Durham University

EXAMINATION PAPER

Examination Session:	Year:	Exam Code:
May	2015	COMP1051WE01

Title: Computational Thinking

Time Allowed:	2 hours	
Additional Material provided:		
Materials Permitted:		
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) ANSWER EACH SECTION IN A SEPARATE ANSWER BOOK

Revision:

Section A Computational Thinking (Dr Stefan Dantchev)

Question 1

(a) Explain **Selection-sort** first in plain English and then in pseudo-code. **[8 Marks]**

(b) Describe how your **Selection-sort** algorithms works on the following list of numbers: $4, 5, 3, 2, 5, 2$. **[5 Marks]**

(c) Describe **Binary-search** in plain English. **[5 Marks]**

(d) Describe how **Binary-search** tries to find 4 in the list $0, 1, 3, 5, 6, 8, 9$. **[3 Marks]**

(e) What are the worst-case running times of **Selection-sort** and **Binary-search**, respectively if the input list is of length n? Give a brief justification of your answers. **[4 Marks]**

Question 2

(a) Recall the definition of a regular expression over some alphabet Σ:

- any symbol $a \in \Sigma$ is a regular expression;
- $\emptyset$ and ε are special regular expressions; and
- if α and β are regular expressions then so are:
  - $(\alpha\beta)$
  - $(\alpha \mid \beta)$
  - (α^*).

Explain how a regular expression defines a set of words over Σ. **[7 Marks]**

(b) Give a regular expression over the alphabet $\{a, b\}$ that defines precisely the words that contain at least one a and at least one b. **[4 Marks]**

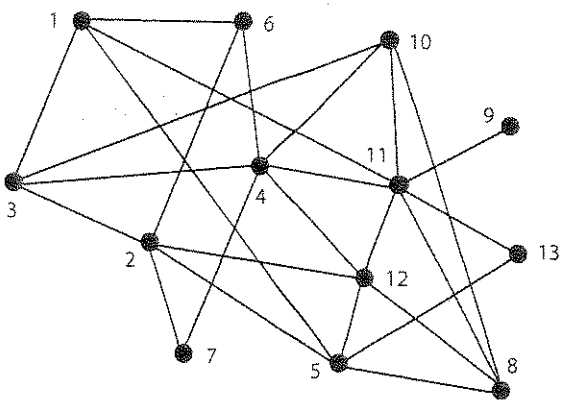
(c) Explain what a finite-state machine is and how such a machine accepts or rejects a word. **[9 Marks]**

(d) Give a finite-state machine that accepts precisely the words over the alphabet $\{a, b\}$ that start and end with an a. **[5 Marks]**

Question 3

(a) Explain **a greedy algorithm for graph colouring** in plain English. Does your algorithm always find an optimal colouring, i.e. one with the minimum number of colours? (No detailed explanation needed.) **[5 Marks]**

(b) Describe how your **greedy algorithm** works on the following graph. **[7 Marks]**



(c) Sometimes we can decide whether a graph can be coloured with a certain number of colours, k say, by transforming the graph into a smaller graph so that the original graph can be k-coloured if, and only if, the transformed graph can. We can iteratively do this so that we get a graph so small that we can trivially see whether or not it can be k-coloured. Give an example of such a transformation that we might use in this way. Apply your transformation to correctly decide whether the graph above can be coloured using 4 colours. **[5 Marks]**

(d) What is an independent set in a graph? Find an independent set of size at least 6 in the graph above. **[4 Marks]**

(e) What is the running time of your **greedy algorithm** in item (a)? **[4 Marks]**

End of Section A

Section B Computational Thinking (Dr Stefan Dantchev)

Question 4

(a) Explain what a **decision problem** is. What does it mean to solve a decision problem? **[4 Marks]**

(b) Explain what a **search problem** is. What is meant by a solution of a search problem? **[6 Marks]**

(c) Explain how there is always a decision problem that corresponds to a search problem. Give an example of a search problem whose respective decision problem is trivial. **[4 Marks]**

(d) Explain what an **optimisation problem** is. **[5 Marks]**

(e) Abstract a computational problem out of the real-world problem below.

A cinema has a number of films for the day. Each film is no longer than two hours, so the manager decides to divide the evening into two-hour slots. There are enough screens, so that all films could be shown in parallel, which would be ideal for the manager as he wants to go home early. However, each viewer has purchased tickets for a number of films that they want to see. With the knowledge of the tickets, purchased by every viewer, how can the manager schedule the screenings so that every film is screened just once, every viewer can see all the films that they have tickets for, and the cinema closes as early as possible. **[6 Marks]**

Question 5

(a) What are the basic general principles behind measuring the time taken by some algorithm, expressed using pseudo-code, on some particular input? **[5 Marks]**

(b) Recall the definition of **big-O** notation:

For two functions $f, g : \mathbb{N} \to \mathbb{N}$, we write $f = O(g)$ if there exists some $n_0 \in \mathbb{N}$ and some positive rational $k \in \mathbb{Q}$ such that $f(n) \leq kg(n)$ whenever $n \geq n_0$.

In the context of comparing running times of two algorithms, explain what the functions f and g are and what the constants k and n_0 are used for. **[6 Marks]**

(c) We are given five different algorithms with quite precise running times, estimated as follows

$$\Theta(n), \Theta(\log^2 n), \Theta(1.1^n), \Theta(n^2), \Theta(n \log n),$$

respectively. Order these algorithms by their efficiency. **[5 Marks]**

(d) Explain what the complexity classes $\mathbf{P}$ and $\mathbf{NP}$ are, and how they are related. What is an $\mathbf{NP}$-complete problem? Is it likely that such a problem is in $\mathbf{P}$ and why? **[6 Marks]**

(e) Give examples of three graph problems, one in $\mathbf{P}$, another $\mathbf{NP}$-complete, and a third one which is unlikely to be either in $\mathbf{P}$ or to be $\mathbf{NP}$-complete. **[3 Marks]**

END OF PAPER