Ph.D. Comprehensive Examination

Computer Science Department
University of Miami

August 19, 2022

Student Name: Christopher Duarte Student Number: C12034480

Problem number	Points (10 max)
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
Total:	

1. **Data organization; Algorithms and complexity**

   Each item x in a set S has a unique key $key[x]$. We need to implement the following operations.

   (a) Search (S, key)
   (b) Insert (S, x)
   (c) Successor (S, x)
   (d) Predecessor (S, x)

   Give the 4 running times as an O() for the following implementations:

   (a) Ordered (sorted) array,
   (b) Ordered doubly linked list,
   (c) Min-Heap, and
   (d) Hash table

a) With an ordered (sorted) array the most optimal search is O(log n) though binary search.
For insert we have to traverse to the actual position in the sorted array then add the element in the correct position. It would take O(log n) for actual traversal and O(1) for actual input/insert. O(log n) total
For Successor it would be a similar to simply searching and viewing the right node in the BST of the selected node. Thus, O(log n) for traversal
The same rundin for Predecessor as it is simply comprised of traversal and view parent of current node.

b) For an ordered doubly linked list, would a runtime similar to a sorted array as the only difference is that each element in the doubly linked list is linked to the previous and next element in the list. Let's assume in this implementation we only have a pointer to the first element of the list. This effectively makes it the same as the ordered list via traversal using start pos + index.
Thus, traversal = O(log n), insert = O(log n), successor = O(log n), predecessor = O(log n)

c) For a min heap, the most costly operation would be to heapify the array. Heapify (a recursive if the min heap condition is not met) requires O(log n) swaps. Search in a Heap can take O(n) due to the need doing a simple linear search (no BST). Insert O(log n) total due to heapify (O(1) swap during add) (min bubbled to heapify).

d) Hash table guarantees O(1) search assuming good collision handling. Insert O(1) w/ collision handling. Same for successor and predecessor

2. **Program control and structure; Programming language and notations**

Suppose that procedure *swap* is declared as follows:

```
procedure swap( x, y: integer);
    procedure f(): integer;
        var z: integer;
        begin // f
            z = x; x = y; return z;
        end // f
    begin // swap
        y = f();
    end // swap
```

Describe the effect of the procedure call *swap*(i, A[i]) under each of the following parameter passing methods:

(a) Call-by-value

(b) Call-by-reference

(c) Call-by-value-result

a) Call by value passes a copy of the value stored for i and A[i] to x and y BUT does not alter i and A[i]. It saves a buffer to not affect both passed arguments i and A[i]. The only value affected would be y.

b) Call by reference would pass the addresses of both i and A[i] to the calling function. Depending on the implementation for the called function, it may change the values of i and A[i] in memory.

c) Call-by-value result is call by value but the result of the called function is stored in i or

3. **Software engineering**

From the software engineering point of view, any software development process can be divided into several sub-disciplines:

(a) Requirement Analysis
(b) Functional Specification
(c) Architectural Design
(d) Implementation
(e) Testing and Evaluation
(f) Maintenance

Choose three sub-disciplines or tasks within these sub-disciplines that involve a mathematical approach, and illustrative them with examples.

a) Requirement Analysis means analyzing which piece(s) or components of technology or requirements are necessary for running a project. This includes defining the amount of time required for the entire project, dividing a task into sub problems, and/determining which tools would suit a particular task. An example of a quantitative approach would be determining how long a particular task (or subtask) would take and allocate the appropriate time.

d) In the implementation stage, it is important to implement or develop the piece of software discussed in previous planning stages. If a software solution is being developed, a correct algorithm analysis is required. For example, if one chooses to implement a sorting algorithm, it is important to have a notion on what sorting algorithm is both space and time optimal

For example, choosing merge sort for worst case scenarios vs quicksort in a average case scenario. This can only be determined upon a given input and task to be accomplished.

e) Testing and Evaluation concerns making unit testing in verifying that the implementation is performing as expected/desired. For example if implemented quicksort in the previous check against both input model and size against desired target platforms. A simple check would be to verify if input sorted (if each elem. $\leq$ it+1) and if the correct number of calls/iterations were expected based on input size.

