Department of Computer Science
Core Graduate Comprehensive Exam
Fall 2019

- **Answer the questions on the paper supplied.**
- **Answer question 1 or 2. Answer question 3 or 4 or 5. Answer question 6 or 7 or 8 or 9 or 10. Answer question 11 or 12 or 13 or 14. You should answer a total of four questions. Please Note: If you answer more than one question in one group, only the one with the LOWEST score will be counted.**
- **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**
{*from Dr. Hwang*}

(a) **Write** the program algorithm in some detail for the Partition function in the following QuickSort (A, p, r) function, given A an array with the first index p and with the last index r.

```
QuickSort (A, p, r)

If p < r

Then q <- Partition(A, p, r)

QuickSort (A, p, q-1)

QuickSort (A, q+1, r)
```

(b) Assuming the Partition function of QuickSort always splits the sub-array of array A into 20% and 80%, and n is size of array A, **use recursion tree** and **log algebra** to **derive** the computation time T(n) for such special case of QuickSort in terms of big O **or** in terms of Theta. (Both proof and result are required.)

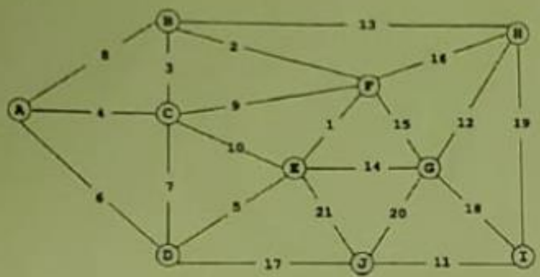
2. **CS 5329 Algorithm Design and Analysis**
{*from Dr. Metsis*}

Consider the following weighted graph.

a) Show the order in which Prim's algorithm (given below) includes edges in the Minimum Spanning Tree (MST) when starting from vertex r=A.

b) Analyze the time complexity of the algorithm using a binary heap as a priority queue (Q).

c) Compare with the time complexity of using an unsorted array instead of a heap to implement the priority queue and comment on that.




```
MST-PRIM(G, w, r)
for each u ∈ G.V
    u.key = ∞
    u.π = NIL
r.key = 0
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    for each v ∈ G.Adj[u]
        if v ∈ Q and w(u, v) < v.key
            v.π = u
            v.key = w(u, v)
```

Group 2

3. **CS 5346 Advanced Artificial Intelligence**
   {*from Dr. Ali*}

   (a) Explain (in detail with examples) how Expert Systems use Forward Chaining and Backward Chaining methodologies in making decisions. Also compare advantages and disadvantages of these two chaining methodologies.

   (b) Consider the Wumpus environment E as a 4*X*4 grid of rooms (E2,3 represents 2nd row and 3rd column). Sensors have detected the following facts:

   S1,3; S2,2; S2,3; S2,4; S3,3. Where Si,j indicates stench in the i th row and j th column. Stench is not detected in any other room.

   Write rules in propositional logic to determine the position of the Wumpus. Using resolution rule, develop a proof tree to determine the location of the Wumpus.

4. **CS 5391 Survey of Software Engineering**
   {*from Dr. Chen*}

   Explain the following software process models and point out their pros and cons:

   (a) Formal methods

   (b) Agile development model

5. **CS 5391 Survey of Software Engineering**
   {*from Dr. Palacios*}

   The Capability Maturity Model (CMM) was developed as a Software Process Improvement framework.

   (a) What is this model predicated on?

   (b) Give examples of the Key Process Areas that the model assesses, and

   (c) Describe the six Capability Levels defined by the model.

Group 3

6. **CS 5306 Advanced Operating Systems**
   {*from Dr. Tamir*}

   Assume a single core system implementing an intra-core preemptive HRRN scheduling policy with a slice size of 1 second. Further assume that at time T there are 3 tasks $\{T_0, T_1, T_2\}$ in the Ready Queue of the core with no task in the execution slot of the core. Additionally, assume that the tasks are compute-bound with no I/O whatsoever. Let $\{P_0, P_1, P_2\}$ be the remaining execution time of $\{T_0, T_1, T_2\}$ respectively and let $\{P_3, P_4, P_54\}$ be the current wait time of $\{T_0, T_1, T_2\}$ respectively, where:

$P_0 = 1,$
$P_1 = 4,$
$P_2 = 2,$
$P_3 = 1,$
$P_4 = 2,$
$P_5 = 3.$

Clearly describe the state of the system in each of the first 20 seconds following time T.

7. **CS 5306 Advanced Operating Systems**
   {*from Dr. Chen*}
   Design a distributed algorithm to synchronize physical clocks in a distributed system in a room.

8. **CS 5332 Data Base Theory and Design**
   {*from Dr. Hwang*}
   (a) Normalize the following table to 3NF tables and expressed as tables or as schema
   (b) Draw an E/R diagram to model the entities from (a).
   You must use crow feet notation for relationship. If you can not master
   the crow feet notation, you should not pick this problem.
   Your entities must specify the primary keys and foreign keys.

