A—

Calculators may be used as specified by the School as follows:

THE UNIVERSITY OF BIRMINGHAM

THIS PAGE TO BE REPLACED BY OFFICE

– —

Introduction to Computer Science

Sample Exam 2014 1.5 hours

[Answer all of part A and 3 out of 4 questions from part B.]

PART A: Answer all of Question 1.

1. (a) What are differences between compilers and interpreters? **[5%]**

   (b) Give one advantage and one disadvantage of using an array, rather than a linked list. **[5%]**

   (c) Two ways of measuring the efficiency of an algorithm are: (i) measuring the time required to run it on a set of benchmark examples; (ii) analysing its time complexity. What are the key differences between these two approaches? **[8%]**

   (d) Explain the difference between the notions of *partial correctness* and *total correctness* for an algorithm. **[7%]**

   (e) Express the following numbers in two's complement fixed point representation using 4 bits to represent the integer component and 4 bits to represent the non-integer component **[10%]**

   i. 3.5
   ii. -4.25
   iii. 7.375
   iv. -6.514

   Comment on your result for part iv.

   (f) What is the minimum possible depth for a binary tree containing 6 items? **[5%]**

PART B: Answer 3 of the 4 questions in this section.

2. (a) Explain each of the following Java bytecode commands in a few words.

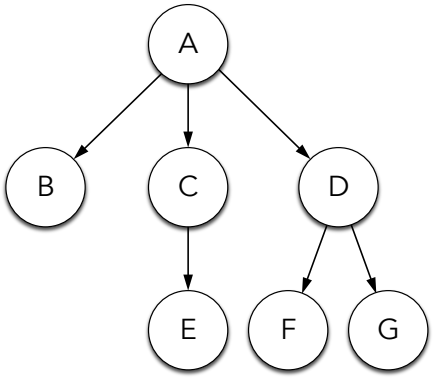
```
 I public int xyz(int, int, int);
0: iload_1
1: iload_2
2: imul
3: iload_3
4: iconst_5
5: iadd
6: istore 4
7: iload 4
8: ireturn
```

[10%]

(b) What is the return value of the method call `xyz(2,3,4)`? **[10%]**

3. (a) Draw a balanced binary search tree containing six items with values: 1,2,3,4,5,6 **[5%]**

   (b) For the following search tree state the order in which the nodes will be expanded assuming that the left branch of a node is expanded before the right branch.
   i. depth-first search **[5%]**
   ii. breadth-first search **[5%]**



   (c) State an advantage of depth-first search over breadth-first search, and one disadvantage. **[5%]**

4. (a) Below is some pseudo-code for an algorithm to compute the maximum distance between `n` points, stored in an array, called `points`. For any point `p`, we can access the x and y coordinates as `p.x` and `p.y`, respectively.

```
1.  maxDist = 0
2.  for i=0...n-1:
3.      for j=0...n-1:
4.          p1 = points[i]
5.          p2 = points[j]
6.          dist = sqrt((p1.x-p2.x)^2 + (p1.y-p2.y)^2)
7.          if dist > maxDist:
8.              maxDist = dist
9.  return maxDist
```

Give both the time complexity and the complexity class for this algorithm and explain your reasoning. **[10%]**

(b) Assume that some additional code is appended to the end of this algorithm, which loops through the points and finds the average of the x-coordinate of the points (by summing them and diving by `n`). Explain how this affects the overall time complexity and complexity class of the algorithm. **[6%]**

(c) Explain how you would derive the complexity class for the extended algorithm in part (b), considering only the complexity class of the original algorithm and its extension, not the precise number of operations required. **[4%]**

5. Translate the following pieces of assembly code into high-level language. Assume that registers `$r1`, `$r2` and `$r3` are used to store variables `x`, `y` and `i` respectively.

(a)
```
     addi  $r3,$zero,1
L1:  sub   $r4,$r3,$r1
     bgez  $r4,L2
     add   $r1,$r1,$r3
     addi  $r3,$r3,1
     j L1
L2:  add  $r2,$r2,$r1
```

[10%]

(b)
```
     sub  $r4,$r3,$ri
     blez $r4,L1
     addi $r2,$r2,1
     j L3
L1:  bgez $r4,L2
     addi $r2,$r2,-1
     j L3
L2:  addi $r2,$r2,0
L3:  mult $r2,$r2,$r1
```

[10%]

You may wish to refer to the following table of instructions:

branch-on-greater-than-or-equal-to-zero	`bgez $rx,L`	`if rx>=0 go to L`
branch-on-less-than-or-equal-to-zero	`blez $rx,L`	`if rx<=0 go to L`
addition	`add $rz,$rx,$ry`	`rz=rx+ry`
immediate addition	`addi $rz,$rx,c`	`rz=rx+c`
subtraction	`sub $rz,$rx,$ry`	`rz=rx-ry`
multiplication	`mult $rz,$rx,$ry`	`rz=rx*ry`
jump	`j L`	`Go to L`