Matric No: _____

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

ALGORITHMS AND DATA STRUCTURES

SET09117

Academic session:	**2015/6**	Diet:	**First**
Exam duration:	**1:30**	Reading time:	**0**
(excluding reading time)		Total exam time:	**1:30**

Please read full instructions before commencing writing

Exam paper information

- Total number of pages:
- Number of questions: 5
- Attempt three question.

Special instructions

- None

Special items

- Calculator

Examiner(s): Neil Urquhart, Simon Wells

1) Explain, with the aid of diagrams, each step within taken by Dijkstra's shunting algorithm to convert following expressions from infix notation to postfix notation. You may assume a precedence order of * / + -

i) 4 + (2 * 3)

(4)

ii) (1 + 2) * 3

(4)

b) Evaluate the following expressions, which are presented in reverse Polish notation:

i) 6 7 3 * +

(3)

ii) 6 2 / 2 +

(3)

iii) 1 2 - 3 4 * + 5 +

(3)

c) Explain why it may be advantageous for a computer to convert expressions from infix notation to Reverse Polish Notation for evaluation.

(2)

d) Using 'Big O' notation explain complexity of the following sections of code. Explain and justify your answers.

Assume that :

```
statement 1;
statement 2;
```

has a complexity of O(1).

i)

```
For (i =0; i < n; i++){
      for (j =0; j < n; j++){
            statement 1;
            statement 2;
      }
}
```

ii)

```
for (i = 0; i < X; i++) {
    statement 1;
}
for (j = 0; j < Y; j++) {
    statement 2;
}
```

(6)

2

a) Describe what is meant by a “divide and conquer” algorithm and give an example of such an algorithm.

(5)

b) Under what circumstance might you choose to use a linear search algorithm instead of another search algorithm (such as binary search)? Explain your reasoning.

(5)

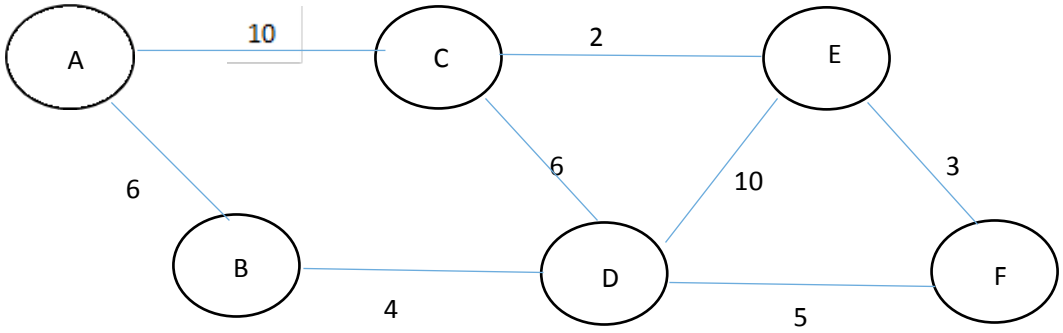
c) Apply the selection sort algorithm to the following array of numbers:

[68, 21, 15, 10, 88, 24,12]

For each iteration of the algorithm write out the array, underlining the numbers that have changed at each stage ,

(15)

3 Answer the following questions with reference to this graph:



a) Kruskal's algorithm may be used to calculate the minimum spanning tree of a graph. Show how Kruskal's algorithm would construct a minimum spanning tree for the above graph commencing from node A. Describe the algorithm, and describe each step taken.

(10)

b) Show how Dijkstra's algorithm can be used to calculate path costs from Node A to every other node. Describe the algorithm, and describe each step taken.

(10)

c) Explains the properties of a Hamiltonian Circuit and provide an example based on the above graph.

(5)

4

a) You are tasked with storing and perfoming lookups on an array of integers to which new elements will be added periodically. Compare and contrast the following strategies for storing and performing lookups on the array and give advantages and disadvantages of each:

(i) Searching the naturally ordered array.

(ii) Sorting the array and using a binary seach.

(iii) Inserting the elements into a hash table using an appropriate hash function.

(9)

b) Given a hash table of size 7 and the following hash function:

h(x) = x modulo 7

Use linear probing to insert the following elements in order,

1,15,14,3,9,5,26

For each element identify:

(i) Each slot that that the element tries to occupy.

(ii) Whether a collision occurs.

(iii) Give the final order of the array.

(16)

5

a) Consider the following grammar:

(R1)	S	→	NVN
(R2)	N	→	girl
(R3)	N	→	cat
(R4)	N	→	yarn
(R4)	V	→	scratches
(R5)	V	→	nuzzles

How many sentences can be generated from this grammar?

(4)

b) From the grammar, identify 2 sentences that make sense and 2 sentences that do not make sense. With reference to the concepts of syntax and semantics, explain why some sentences make sense and others don't.

(7)

c) Add a recursive rule to the grammar from a) using an adjective, such as scary, so that the language generated by the grammar is now infinite. Give 2 examples of sentences that can be generated from the new grammar that both use your new rule.

(6)

d) What is a Lexer? In your answer you should discuss the inputs and outputs of a lexer as well as the wider role that Lexers play within a language-processing pipeline, for example, in relation to a Parser.

(8)