UIT2201: Computer Science and Information Technology Revolution (Fall Semester 2014)

MID TERM TEST (1 hour)

INSTRUCTIONS:

1. This Mid-Term Test is CLOSED BOOK / CLOSED NOTES.
2. Answer ALL questions in this answer book.
3. Make sure you write down your NAME and MATRIC NUMBER

Name: _____

Matric Number: _____

QUESTION	POSSIBLE	SCORE
Q1	15	
Q2	15	
Q3	15	
Q4	15	
TOTAL	60+1	

Fun Question: (1 bonus point)

Give the name of the mathematical principle used to decide *the suit* (namely, ♦♣♥♠, diamond, club, heart or spade) of the *hidden card* in our Magic Card Trick.

Answer: _____

(Now, please *relax* and *enjoy* the Quiz, OK?)

Question 1: (15 points)

True-False (2 point each)

(z) (Sample) *The course UIT2201 is taught by Prof. Leong Hon Wai* ***_TRUE__***

(a) Computer Science is the *study and use* of computers and computer programs and applications (including phone apps). _____

(b) The graph colouring problem can be used to model *many different* problems, *not only* the Tourist Problem introduced in Lecture 1. _____

(c) Of the 3 primitive database operations {e-project, e-select, e-join} the most expensive one is e-join. _____

(d) An algorithm **F** with running time of $2358n$ will be *faster* than an algorithm **G** with running time of $0.2358n^2$ when n is large. _____

(e) In Scratch, a sprite F can ask another sprite G to start executing, and continue executing, without waiting for G to finish. _____

(f) Each *column* of a database table is called a *record*. _____

Short Answer: (3 points)

(g) Explain why you think it is important to have ITEM (IT-Enabled Mindset) when considering the solutions to everyday problems in real life.

Answer:

Question 2: (15 points)

You are in charged of packing *volatile* items {A, B, C, D, E, F, G, H, J, K} into boxes for shipping. Because of the volatile nature of the items, some items conflict, namely, they cannot be packed together with some other items, or else bad things happen. Below, you are given, for each item, the list of items that they conflict with (namely, cannot be packed together with). For example, (from line 2) item ***B*** cannot be packed with A, C, or D.

A : B, C, D	***E*** : D, F, G, H	***J*** : F, K
B : A, C, D	***F*** : D, E, J, K	***K*** : F, J
C : A, B, D	***G*** : E, H	
D : A, B, C, E, F	***H*** : E, G	

We want to pack the volatile items into boxes (without conflicting items in any box) and use a minimum number of boxes. We call this the *Volatile Item Packing Problem* (VIPP).

(i) [3] As a UIT2201 student, you immediately want to model this problem with a conflict graph $G = (V, E)$. In this conflict graph, each vertex represents a volatile item. State clearly *under what condition* we connect any two vertices u and v? (Namely, how the edges in your conflict graph are defined.)

Answer: We connect two vertices u and v when _____

_____

(b) [5] Draw the conflict graph for the above example of the 10 volatile items.

(c) [3] Explain how this conflict graph model can be used to solve the VIPP.

(d) [4] Give an optimal packing for the above example. How many boxes are needed?

Question 3: (15 points)

Perform a binary search on the following sorted array A[1..6] of numbers.

A = [7, 13, 31, 43, 61, 89]

(a) [3] Draw the ***search tree*** that is used to visualize this **Binary-Search** algorithm applied to the sorted list **A** of size 6. (Include the "square" nodes for unsuccessful searches).

(b) [3] What is the ***number of comparisons*** needed for a *successful* search in (i) the worst-case and (ii) average-case? (Assume that all names are *equally likely* to be searched.) [**Note:** You can leave your answers in fractions.]

Worst Case: _____ comparisons

Average Case: (Assume all names equally likely)

(c) [3] What is the *number of comparisons* needed for an *unsuccessful* search in (i) the worst-case and (ii) average-case? (Assume that each gap "between the numbers" is *equally likely* to be searched.)

Worst Case: _____ comparisons

Average Case: (Assume all gaps equally likely)

Question 3: (continued…)

You are given the following algorithm called Mystery(A,n), where n is a positive integer and A[1..n] is an array. You can assume that array A[1..n] has been initialized with data.

```
Algorithm Mystery (A,n);
(* n is a positive integer; *)
(* Array A[1..n] has been initialized *)
begin
  k ← 1;
  while (k < n) do
    if (A[k+1] > A[k]) then
       swap (A[k], A[k+1]);
    endif
    k ← k+1;
  endwhile
  print A[n];
end;
```

(d) [3] Simulate algorithm Mystery(A,n) with input array **A** = [6, 8, 2, 4, 7], and *n*=5. Print out the final array A and the output from the print statement.

Answer: After running the algorithm, the update array **A** becomes:

A = [ ]

Output of "Print A[n]": _____

(e) [3] What is the worst-case running time of Algorithm Mystery(A, n), expressed in terms of *n*, using Θ-notation.

Worst Case: _____

Question 4: (15 points)

Given a database with the following 3 tables: **{SI, CI, EN}**. You should use these short table names to save space and writing. (Use the reverse blank pages, if necessary)

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
---	---

For **(a)** below, give the appropriate (i) **SQL query**, and (ii) a sequence of basic database primitives (using **e-project, e-select, e-join**):

(a) [6 pts] List the **Name, Address, Tel-No** of all students from the "Engg" faculty who are taking course with Course-ID "UIT2201".

(i) SQL Query:

(ii) Using Basic Primitives:

Question 4: (continued…)

The given database schema was designed some years ago when each course is taught by *exactly one instructor* who takes charge of everything. As the university expands, some popular large courses *need multiple lecture-sections*, each taught by a different instructor. Of course, *each student in the course takes only one lecture-section*.

In planning for this change, it was suggested that we just "add additional records for course, one each for the different instructors" in the Course-Info (CI) table. An example for CS1001 is shown below:

CI (COURSE-INFO)					
Course-ID	**Name**	**Day**	**Hour**	**Venue**	**Instructor**
CS1001	CS Intro	Wed	10:00	SR1	Fish Leong
CS1001	CS Intro	Tue	11:00	SR2	LeongHW
---	---	---	---	---	---

EN (ENROLMENT)	
Student-ID	**Course-ID**
A2223333	CS1001
A3334444	UIT2201
---	---

(b) [4] As a UIT2201 student, you immediately know that this "solution" will not work with the current database schema. Describe exactly why this "solution" will not work?

(c) [5] Suggest changes you can make either to the schema or to how to create the database in order to accommodate the new change – namely, having the ability to have multiple lecture-sections (each taught by a different instructor) while students can choose to attend only one of the lecture-sections.

~~~ END OF QUIZ ~~~