Ph.D. Comprehensive Examination

Computer Science Department
University of Miami

August 19, 2022

Student Name: Katarzyna Pasternak Student Number: C11998922

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, $O(n)$
(b) Ordered doubly linked list, $O(n)$
(c) Min-Heap, and $O(n \log n)$
(d) Hash table $O(n \log n)$

```
Search (S, key)
    v = S(key)
    return v
```

```
Search
    foreach x in S
    if key[x] equals key  return index x
    elif index @ x equals S.LAST
        return error
```

```
Insert (S, x)
k = hash(x)
S(key, value) = (k, x)
```

```
Predecessor (S, x)
    last = x
for k, v in S
    if last == x and v < x
        last = v
elif v < x and v > last
        last = v
return hash(last), v
```

```
Successor (S, x)
    next = x
    for k, v in S
        if next == x and v > x
            next = v
        elif v > x & v < next
            next = v
    return hash(next), next
```

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) If we call ex swap(4,5) it will cause for the program to give error / not give a swap

b) If we call-by-reference after execution the values at reference will be swapped ~~not~~ by assigning new reference to x and y

c) it will change the value at x and at y

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.

d) implementation:
ex. finding a page in a book (could be a tool for a reading app)
page n ; we look for
total pages x
to find we find middle : $x/2 = k$ $x - k$ or $x + k$
and see if $n > k$ or $n < k$ now our k becomes x_1
then we go to the half and keep dividing by 2 till we find the page
if x is odd $x+1/2$

e) testing & evaluation:
$n = 13$ we can test our implementation
$x = 54$
↳ $54/2 = 27 \rightarrow n < 27$
↳ $(27+1)/2 = 14 \rightarrow n < 14$
↳ $14/2 = 7 \rightarrow n > 7$
↳ $(7+1)/2 + 7 = 11 \rightarrow n > 11$
↳ $4/2 + 11 = 13$ $n = 13$

as well evaluate it
for ex by evaluating running time : $O(\log n)$ ⇒ can be done with math

b) functional specification
we design our architecture to accommodate requirements of the software
and in order to find specific requirements we can evaluate needed components:
ex. needed space/memory for 1 or multiple operations at the same time for one or multiple users

4. **Systems**

(a) Dynamic 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, physical 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 avoidance.

denial prevention avoidance recovery

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

FIFO LRU optimal SJF

(f) Direct 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.

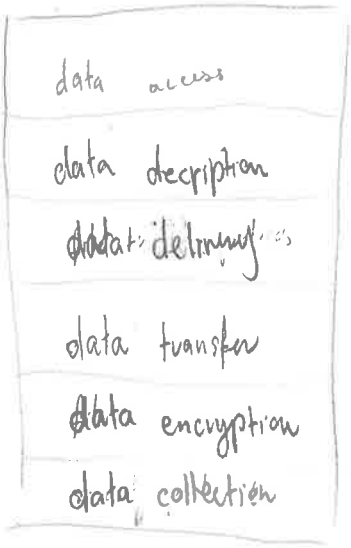
(e) Only bears eat fish.

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)



b)

← data is ready to send

← data is decrypted using a receiving end receive key

← data is received and confirmation of receiving sent

← data is transfred to destination in chunks if

← data is encrypted with a secure key

← data is collected

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.

f(A,B)
- sort array A
- sort array B log merge sort
- compare the two in a loop
  - traverse thru A compared to B as follow
    - if we find A[k] < B[l] we return it
    - if we find A[k] == B[l] we go to next k
    - if we find A[k] > B[l] we go to next l

8. **Automata and language theory**

Consider the following grammar:

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

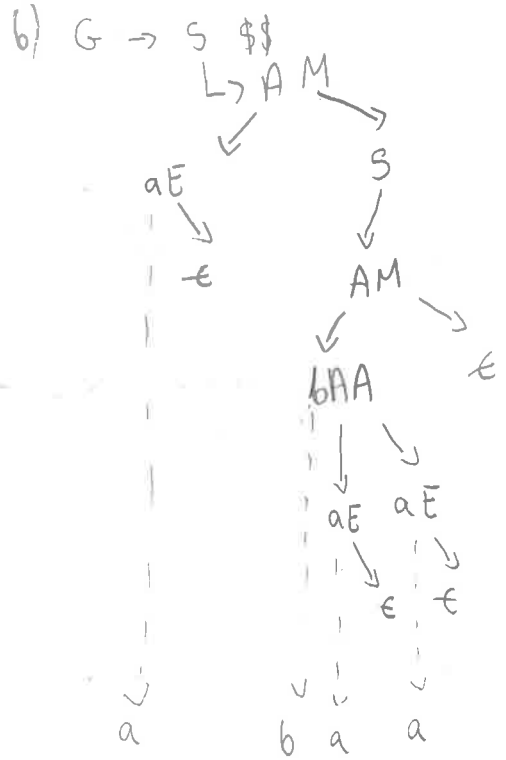
(a) Describe the language that the grammar generates in English.

(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.

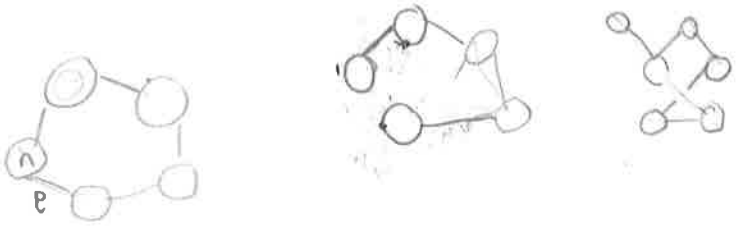
a) in this language we get words containing combinations of letters a and b with shortest being a single 'a'. There is not such word that doesn't cont... letter 'a'. it is impossible to have string such as $(ba)^+$

c) The grammar is not LL(1)



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 decide if it has a cycle visit visiting a

n - nodes
e - edges

$\frac{n}{e} = 1$ -> each node has 2 neighbors

$\frac{n}{e} \geq 1$ -> not all nodes have 2 neighbors

depending on the starting point we might have not or all nodes visited

for n node there is n possibilities with n edges

so each check take O(n) and n start points &: ....

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.

supervised learning lets say we want a machine to recognized diffrent kitchen items : fork, spoon, plate & bowl

we provide the image examples of these items with labes to the machine and let it learn. To improve its performance we might want to crop the images of the objects and paste them on diffrent background in diffrent size and orientation and provide bounding box of where we put the object on the background. As three object can appear together in the same frame, we can paste multiple object on one background and provied bounding box and lable of each. We can also scale them depending on the location of the image ex if object is further away we make it smaller and higher on the image etc. We can also provide negative examples to eliminate bias and improve the performance of our approach.

The interesting thing learned would be the performance on similar but not the same objects as used for training in comparison to used examples and to check if extra steps such as scaling or negative examples truly improve the performance of the model.