Computer Engineering Department
PhD Qualifier Exam

19 September, 2014

- Time: 10:00 - 13:00
- The exam contains 8 questions in total. You are to choose 6 of these questions and answer only to them.
- By submitting this exam, you agree to fully comply with Koç University Student Code of Conduct, and accept any punishment in case of failure to comply. If you do not submit this exam on time, you will directly fail the qualifier.
- The exam is open-book and open-notes.
- You are not allowed to use any electronic equipment such as computers and mobile phones.
- In general, you are not allowed to ask questions during the exam.
- Good luck!

1. a) Given an array A of n **integers**, such that $A_0 < A_1 < \ldots < A_{n-1}$, determine whether or not there exists an integer i such that $i = A_i$ (i.e., the i^{th} location in the array contains the integer i). If it exists, return the *first* such i that you found. If none exists, return -1. Your algorithm should run in $O(\log n)$ time.

   b) Give an $O(n \log k)$ algorithm to merge k sorted lists $L_1, \ldots, L_k$ (sorted in increasing order) into one sorted list (again in increasing order), where n is the total number of items in all the lists combined.

2. Random variable X has the PDF:

$$f_X(x) = \begin{cases} 2x/3 & \text{if } 1.0 < x \leq 2.0 \\ 0 & otherwise \end{cases}$$

Let A be the event $x \geq 1.5$. Determine the numerical values of the following quantities:

a) $E(x)$

b) $P(A)$

c) $E(x|A)$

The conditional PDF for random variable Y is

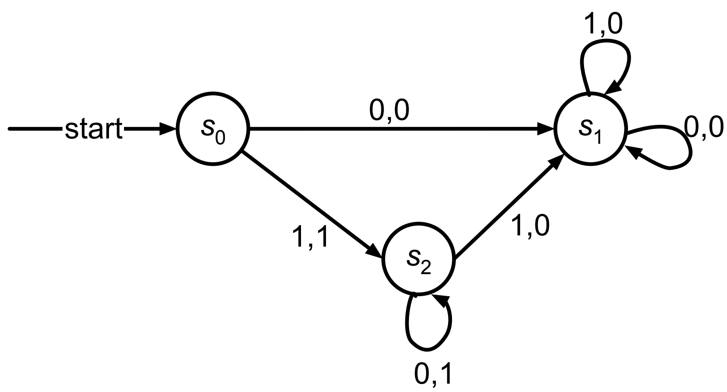
$$f_{Y|X}(y|x) = \begin{cases} 1/(2x) & \text{if } 0 \leq y \leq 2x \\ 0 & otherwise \end{cases}$$

d) Determine the numerical value of $P(x + y < 3)$.

e) Prepare a neat, fully labeled plot of the marginal pdf for random variable Y for all possible experimental values.

3. a) Find the language L generated by the phrase-structure grammar $G = (V, T, S, P)$, where $V = \{S, A, 0, 1\}$ (vocabulary with nonterminal S), $T = \{0, 1\}$ (terminals), and $P = \{S \to 1A, A \to 0A, A \to \lambda\}$ (productions).

b) Show that the finite-state machine (with output) given below *recognizes* (or *accepts*) the language $L = \{10^n | n \geq 0\}$.



c) Construct a finite-state automaton that recognizes the language $L = \{10^n | n \geq 0\}$.

4. **a) Scheduling**

   Consider **round-robin scheduling** algorithm with time quantum duration denoted by q. Assume that n processes each with equal CPU burst-time denoted by 4q exist.
   For such a system using round-robin scheduling, find the expression for **average waiting time** of n processes in terms of n and q. You can assume that context-switch time is zero. You must show your work to receive credit.
   (Note that, waiting time of a process is defined as the amount of time the process has been waiting in the ready queue.)

b) Virtual memory

Consider a **demand-paging system** with the following time-measured utilizations:

- CPU utilization 19%
- Paging disk 98%
- Other I/O devices 5%

Explain whether each of the following will **improve CPU utilization** or not.

- Install a faster CPU.
- Decrease the degree of multiprogramming.

5. Consider a weighted directed graph $G = (V, E)$ where edge weights may be negative. This question is about the *all-pairs shortest paths problem.* You are asked to devise a dynamic-programming algorithm that will compute, for *all* pairs of vertices $u, v \in V$, the length of the shortest simple path from u to v, shown by $\delta(u, v)$. If no such path exists, we say that $\delta(u, v) = \infty$.

