NATIONAL UNIVERSITY OF SINGAPORE

UIT2201 : COMPUTER SCIENCE AND THE IT REVOLUTION

(Semester I : AY 2014-15)

November 2014 – Time Allowed: 2 Hours

INSTRUCTIONS TO CANDIDATES

1. This examination paper consists of **FIVE** questions and comprises **ELEVEN** printed pages, including this page.
2. Answer **ALL** questions.
3. **Write ALL your answers in this examination book.**
4. This is an **OPEN BOOK** examination.

Matric. Number: ____________________

QUESTION	POSSIBLE	SCORE
Q1	15	
Q2	15	
Q3	15	
Q4	15	
Q5	20	
TOTAL	**80**	

Question 1: (15 marks)

True-False questions. (2 marks each)

(a) The "*Guess the number between 1 to 100*" game done in the lectures uses the *same idea* as the *binary search algorithm*, and the answer can be correctly "guessed" within 7 guesses. ________

(b) An algorithm **S** with running time $0.2014n^2$ is faster than an algorithm **T** with running time of $2014n$ for large values of n. ________

(c) We can build a logic circuit for *any* logic function on inputs A, B, C using ***only*** NAND gates and AND gates. ________

(d) Computing the *average* of n numbers in an array $\boldsymbol{D} = (D_1, D_2, \ldots, D_n)$ can be done in *logarithmic time* $\Theta(\lg n)$, i.e. directly proportional to $(\lg n)$ using a repeated-divide-by-2 algorithm. ________

(e) The MDR is one of the two registers used to interface between the actual memory and the rest of the CPU design. The MDR stores the *address* of the memory cell to be accessed during a READ operation. ________

(f) When we re-use a design or a function as a *black box*, and use it in more complex designs or functions, we call that "*reusing the result*". ________

Short answer. (3 marks)

(g) Explain the meaning of the *closed-world assumption* in the context of *knowledge-based systems*. Give *one important consequence* of the closed-world assumption.

Fun Question: (1 bonus mark)

Prof. Leong uses the term ________________________ to describe the Magic tricks he performs, which are based on one of the branches of STeM (*Science*, *Technology-*and-*CS*, *Engineering*, and *Mathematics*).

Question 2: (15 marks)

(a) Consider a database with the following 3 tables: **`{SI, CI, EN}`**. We assume that |**`SI`**|=30,000, |**`CI`**|=1000, |**`EN`**|=100,000. (To save space and writing, you should use the short table names.)

SI (STUDENT-INFO)						
Student-ID	**Name**	**NRIC-No**	**Address**	**Tel-No**	**Faculty**	**Major**
---	---	---	---	---	---	---

CI (COURSE-INFO)					
Course-ID	**Name**	**Day**	**Hour**	**Venue**	**Instructor**
---	---	---	---	---	---

EN (ENROLMENT)	
Student-ID	**Course-ID**
---	---

We want to obtain the list of students who are from the Faculty "**`FASS`**" and taking the course with Course-ID "**`CS3230`**". For these students, we want to list their **`Student-ID, Name, Faculty, Course-ID`**.

(i) (4 marks) Give an appropriate **SQL query** to accomplish this task.

(ii) (4 marks) Give an appropriate sequence of ***basic database primitives*** operations (using **`e-project, e-select, e-join`**) to accomplish this task. Make it ***as efficient as you can***.

Question 2: (continued)

(b) At the end of the semester, Prof. Leong of UTP (*University Talent Programme*) gave a bonus score (an integer from 1 to 5) to each of the *n* students in his class. These bonus scores are stored in the array (list) ***BS***[1..*n*].
Next, he wants to calculate the *histogram* of the bonus scores ***H***[1..5], where ***H***[*k*] is the number of times the score *k* appears in ***BS***[1..*n*]. For example, ***H***[4] is the number of times the score of 4 appears in ***BS***[1..*n*], or equivalently, it is the number of students who got a bonus score of 4.

(i) (1 mark) Consider this example of the bonus scores ***BS***[1..8] for 8 students. Calculate the histogram ***H***[1..5] for this example.

BS[1..8] = [1, 3, 4, 5, 4, 3, 5, 4]

Answer: ***H***[1..5] = [ ]

(ii) (4 marks) Give an *efficient algorithm* for computing the histogram ***H***[1..5] given the bonus scores ***BS***[1..*n*] as input. Give your algorithm in pseudo-code.

(iii) (2 marks) What is the time complexity of your algorithm, given in Θ-notation?

Question 3: (15 marks)

You are the new intern in the UTP (*University Talent Programme*). In UTP, the *faculty members* (denoted by *A, B, C, D, E, F, G*) belong to various *committees* (denoted by AA, BW, CC, DD, ER, FL) as given below:

A :	{ BW, CC, DD }	*B* :	{ CC, DD }
C :	{ BW, CC, FL }	*D* :	{ CC, ER }
E :	{ DD, FL }	*F* :	{ AA, ER }
G :	{ AA, CC, ER }		

As the intern, you are asked to schedule a 1-hour meeting *for each committee* for next Friday. And you are also told to schedule the meetings so as *to finish all the meetings in the fewest number* of hourly time-slots possible (so that the members can have time do other things on Friday).

