Durham University

EXAMINATION PAPER

Examination Session: May/June	Year: 2016	**Exam Code:** COMP1051-WE01

Title:

COMPUTATIONAL THINKING

Time Allowed:	2 hours		
Additional Material provided:	None		
Materials Permitted:	None		
Calculators Permitted:	No	Models Permitted:	
Visiting Students may use dictionaries: Yes			

Instructions to Candidates:	Answer THREE questions. (TWO from Section A and ONE from Section B) PLEASE ANSWER EACH SECTION IN A SEPARATE ANSWER BOOKLET

Revision:	

Section A Computational Thinking (Prof. Iain A. Stewart)

Question 1

(a) State Moore's Law. Irrespective of Moore's Law, what is a problem as regards modern CPUs? **[2 Marks]**

(b) Define what a Boolean function is. In relation to Boolean functions, what is so special about NOT-, AND- and OR-gates? **[3 Marks]**

(c) Show how a NOT-gate, an AND-gate and an OR-gate can be constructed using just NAND-gates. **[8 Marks]**

(d) What are the three fundamental phases of integrated circuit design? You should briefly explain what each phase does. **[6 Marks]**

(e) Name three key components in a CPU microarchitecture and briefly describe the purpose of each. **[6 Marks]**

continued

Question 2

(a) The Boyer-Moore algorithm looks for an occurrence of a pattern string P within a text string T. Without using pseudo-code, explain how the two heuristics used within the algorithm work. (You should first explain how the naive string matching algorithm works and then describe your heuristics within this context.) **[10 Marks]**

(b) Why does the Boyer-Moore algorithm perform better than the naive string matching algorithm? **[3 Marks]**

(c) Suppose that we have an alphabet of $\{a, b, c\}$ and a pattern string $P = caba$. Calculate the values proposed by the two heuristics used in the Boyer-Moore algorithm. **[12 Marks]**

continued

Question 3

(a) What is an abstraction in relation to real-world problem solving? Consider the following two real-world problems.

   i. Suppose that you have a set of jobs that need to be allocated to (equal length) time-slots. Jobs can be scheduled in any time slot but some pairs of jobs are in conflict and so they cannot be scheduled in the same time slot. The problem is to schedule the jobs so that as few time slots are used as possible. We wish to find this optimal number.

   ii. The radio frequency assignment problem is the problem of assigning a radio frequency to every transmission mast so that any two masts that are less than 1 km apart are assigned different frequencies and so that as few frequencies are used as possible. We wish to find this optimal number.

Explain how you would abstract these two real-world problems for computational solution as *the same* optimization problem. **[8 Marks]**

(b) Describe a greedy algorithm to solve your optimization problem. (You may use natural language to describe your algorithm and need not supply pseudo-code.) Is your algorithm optimal? (If so then you should prove this; otherwise, you should give an instance where your algorithm does not give the optimal solution.) **[7 Marks]**

(c) The Satisfiability Problem was the first decision problem that was shown to be **NP**-complete. Define the Satisfiability Problem (be sure to say how we define the size of an instance). **[4 Marks]**

(d) Suppose that you had a fast algorithm to solve the Satisfiability Problem. What would this mean for the job scheduling and frequency assignment problems above? **[6 Marks]**

Section B Computational Thinking (Prof. Iain A. Stewart)

Question 4

(a) Define precisely what we mean when we say that two functions $f : \mathbb{N} \to \mathbb{N}$ and $g : \mathbb{N} \to \mathbb{N}$ are such that $f = O(g)$. Give two reasons why we use the Big-O notation when describing the time complexity of algorithms? **[5 Marks]**

(b) What is a non-deterministic algorithm? Define how a non-deterministic algorithm solves a decision problem. How do we measure the time taken by a non-deterministic algorithm? **[6 Marks]**

(c) What are the complexity class **P** and the complexity class **NP**? **[2 Marks]**

(d) Consider the following.

i. The Independent Set (decision) problem has as its instances pairs (G, k) where G is a graph and k is a natural number so that an instance is a yes-instance if there is an independent set of size k.

ii. The Clique (decision) problem has as its instances pairs (G, k) where G is a graph and k is a natural number so that an instance is a yes-instance if there is a clique of size k.

iii. The algorithm α takes as input a graph G and outputs the graph H on the same vertex set but where there is an edge (u, v) in G if, and only if, there is no edge (u, v) in H. (Note that G has a clique of size k if, and only if, H has an independent set of size k.)

How are the two statements "the Clique problem is in **P**" and "the Independent Set problem is in **P**" related? (You should explain your reasoning in detail.) **[9 Marks]**

(e) What is the Church-Turing Thesis? Can it be proven? What do we mean when we say that a decision problem is solvable? **[3 Marks]**

continued

Question 5

(a) Give two different ways in which a software bug can arise and give two different effects of a software bug. **[2 Marks]**

(b) Give the different aspects of the software engineering software design process known as the Waterfall Model and very briefly explain what they are. **[5 Marks]**

(c) When it comes to establishing program correctness, briefly compare and contrast the different approaches taken in software testing and in formal methods. **[4 Marks]**

(d) What is the difference between an algorithm being totally correct and being partially correct? **[2 Marks]**

(e) Consider the following algorithm where the input is a non-negative integer supplied via the variable y:

```
algorithm:  f(y)
IF y == 0:
  return 1
ELSE:
  x = f(y - 1)
return y * x
```

What does this algorithm do? By using induction, prove that it is totally correct. **[7 Marks]**

(f) What is the lexicographical order on pairs of natural numbers? How might you undertake an induction on a property indexed by pairs of natural numbers? **[5 Marks]**

END OF PAPER