   Suggested guidelines for solving the problem:

   - Let us define a subproblem of the all-pairs shortest paths problem as follows. Let $D(a, x, b)$ denote the length of the shortest simple path from a to b that goes through vertex x. What can you say regarding how $D(a, x, b)$ is related to $\delta(u, x)$ and $\delta(x, b)$?
   - Does this problem have the *optimal substructure property*, i.e., supposing that there is a shortest path from a to b that goes through vertex x. What can you say about the portions of this path that goes from a to x and from x to b?
   - For a fixed a and b, suppose you are given $\delta(a, x)$ and $\delta(x, b)$ for *all* vertices $x \in V$. Can you compute $\delta(a, b)$? How? How much does it cost asymptotically to do so?

   Using this or another problem decomposition, formulate a dynamic programming algorithm for the all-pairs shortest paths problem. Analyze the commplexity of your algorithm. Justify its correctness, i.e., that it does correctly compute the shortest path distance for every pair of vertices.

6. Using an object-oriented language you are comfortable with, write classes representing

   - `Atom`s, each with a name, an integer *atomic number* and *mass*, (Example: Oxygen would have name "Oxygen", atomic number 8 and mass 16
   - `Molecule`s, each with a name, and multiple `Atom`s, (Example: Water would have name "Water", and consist of two Hydrogen atoms and one Oxygen atom)

   Both `Atom` and `Molecule` objects should have a `getMass()` method that returns an integer. The mass of a `Molecule` is the sum of the masses of the `Element`s it consists of. For instance, `getMass()` should return $16 + 2 \times 1 = 18$.

   Write a class called `ChemUtil` that has a *static* method `sortByMass(listOfItems)` that takes in a *single argument* `listOfItems` (whose type you should decide) holding a sequence consisting of `Atom` and `Molecule` objects and sorts the sequence in non-decreasing order by mass.

   Feel free to use standard libraries you know about. If you can't remember the name of a class or method but are sure it exists in a library, give it a name and explain what it does. You will receive full credit.

7. a) Identify all variable declarations (circle, and label). Each missing declaration and non-declarations identified as declarations lose 1 point.

```
let x = 10

   in let f = proc(x) -(x,-(0,x))

      in let y = 1

         in let g = proc(x) begin set x=5; x; y end

            in -( (f begin set x=-(x,-(0,x)); y; x end), (g (g y)) )
```

b) Identify all variable references (circle, and label). Each missing reference and non-references identified as references lose 1 point.

```
let x = 10

   in let f = proc(x) -(x,-(0,x))

      in let y = 1

         in let g = proc(x) begin set x=5; x; y end

            in -( (f begin set x=-(x,-(0,x)); y; x end), (g (g y)) )
```

c) Write down the lexical depth of each variable reference. Missing and incorrect depths lose 1 point each.

```
let x = 10

   in let f = proc(x) -(x,-(0,x))

      in let y = 1

         in let g = proc(x) begin set x=5; x; y end

            in -( (f begin set x=-(x,-(0,x)); y; x end), (g (g y)) )
```

8. **Part 1.** The decomposition of a relation R is said to be lossless if, when you project R onto each of the schemas of the decomposition and then take the natural join of the results, you are guaranteed to get back no tuple other than those originally in R. (Note that projection onto any decomposition whatsoever, followed by a natural join of the projections will always produce at least all the tuples originally in R.) Consider the relation $R(A, B, C, D, E)$ with functional dependencies $A \to E$, $E \to D$, $CD \to A$, and $BC \to E$. Let the decomposition be onto relations with schemas ABC, BCD, and CDE. Prove that this decomposition is lossless.

Part 2. Let relations $R(A, B, C)$ and $S(C, D, E)$ have the following properties: R has $10,000$ tuples, S has $30,000$ tuples, 25 tuples of R fit on one block, and 30 tuples of S fit on one block. Assume that teh buffer (main memory) has N pages which is less than the number of blocks of the smaller relation. Estimate the number of disk block transfers and seeks required, using each of the following join strategies for $R \bowtie S$:

a) Nested-loop join.

b) Block nested-loop join.

c) Merge join.

Part 3.

Consider the following schema:

Suppliers(*sid*: integer, *sname*: string, *address*: string)
Parts(*pid*: integer, *pname*: string, *color*: string)
Catalog(*sid*: integer, *pid*: integer, *cost*: real)

The key fields are underlined, and the domain of each field is listed after the field name. The Catalog relation lists the prices charged for parts by Suppliers. Write the following queries in SQL using **one** SELECT statement:

a) Find the names of suppliers who supply some red or some green parts.

b) Find the names of suppliers who supply some red and some green parts.