4. **Systems**

(a) _____ linked libraries can support shared library code, allowing one copy of a library routine to be used by several different processes.

absolute **relative** **static** **dynamic** **none of these is correct**

(b) When it is not known at compile time where a process will reside in memory, _____ code must be generated.

logical **physical** **absolute** **relocatable**

(c) A UNIX process calls *fork()* to create a child process as shown: *pid = fork()*;

i. What value will be assigned to pid in the parent process by the call to *fork()*?

the parent's process id **the child's process id** **zero none of these**

ii. What value will be assigned to pid in the child process by the call to *fork()*?

the parent's process id **the child's process id** **zero none of these**

(d) The Banker's algorithm is used for deadlock _____.

denial **prevention** **avoidance** recovery

can be considered as one

(e) Belady's anomaly can affect the performance of the _____ page replacement algorithm.

FIFO **LRU** **optimal** **SJF**

(f) _____ access files are made of fixed length records that allow programs to read and write records in no particular order.

sequential **direct** **logical** **none of these is correct**

(g) When an I/O request is being handled for a user's process, which term refers to the policy of returning control to the user process before the I/O is completed?

synchronous I/O **asynchronous I/O** **delayed I/O** **none of these**

(h) Which multithreading model requires that a new kernel thread be created for each new user thread?

many-to-one **one-to-one** **many-to-many** **none of these is correct**

(i) A process that does not affect, and is not affected by, another process is referred to as:

static **independent** **cooperating** **dynamic** **unbounded**

5. **Software, Programming Techniques**

Given that

$B(x)$ means "x is a bear"
$F(x)$ means "x is a fish", and
$E(x, y)$ means "x eats y",

what is the best English translation of
$\forall x[F(x) \rightarrow \forall y(E(y, x) \rightarrow B(y))]$?

(a) All fish eat bears.
(b) Every fish is eaten by some bear.
(c) Bears only eat fish.
(d) Every bear eats fish.
(e) Only bears eat fish.

$E(x,y)$ = x eats y
$B(x)$ = x is a bear

$\forall y(E(y,x)$
, For all y, if y eats x then y is a bear,

$F(x)$ = x is a fish

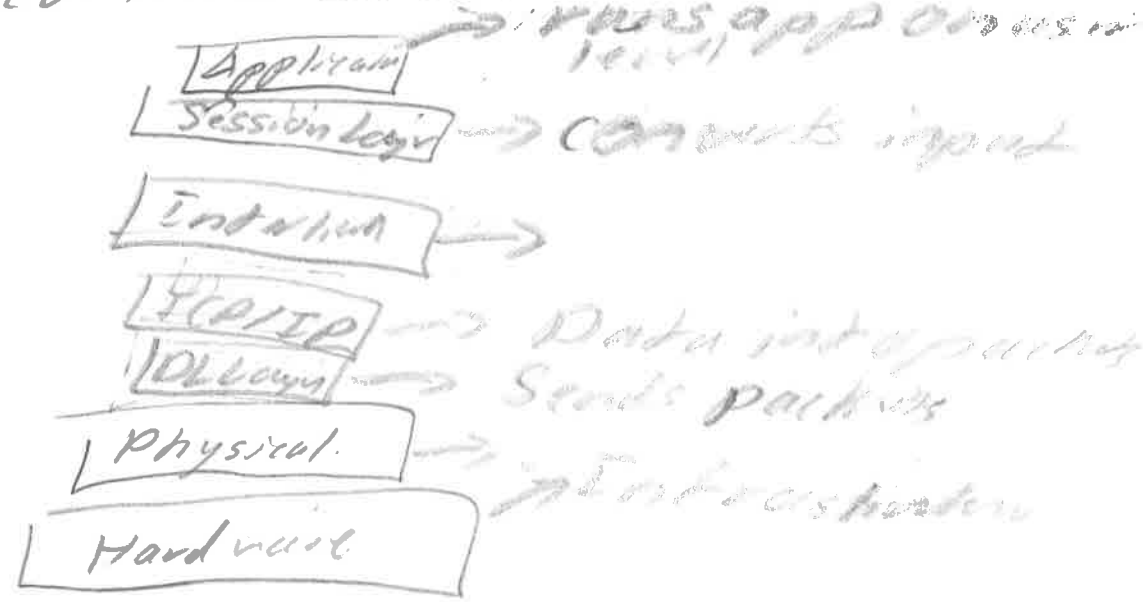
For all x, if x is a fish then if something eats it then that something is a bear.