ClientRental

clientNo	propertyNo	cName	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	PG4	John Kay	6 Lawrence St, Glasgow	1-Jul-12	31-Aug-13	350	CO40	Tina Murphy
CR76	PG16	John Kay	5 Novar Dr, Glasgow	1-Sep-13	1-Sep-14	450	CO93	Tony Shaw
CR56	PG4	Aline Stewart	6 Lawrence St, Glasgow	1-Sep-11	10-Jun-12	350	CO40	Tina Murphy
CR56	PG36	Aline Stewart	2 Manor Rd, Glasgow	10-Oct-12	1-Dec-13	375	CO93	Tony Shaw
CR56	PG16	Aline Stewart	5 Novar Dr, Glasgow	1-Nov-14	10-Aug-15	450	CO93	Tony Shaw

9. **CS 5332 Data Base Theory and Design**
   {*from Dr. Ngu*}
   Consider the following typed relational schema describing books, authors, borrowings and borrowers in a library:

   Book(isbn:integer, title:string, publisher:string, year:integer)
   Author(isbn:integer, name:string, rank:integer)
   Borrower(bid:integer, name:string, address:string)
   Borrowings(isbn:integer, borrower:integer, whenTaken:date, whenReturned:date)

   Assume the following:

   - the fields that are underlined are the primary keys of the relations
   - there is a referential integrity constraint between "borrower" in relation Borrowings and "bid" in relation Borrower
   - there are also referential integrity constraints between "isbn" in relation Book and "isbn" in relations Author and Borrowings

- dates are represented as a count of the number of days since Jan 1 2019
- there is a system function `today` that returns today's `date`
- the relational operators $<$ and $>$ allow you test *before* and *after* on `dates`
- all loans are for a period of 14 days
- the `whenReturned` field is `null` until the book is returned
- the `rank` field specifies whether the author is the first, second, ... author

Implement solutions to the following:

(a) Write an SQL query to show all the books (`title` only) written by a particular author called "Ullman".

(b) Write an SQL query to show which books (`title` only) have multiple authors.

(c) Write an SQL query to show which books (`title` only) have never been borrowed.

(d) Write an SQL query to show which books (`title` only) are currently out on loan.

(e) Write an SQL query to list overdue books (`title`) and the borrower's name

10. **CS 5310 Network and Communication Systems**
{*from Dr. Peng*}

(a) Describe the steps and protocols used when you send an email message using a computer in CS lab to a friend of yours who uses an email server on a campus in California.

(b) Describe briefly why network layer is not an end-to-end layer.

Group 4

11. **CS 5338 Formal Languages**
{*from Dr. Gao*}

(a) Draw a PDA M for language $L(M) = \{a^{3n}b^n : n \geq 0\}$.

(b) In plain English, describe a turning machine M that decides $a^n b^n c^n$.

12. **CS 5338 Formal Languages**
{*from Dr. Singh*}
What are different ways to check for membership?

13. **CS 5318 Design of Programming Languages**
{*from Dr. Shi*}

(a) (20pts) In the pure lambda calculus, constants are defined as functions. We can code a natural number by the number of times a function parameter is applied as follows:

$$\begin{aligned}
0 &\equiv \lambda f.\lambda x.x \\
1 &\equiv \lambda f.\lambda x.fx \\
2 &\equiv \lambda f.\lambda x.f(fx) \\
3 &\equiv \lambda f.\lambda x.f(f(fx)) \\
&\cdots \\
6 &\equiv \lambda f.\lambda x.f(f(f(f(f(fx)))))
\end{aligned}$$

We can define the *mult* function for multiplication as follows:

$$mult \equiv \lambda m.\lambda n.\lambda f.m(nf)$$

i. Explain the applicative order and normal order reduction strategies.
ii. Show that $mult\ 2\ 3 = 6$ using the applicative order reduction strategy. How many β-reductions are needed?
iii. How many β-reductions would be needed if the normal order reduction strategy were followed in the above?

CS 5351 Parallel Processing

{*from Dr. Burtscher*}

Assume we want to parallelize the outer loop of the following code on a shared-memory system without caches.

```
for (int i = 0; i < n; i++) {
  int a = i;
  for (int j = 0; j < i; j++) {
    ... // do O(1) work here that reads variable a
    a += 3;
    ... // do some other O(1) work here that reads variable a
  }
}
```

1) Which of the following three workload partitioning schemes results in the least amount of load imbalance and why: blocked, cyclic, or block-cyclic?

2) If the system had a data cache, would a different workload partitioning scheme be better? If not, why not? If so, under what conditions and why?

3) Assuming that *n* is a large positive number and that each *i*-loop iteration is assigned to a separate thread (and no other parallelization is used), what is the parallel efficiency?

4) Assume each thread has a private variable called "*rank*" that contains the thread's unique rank. Assume further that there is a shared variable called "*threads*" that contains the number of running threads and that all ranks are in the range 0 through *threads* - 1. Using these variables, rewrite the above code such that it assigns the *i*-loop iterations cyclically to the threads.

5) Rewrite the above code such that the loop-carried data dependence on variable "*a*" is eliminated but the code maintains its semantics.