17 and 23 to binary, add these numbers directly in binary, and then convert the result back to decimal. Show work. [2%]
- (f) Give a binary approximation of the number $1/10$ with 10 binary digits after the fractional point, chopping off the remaining digits. Show work. [2%]

2. Question on algorithm specification, design, correctness, and run-time analysis.

The logarithm of a positive integer can be calculated with the following algorithm: we repeatedly divide the given number by two, using integer arithmetic, until the number becomes 1. We count how many divisions were performed, and this count gives an integer approximation of the logarithm.

- (a) Does this count give the floor or the ceiling of the logarithm? Justify your answer. [3%]
- (b) Write this algorithm as a Java method. [4%]
- (c) Supply pre-conditions and post-conditions to specify the algorithm. Also supply a loop invariant and prove its correctness. [4%]
- (d) How many loop iterations are needed to compute the logarithm of x using this algorithm? Justify your answer. [4%]

No calculator

3. Develop an algorithm, written as a Java method, with the following specification:

- (a) **INPUTS:** an integer array a , and an integer x .
RETURNS: an integer s .
PRE-CONDITION: the array a is sorted.
POST-CONDITION:
 $0 \leq s \leq a.length$,
 $a[i] < x$ for all valid indices $i < s$, and
 $a[i] \geq x$ for all valid indices $i \geq s$.

Remark. If the number x occurs in the array a , then s is the smallest index with $a[s]=x$.

[10%]

- (b) **RUN-TIME:** it is required that no more than $\log(a.length)$ loop iterations are performed, where the logarithm is in base 2. A justification that this requirement is met should be included, where termination should also be argued. [5%]
- (c) **LOOP-INVARIANT:** should be provided, with justification, to establish the correctness of your algorithm. [5%]

[Use a Separate Answer Book for THIS Section]

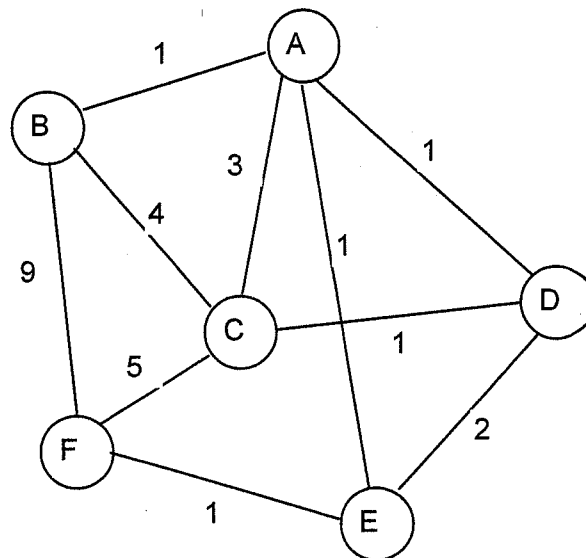
SECTION B

4. (a) List and describe the standard primitive constructors, selectors and condition for a binary tree. [3%]
- (b) Using those primitives write a procedure `isLeaf(bt)` that returns `true` if the binary tree `bt` is a leaf node, and `false` if it is not. [3%]
- (c) Then write a procedure `numNonLeaf(bt)` that returns the number of nodes in the binary tree `bt` that are not leaves. [4%]
- (d) Draw the binary search tree that results from inserting the items [25, 8, 12, 6, 31, 27, 7, 10, 28, 33, 11] in that order into an initially empty tree. [3%]
- (e) Draw the binary search tree that results from deleting the items 8 and 27 in that order from that tree. [3%]
5. (a) Sorting the set of numbers [4, 3, 4, 8, 1, 7] to illustrate your answer, explain how *selection sort* and *insertion sort* work. [6%]
- (b) In terms of the number of items n to be sorted, what are the time complexities of these sort algorithms? Is it possible for a sorting algorithm to do better than that? If so, state what is the best possible time complexity, and name an algorithm that achieves it. [2%]
- (c) What does it mean to say a sorting algorithm is *stable*? Again using the set of numbers [4, 3, 4, 8, 1, 7] to illustrate your answer, comment on the stability of selection sort and insertion sort. [4%]

No calculator

6. (a) Explain what is meant by the terms *Hash Table*, *Hash Function* and *Hash Collision*. [3%]
- (b) Comment on how the costs associated with good hash tables vary with the number of entries, and why that makes them useful. [3%]
- (c) Suppose strings of three digits are to be stored in a Hash Table represented as an array of size 11. The primary hash function is the sum of the first and second digits modulo 11. The secondary hash function is the sum of the second and third digits modulo 7 plus 1. Insert the following strings into an initially empty Hash Table: "228", "127", "737", "555", "371" and "234". [4%]

7. (a) Explain the steps involved in Kruskal's edge-based algorithm for determining a minimal spanning tree of a weighted graph. Why is it said to be a greedy algorithm? [4%]
- (b) Use Kruskal's algorithm to determine a minimal spanning tree of the following weighted graph. Show the graph (without the weights) after each step. [4%]



- (c) What is the time complexity of Kruskal's algorithm? Comment on the speed of Kruskal's algorithm compared with Prim's vertex-based algorithm for the same problem. [4%]