Department of Computer Science
Core Graduate Comprehensive Exam
Spring 2021

- **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. Answer question 10 or 11 or 12 or 13. 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*}

   (1) Write the procedure of BUILD-MAX-HEAP(A), A is an array of size n.

   (2) Draw a binary tree to represent A
   where A=(8, 32, 27, 10, 20 , 6, 20, 84, 9, 14)

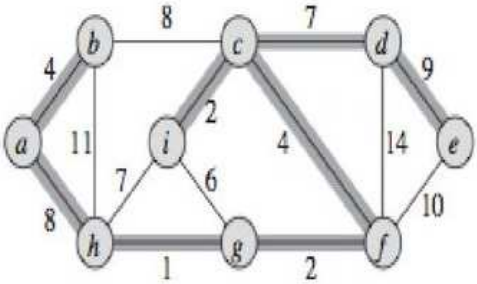
   (3) Do operation of BUILD-MAX-HEAP(A) on the A above.
   (step by step operation is required.)

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

   Question: Shortest Paths

   Given a connected, undirected graph G, a shortest-path tree rooted at vertex v is a spanning tree T of G, such that the path distance from root v to any other vertex u in T is the shortest path distance from v to u in G.

   Assume the following graph:



   a) Do the shaded edges of the graph form a shortest path tree rooted at vertex a? If your answer is yes, justify your answer. If your answer is no, re-draw the graph marking a different set of edges that form a shortest path tree. Also, please show the shortest distance of each vertex from a in your new tree.

   b) Design an algorithm that given a graph G(V, E) and one of its spanning trees T, can determine if T is a shortest path tree of G in O(E) time.

Group 2

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

   1. a. A database of a county bank's loan department is represented as positive and negative instances in the following table:

Individual	INC	BAL	APP	RATING	OK
1	1	0	0	1	0
2	0	0	1	0	0
3	1	1	0	1	1
4	0	1	1	1	1
5	0	1	1	0	0
6	1	1	1	0	1
7	1	1	1	1	1
8	1	0	1	0	0
9	1	1	0	0	0

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

   1. 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**
   { *from Dr. Chen*}
   Describe the following software development processes.

   - agile
   - cleanroom

5. **CS 5391 Survey of Software Engineering**
   { *from Dr. Palacios*}
   The Capability Maturity Model (CMM) was developed as a Software Process Improvement framework.

   - What is this model predicated on?
   - Give examples of the Key Process Areas that the model assesses, and
   - Describe the five 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 $\{P_3, P_4, P_5\}$ be the current wait time of $\{T_0, T_1, T_2\}$ respectively, where:

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

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

7. **CS 5332 Data Base Theory and Design**
   { *from Dr. Ngu*}

   Consider the following employee database, where the primary keys are underlined.

   Employee(<u>ename:string</u>, street:string, city:string)
   Works(<u>*employee:string, company:string*</u>, salary:real)
   Company(<u>cname:string</u>, city:string)
   Manages(<u>*employee:string, company:string*</u>)

   Give a single SQL statement for each of the following queries:

   (1) Find the names, street, and cities of residence of all employees who work for the "First Bank Corporation" and earn more than $40,000.
   (2) Find the names of all employees in the database who live in the same cities as the companies for which they work.
   (3) Give all managers of "First Bank Corporation" a 10 percent salary raise.
   (4) Find the name and, salary of employees in the database who earn more than every employee of 'the 'Small Bank Corporation".
   (5) Find the name of the company that has the most employees. Your solution should not involve an intermediate view.

8. **CS 5332 Data Base Theory and Design**
   { *from Dr. Hwang*}

   (1) Draw a complete E/R Diagram from the following tables by using crow feet notations including pk's, fk's, and relationships.
   (2) Display the resulting table (Not SQL statements) by naturally join the Client table and the Viewing table from the tables in this page.

Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Viewing

clientNo	propertyNo	viewDate	comment
CR56	PA14	24-May-13	too small
CR76	PG4	20-Apr-13	too remote
CR56	PG4	26-May-13	
CR62	PA14	14-May-13	no dining room
CR56	PG36	28-Apr-13	

PropertyForRent

propertyNo	street	city	postcode	type	rooms	rent	ownerNo	staffNo	branchNo
PA14	16 Holhead	Aberdeen	AB7 5SU	House	6	650	CO46	SA9	B007
PL94	6 Argyll St	London	NW2	Flat	4	400	CO87	SL41	B005
PG4	6 Lawrence St	Glasgow	G11 9QX	Flat	3	350	CO40		B003
PG36	2 Manor Rd	Glasgow	G32 4QX	Flat	3	375	CO93	SG37	B003
PG21	18 Dale Rd	Glasgow	G12	House	5	600	CO87	SG37	B003
PG16	5 Novar Dr	Glasgow	G12 9AX	Flat	4	450	CO93	SG14	B003

Client

clientNo	fName	lName	telNo	prefType	maxRent	eMail
CR76	John	Kay	0207-774-5632	Flat	425	john.kay@gmail.com
CR56	Aline	Stewart	0141-848-1825	Flat	350	astewart@hotmail.com
CR74	Mike	Ritchie	01475-392178	House	750	mritchie01@yahoo.co.uk
CR62	Mary	Tregear	01224-196720	Flat	600	maryt@hotmail.co.uk

PrivateOwner

ownerNo	fName	lName	address	telNo	eMail	password
CO46	Joe	Keogh	2 Fergus Dr, Aberdeen AB2 7SX	01224-861212	jkeogh@lhh.com	********
CO87	Carol	Farrel	6 Achray St, Glasgow G32 9DX	0141-357-7419	cfarrel@gmail.com	********
CO40	Tina	Murphy	63 Well St, Glasgow G42	0141-943-1728	tinam@hotmail.com	********
CO93	Tony	Shaw	12 Park Pl, Glasgow G4 0QR	0141-225-7025	tony.shaw@ark.com	********

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

   (1) Briefly using your own words explain why the CSMA/CD protocol requires the minimum length of a frame to be 64 bytes.