6. Networking and Communications

(a) Draw a diagram showing layers of the Internet Protocol Stack and briefly discuss role of each layer.

(b) Describe functions of each layer when a file is transferred from a source to destination using (*file transfer protocol* (FTP).

a) OSI mode also known as the IP stack is composed of 7 layers each with their own functions and form of inter communication.



b) Hardware: The actual components that can physically seen and touched on a computer.

Physical: Involves physically sending signals over the wire (digital to analog conversion)

TCP/IP: Involves the protocol of packet exchange and routing the data to the correct router.

A=

7. **Algorithms and complexity**

Describe an algorithm that takes two input lists of integers $A = a_1, ..., a_n$ and $B = b_1, ..., b_m$ and delivers the list of all the elements that belong to A but not to B. A and B do not contain redundant elements, however, the elements of A and B might have a large range.

The algorithm should run in $O(n \log m + m \log m)$ time.

```
C = []  # to hold values
c_elem = null  # represents current element var
for i in range(0, len(A)):
    c_elem = A[i]
    // Sort Array B to guarantee
    O(log n) time. This would be NO(n log n)
    using Mergesort and O(log n) for
    BST to search for elem.
    ∴ O(n log n + log n) = O(n log n).
    if(binary_search(c_elem, B.sorted) == false)
        C.append(i)

}

return C.
```

8. **Automata and language theory**

Consider the following grammar:

$G \longrightarrow S\ \$\$$
$S \longrightarrow A\ M$
$M \longrightarrow S \mid \epsilon$
$A \longrightarrow a\ E \mid b\ A\ A$
$E \longrightarrow a\ B \mid b\ A \mid \epsilon$
$B \longrightarrow b\ E \mid a\ B\ B$

(a) Describe the language that the grammar generates in English.

(b) Show a parse tree for the string a b a a.

(c) Is the grammar LL(1)? If so, show the parse table; if not, identify a prediction conflict.

a) the language describes a sequence of a's and b's such that for each b there are two a's.

b) $G \rightarrow S \rightarrow AM \rightarrow aEM \rightarrow abAAM$

$\rightarrow abaEAM \rightarrow abaAM \rightarrow abaaE \rightarrow abaa$

c) Assuming this asks if there is simply one derivation per string in language,

$G \rightarrow S \rightarrow AM \rightarrow aEM \rightarrow abAM \rightarrow$
$abaEM \rightarrow ababAM \rightarrow ababaEM \rightarrow$
$ababa$

9. **Discrete Structures**

Recall that the Hamiltonian Cycle Problem is the problem of deciding, on input graph G, whether G has a cycle that visits all the nodes exactly once. Show that this problem is polynomial time decidable if the input is restricted to the graphs with the property that each node has at most two neighbors (i.e., at most two adjacent nodes).

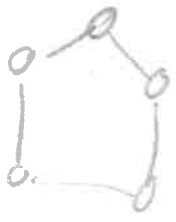
to demonstrate

It is possible if a graph visits all inputs exactly once in a cycle through a base step and then demonstrate an inductive step.

The most minimal graph that has at most two neighbors is as follows:

This clearly shows that a cycle can be a determined by starting at a node and using a sequence of distinct edges to navigate back to source. Now, for any n edges less than the # of nodes would violate the def of a cycle and more edges violate the overall premise.

Thus, if the starting graph is expanded:



graph composed of n nodes and n edges

Thus, it is possible to determine the Hamiltonian Cycle Problem in polynomial time as it correlates with the # of nodes/edges. Any less edges per node and a cycle isn't present.

10. **Other Topics**

Give a detailed explanation of any one approach to machine learning. Give a substantial example that illustrates the technical operation of the approach, and demonstrates interesting knowledge learned.

One simple approach used machine learning is binary classification through decision trees (DTS).

Example:

Simple Decision Tree to determine if someone is going to buy milk

The benefit of a decision tree is its simplicity. It reaches a binary decision in O(log n) assuming a balanced tree. Otherwise O(n) for degenerate. Simplicity is also a negative aspect of decision trees as a binary classification can be reached that is a misclassification. For example, doesn't have kids but older than 10 may still like drinking milk.

An aspect in which simple DTs can be expanded upon is through DT forests.

This involves merging several trees to form one forest which will enhance the overall decision capability of the chosen model by improving the accuracy of the classification.