Department of Computer Science  
Core Graduate Comprehensive Exam  
Fall 2012

- Answer the questions on the paper supplied.
- Answer question 1 or Answer question 2. Answer question 3 or 4. Answer question 5 or 6 or 7. Answer question 8 or 9. You should answer a total of four questions.
- Start each question on a new page. Write on only one side of the paper.
- Write your SIX-DIGIT Texas State ID in the top right corner of each page of your answer. Do NOT put your name anywhere on the answers.
- Put the number of the question being answered in the top left corner of each answer page. Put the CORRECT question number to avoid missing your answer.
- If the answer to a question is written on more than one page, number the pages consecutively.

**Group 1**

1. **CS 5329 Algorithm Design and Analysis**
   Use recursion tree and log algebra to find the total running time of the recurrence equation:
   \[ T(n) = T(n/3) + T(2n/3) + cn \]
   (You are required to write the detail algebra for the calculation.)

2. **CS 5329 Algorithm Design and Analysis**
   Provide a pseudocode for the Insertion Sort algorithm. Analyze the running time performance of your implementation of the Insertion Sort in terms of Big-Oh notation separately for "assignments" and "comparisons" for the best-case scenario.

**Group 2**

3. **CS 5346 Advanced Artificial Intelligence**
   (a) A database of a county bank's loan department is represented as positive and negative instances in the following table:
   
<table>
<thead>
<tr>
<th>Individual</th>
<th>INC</th>
<th>BAL</th>
<th>APP</th>
<th>RATING</th>
<th>OK</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

   Derive a set of rules that covers all and only the positive instances.

   (b) (i) An algorithm has the following evaluation function:
   \[ f(n) = (2 - w) * g(n) + w * h(n) \]
   For what kind of search does this perform for \( w = 0, w = 1, w = 2 \)? For what values of \( w \) is it complete and optimal?

   (ii) Another algorithm uses the following evaluation function:
   \[ f(n) = (2 - W) * [g(n) + W/(2 - W) * h(n)] \]
   Assuming \( h(n) \) is admissible, what value of \( W \) will make it behave exactly like \( A^+ \)?

4. **CS 5391 Survey of Software Engineering**
   (a) state the pros and cons of lines of code and function points metrics
   (b) describe incremental development model and its limitation
5. **CS 5306 Advanced Operating Systems**
Suppose a machine with 4 physical pages starts running a program (in other words, the physical pages are initially empty). The program references the sequence of virtual pages as follows: A B C D E D C B A E D C B A C E

(a) Explain each of the following paging algorithms.
(b) For each paging algorithm, replicate the reference pattern and underline each reference that causes a page fault:
   i. LRU
   ii. FIFO
   iii. Optimum

6. **CS 5310 Network and Communication Systems**
Define the link utilization of the Idle RQ protocol. Explain using your own words the concepts of link utilization. Is higher link utilization always good? Justify your answer.

7. **CS 5332 Data Base Theory and Design**
Construct an E/R diagram for a database system for the following problem. You must specify the PKs, FKs, and relationships for each entity. Your relations (entities) must be normalized up to 3NF or BCNF. (Prefer you use crow feet notation for E/R model.)
The problem: A large organization has several parking lots, which are used by staff. Each parking lot has a unique name, location, capacity, and number of floors (where appropriate). Each parking lot has parking spaces, which are uniquely identified using a space number. Members of staff can request the use of a parking space. Each member of staff has a unique number, name, telephone extension number, and vehicle license number.

8. **CS 5318 Design of Programming Languages**
Lambda calculus is the basis of functional programming. Assume the following functions in lambda calculus:

\[
square = (\lambda x. \ast xx)\\
\]

\[
twice = (\lambda x. + xx)\\
\]

(a) Explain the applicative order and normal order reduction strategies.
(b) Show the steps in the applicative order and normal order reductions of the expression (twice (twice square))
(c) Compare the applicative order and normal order reduction strategies in terms of reduction efficiency.

9. **CS 5338 Formal Languages**
1 ~ 20 are True or False questions. 21 and 22 are short-answer questions. Some terms used in the questions are listed below:

- DFSM: Deterministic Finite State Machine
- NDFSM: Non-deterministic Finite State Machine
- DPDA: Deterministic Pushdown Automata
- NDPDA: Non-deterministic Pushdown Automata
- DTM: Deterministic Turing Machine
- NDTM: Non-deterministic Turing Machine
- D: Decidable
- SD: Semi-decidable
- \(w\): An input string
- \(L\): A language
- \(M\): A machine
(a) A DFSM $M$ must halt for any $w$.
(b) An NDFSM $M$ must halt for any $w$.
(c) For any NDFSM $M$ we can find a DFSM $M'$, such that $L(M) = L(M')$.
(d) Let $M$ be an NDFSM. If $M$ does not have $\varepsilon$-transitions, then $M$ must halt for any $w$.
(e) A DPDA $M$ must halt for any $w$.
(f) An NDPDA $M$ must halt for any $w$.
(g) For any NDPDA $M$, we can find a DPDA $M'$, such that $L(M) = L(M')$.
(h) For any NDPDA $M$, we can find an NDPDA $M'$ that halts, such that $L(M) = L(M')$.
(i) Let $M$ be an NDFSM. We can find some DPDA $M'$ that decides $L(M)$.
(j) Any regular language or context-free language can be decided by some TM.
(k) A DTM $M$ must halt for any $w$.
(l) If both $L$ and $\neg L$ are in SD, then $L$ is in D.
(m) NP-complete is a subset of NP-hard.
(n) If $P = NP$, then NP-complete languages will be in P as well.
(o) The union of a finite number of countably infinite sets is countable.
(p) The union of a countably infinite number of countably infinite sets is countable.
(q) Regular languages and context-free languages are in P.
(r) If $L = L_1 \cap L_2$ and $L$ is regular, then $L_1$ and $L_2$ must be regular.
(s) If $L$ is regular, then $\neg L$ is also regular.
(t) $\neg A^nB^nC^n$ is context-free, but $A^nB^nC^n$ is not. Thus context-free languages are not closed under complement.
(u) Given the FSM $M$ as in Fig. 1, write down the first five strings in $L(M)$ in lexicographical order.

![Figure 1: An FSM.](image1)

(v) Given the PDA $M$ as in Fig. 2, write down the first five strings in $L(M)$ in lexicographical order.

![Figure 2: A PDA.](image2)