   (2) Explain in your own words how limited bandwidth can impair communication.

   (3) Compare TCP and UDP protocols. Describe their strength and weakness.

Group 4

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

Questions 1 $\sim$ 20 are True or False questions. 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

(1) A DFSM M must halt for any w.

(2) An NDFSM M must halt for any w.

(3) For any NDFSM M we can find a DFSM M', such that $L(M) = L(M')$.

(4) Let M be an NDFSM. If M does not have ε-transitions, then M must halt for any w.

(5) A DPDA M must halt for any w.

(6) An NDPDA M must halt for any w.

(7) For any NDPDA M, we can find a DPDA M', such that $L(M) = L(M')$.

(8) For any NDPDA M, we can find an NDPDA M' that halts, such that $L(M) = L(M')$.

(9) Let M be an NDFSM. We can find some DPDA M' that decides $L(M)$.

(10) Any regular language or context-free language can be decided by some TM.

(11) A DTM M must halt for any w.

(12) If both L and $\neg L$ are in SD, then L is in D.

(13) NP-complete is a subset of NP-hard.

(14) If P = NP, then NP-complete languages will be in P as well.

(15) The union of a finite number of countably infinite sets is countable.

(16) The union of a countably infinite number of countably infinite sets is countable.

(17) Regular languages and context-free languages are in P.

(18) If $L = L1 \cap L2$ and L is regular, then L_1 and L_2 must be regular.

(19) If L is regular, then $\neg L$ is also regular.

(20) $\neg A^n B^n C^n$ is context-free, but $A^n B^n C^n$ is not. Thus context-free languages are not closed under complement.

(21) Given the FSM M as in Fig. 1, write down the first five strings in $L(M)$ in lexicographical order.

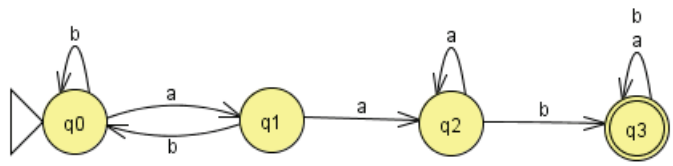


Figure 1: An FSM.

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

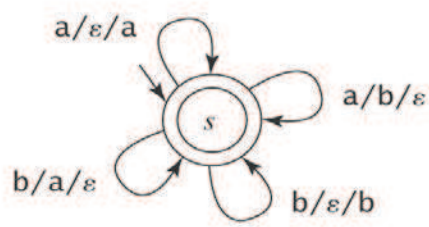


Figure 2: A PDA.

11. **CS 5338 Formal Languages**
{ *from Dr. Singh*}
What is Pumping Lemma for regular languages? Using Pumping Lemma prove $L = \{0^n 1 0^n | n \geq 1\}$ is not regular.

12. **CS 5318 Design of Programming Languages**
{ *from Dr. Shi*}
Consider the following grammar:

$$\begin{array}{lcl} <A> & ::= & <A>\ b\ <B> \ \mid \varepsilon \\ <B> & ::= & <B>\ a\ <A> \ \mid b \end{array}$$

(1) Give a leftmost derivation for the following input:

```
b b a b b a
```

(2) Draw the parse tree corresponding to the above leftmost derivation.

(3) Re-write the grammar in EBNF (Extended BNF).

(4) Write a recursive-descent parser for the grammar in EBNF.

(5) Design an attribute grammar to count the a's for any input.

(6) Decorate the parse tree (you drew in 2) to show how the number of a's is computed.

13. **CS 5351 Parallel Processing**
{ *from Dr. Burtscher*}
The following code fragment from a shared-memory parallel program sums up the return values of $func(0)$, $func(1)$, ..., $func(n-1)$. It uses busy waiting to provide mutual exclusion for updating the variable *sum*. The variable *thread_count* stores the number of running threads. The variable *thread_ID* is a unique thread identifier in the range $0...thread\_count - 1$.

```
// sum and flag are initialized to 0
for (i = thread_ID; i < n; i += thread_count) {
  res = func(i);
  while (flag != thread_ID) {};
  sum += res;
  flag = (flag + 1) % thread_count;
}
```

1) Specify, for each variable in the code, whether it should be shared or private and explain why.

2) Are the iterations of the for loop assigned to the threads in a blocked, cyclic, or block-cyclic distribution? Explain how you can determine that.

3) There are data races on some shared variables. List those variables and, for each of them, explain where the data race is coming from.

4) Explain whether the busy waiting can safely be replaced by a mutex. If so, explain whether the mutex would provide any performance benefit.

5) Explain whether the busy waiting can safely be replaced by an atomic operation. If so, explain whether the atomic operation would provide any performance benefit.