Note: Because each faculty member belongs to several committees, some of the committee meetings cannot be held concurrently (since the faculty member cannot be in two meetings at the *same* time).

(a) (4 marks) Show how you can model this scheduling problem as a *graph colouring problem*. Clearly define the graph you are using for this purpose.

(b) (4 marks) Draw the graph for the example of the problem instance given above.

(c) (2 marks) Give a meeting schedule that uses the *minimum* number of time-slots.

Time-Slot-1: ________________ Time-Slot-4: ________________

Time-Slot-2: ________________ Time-Slot-5: ________________

Time-Slot-3: ________________ Time-Slot-6: ________________

Question 3: (continued)

(d) **(5 marks)** The use of the *graph colouring problem* to model and solve the *Committee Meeting Scheduling Problem* described in this question is an example of problem transformation.

Give a brief description of this problem transformation process. Give one advantage of this method of problem solving.

Question 4: (15 marks)

Consider the following logical formula:

F = A*~B + ~A*C + B*C

where we use "**`*`**" to denote logical **`AND`**,
"**`+`**" to denote logical **`OR`**, and
"**`~`**" to denote the logical **`NOT`**.

(a) (3 marks) Give the truth table for **F**. Complete the table below.

A	B	C	F			
0	0	0				
0	0	1				

(b) (1 marks) From the truth table, give a logical formula for **F**.

ANSWER: F =

(c) (4 marks) Simplify the formula for **F** *as much as you can*.
(*Hint*: Simplest form uses less than 3 AND/OR gates.)

ANSWER: F =

For working:

(d) (2 marks) Give a logic circuit for **F** using your answer for (c) above.
(Use only NOT gates and 2-input AND/OR gates in your circuit.)

Question 4: (continued…)

(e) **(2 marks)** In the lectures, it was stated that the set **{*, +, ~}** is *logically complete* where *, +, and ~ represents the AND, OR, and NOT gates, respectively. Explain clearly what is meant by *logically complete*.

(f) **(3 marks)** Explain why the set **{+, ~}** is *also* logically complete.

Question 5: (20 marks)

(a) You are given a knowledge-based system with the following knowledge base,

```
male(eli)            male(bill)
male(joe)
female(mary)         female(betty)
female(sarah)        female(annie)

parent-of(eli, bill)     parent-of(mary, bill)
parent-of(bill, joe)     parent-of(bill, betty)
parent-of(mary, sarah)   parent-of(sarah, annie)
```

where

`male(eli)`	*asserts that* "**`eli`** is male",
`female(mary)`	*asserts that* "**`mary`** is female",
`parent-of(eli, bill)`	*asserts that* "**`eli`** is parent of **`bill`**"

(i) (3 marks) Draw a "relationship-diagram" based *only on* these facts.
Use different representations for **`male`**, **`female`**, and **`parent-of`**.

(ii) (3 marks) Give the inference rules to define the following:

`father-of(X, Y)`	*asserts that* "**`X`** is father of **`Y`**",
`ancestor-of(X, Y)`	*asserts that* "**`X`** is ancestor of **`Y`**".

R1: father-of(X, Y) if

R2: ancestor-of(X, Y) if

R3: ancestor-of(X, Y) if

(iii) (6 marks) Answer the following queries:

?parent-of(**P**, sarah) **P =** ______________________

?father-of(**Q**, bill) **Q =** ______________________

?ancestor-of(**S**, annie) **S =** ______________________

Question 5: (continued…)

(b) In the pancake-flipping problem, you are given the following initial configuration of four pancakes (of sizes 1,2,3,4):

1 2 4 3

which means that the size-1 pancake is on top, then the size-2 pancake is next, followed by the size-4 pancake next, and finally, the size-3 pancake is at the bottom.

(i) (3 marks) Given this initial configuration, use the *Greedy algorithm* (given in the lectures) to sort these pancakes. Show the intermediate configurations during the sorting process. How many flips are needed?

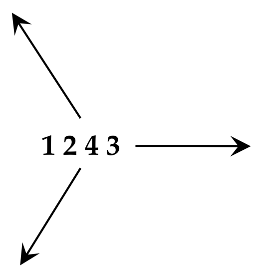
Answer:

1 2 4 3 →

Number of flips needed: ____________

(ii) (3 marks) Draw the portion of the *state space graph* consisting of all the states and edges that are *within 2 moves* from the starting state (configuration).

Your Answer:



(iii) (2 marks) Using this or otherwise, give an optimal sequence of flips for this instance of the pancake-flipping problem that uses the fewest number of flips.

Your Answer: (optimal sequence of flips)

1 2 4 3 →

Number of flips needed: ____________

(Extra page for your use)

~~~ END OF QUESTIONS ~~~