UIT2201: Computer Science and Information Technology Revolution (Spring Semester 2012)

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	20	
Q2	15	
Q3	15	
Q4	10	
TOTAL	60+1	

Fun Question: (1 bonus point)

Name the Nobel Laurette who is associated with the famous “Piano Tuner Problem?

____

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

Question 1: (20 points)

True-False (2 point each)

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

(a) An algorithm **X** with running time $2012n$ will be faster than another algorithm **Y** with running time $0.2012n^2$ for all values of n. ________

(b) In Scratch, we can define a *variable* that is capable of storing *many* data values, under *one variable name* (thus, reducing the need to "invent" many variable names). ________

(c) The binary search algorithm is *very fast* because it keeps reducing the problem size by a factor of two. ________

(d) The `Find-Max(A,n)` primitive that finds the ***maximum*** in an array `A[1..n]`, of n unsorted numbers has time complexity $\Theta(\lg n)$. ________

(e) *Query processing* (for example, answering SQL query) is no longer a complex research issue since we have very fast computers nowadays. ________

(f) Each *column* of a database table is called a *schema*. ________

Not True-False

(g) **(4 points each)** Rank the following time complexity in *increasing rate of growth* (from slowest to fastest).

$\Theta(n^2)$, $\Theta(n)$, $\Theta(n \lg n)$, $\Theta(2^n)$.

Answer: ____

Fill in the blanks: (2 points each)

(h) Give a common application of the **`Pattern-Match`** algorithm covered in class.

Answer: ____

(i) Name your favourite ***recurring principle*** covered in class.

Answer: ____

Question 2: (15 points)

You are given two lists (arrays) of characters, *T*[1..*n*] and *P*[1..*m*] (where *n* >> *m*). You can assume that *T* and *P* have already been filled with data (i.e. data has been read in).

You are given the following algorithm called **`U-Count(T, k, P, m)`** :

```
U-Count(T, k, P, m)
(* Compute some unknown quantity from   *)
(* array T[k..k+m-1] and array P[1..m]  *)
begin
1  s-count ← 0;
2  j ← 1;
3  repeat until (j > m);
4    if (P[j] = T[k+j-1]) then
5      s-count ← s-count + 1;
6    endif
7    j ← j + 1;
8  endrepeat
9  u-count ← m – s-count;
10 print u-count;
end;
```

(a) **(6 points)** Suppose that *T*[1..10] = [R E V O L U T I O N] and *P*[1..4] = [V O L T]. If we run algorithm `U-Count` with different inputs, what is the output (of line 10), and what is the final value of the variable `s-count` (at the end of the algorithm)?

	Output produced	**Value of `s-count`**
`U-Count(T, 1, P, 4)`	_______________	_______________
`U-Count(T, 3, P, 4)`	_______________	_______________
`U-Count(T, 8, P, 3)`	_______________	_______________

(b) **(3 points)** Describe in plain English, what `U-Count` actually counts?

(b) **(2 points)** What is a dominant operation for this algorithm `U-Count`?

Dominant Operation: ___________________________

(b) **(4 points)** Give the worst-case time complexity (expressed in the Θ-notation) of the algorithm `U-Count(T, k, P, m)` for general *T* and *P*?

Time Complexity: _______________

Question 3: (15 points)

You use the ***binary search algorithm*** `Bin-Search(SL, 9, aName)` given in class to search for `aName` (a given name) in a sorted list `SL` of `9` names, given below.

`SL = (Alvin, Beth, Cathy, David, Eve, Faith, Gail, Hon, John)`

(a) [5] Draw the ***search tree*** that is used to visualize this ***binary search algorithm*** applied to the sorted list `SL` of size `9`. (Include the "square" nodes for unsuccessful searches).

(b) [4] How many **name-comparisons** are needed to search (successfully or unsuccessfully) for each of the names given in the table below:

`aName`:	**Number of Name-Comparisons**
`Eve`	____________
`Cathy`	____________
`Bunny`	____________
`John`	____________

(c) [3] Assuming that each name is *equally likely* to be searched, what is the ***average number*** of name-comparisons used in a ***successful search***?

(d) [3] Estimate the average number of name-comparisons used in an ***unsuccessful search***. (You can assume that each "unsuccessful search node" is *equally likely*.)

Question 4: (10 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)** and **(b)** below, give the appropriate (i) **SQL query**, and (ii) a sequence of basic database primitives (using **`e-project, e-select, e-join`**):

(a) [4 pts] List the **Course-ID, Day, Hour** of all courses taught in the venue `"USP-SR1"`.

(i) SQL Query:

(ii) Using Basic Primitives:

(b) [6 pts] List the **Student-ID, Major** of `FASS` students who have classes in `"USP-SR1"`.

(i) SQL Query:

(ii) Using Basic Primitives:

~~~ END OF QUIZ ~~~