Ph.D. Comprehensive Examination

Computer Science Department
University of Miami

August 19, 2022

Student Name: Nathaniel Dean

Student Number:

Problem number	Points (10 max)
1	7.372
2	
3	0
4	
5	
6	4
7	10
8	
9	
10	10
Total:	

Point gained: 7.372

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) sorted array
search : O(lgn)
insert : O(n)
successor : O(1)
predecessor : O(1)

I'm assuming this is basically random access

b) doubly linked list
search : O(n)
insert : O(1)
successor : O(1)
predecessor : O(1)

The actual insertion is O(1), if you include searching for a particular position, its O(n)

c) min-heap
search : O(lgn)
insert : O(lgn)
successor : O(1)
predecessor : O(1)

wrong

d) Hash Table
search : O(n) worst
insert : O(n) worst
successor : O(n) worst
predecessor : O(n) worst

should give the best as well -2

I am assuming that even with "unique" keys there could be collisions. The problem does not say each item has a unique hash value in which case we could do these in O(1)

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
```

z is local variable it appears

z=x z gets x
x=y x gets y
return z

y gets output of f() which is z

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

local z = $\overset{x}{i}$
$\overset{x}{i}$ = A[i]
return z = i
A[i] = z

a) when we call procedure "swap" by value, we are not referencing the original A object, so we don't affect the original A object. Rather we are just affecting local copies of primitives, ie, whatever numbers i and A[i] are, but scope remains local

b) Calling by reference, we can affect the original object A. In this case $i_{new} \leftarrow A[i]$ and $A[i] \leftarrow i_{new}$.

c) I'm not sure what call by value result is

Point gained: 0

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.

b) Functional specification

The goal here is to develop a requirements document that will meet the needs of the customer and problem and can feed into a design document.

Mathematically, the problem and customer we are addressing may have quantitative requirements such as.

1) float precision of input and output
2) running time between input to answer
3) what memory capacity of computer will be running this program
4) what are the I/O interfaces and do these put limits on data flowrate

c) Architectural Design

Here we develop a design and design document that can meet the requirements of part b)
we will be choosing modules, how the code is structured, and programming languages.

Some mathematical things to consider:

1) For projects such as OS, what preemptive job scheduling algorithm should be used and where? Will bottlenecks/starvation be developed in certain parts of the design based on this.
2) what language will provide sufficient compute time for our problem, trading off against perhaps compile time and ease of use.
3) How many classes/modules can be used? Using more may be better for the team, but may slow down the program due to all the data passing between modules

9
More on back

e) Testing and Evaluation

In this phase we are checking the program for bugs and evaluating performance.

1) For the different input cases, do we meet the compute times in the requirements
2) Using a tool like gprof, which functions are taking up the most compute time and can these be addressed specifically to improve performance
3) For the list of bugs that are found we should measure:
    1) rate of occurrence
    2) severity
    3) risk to producing more bugs upon fixing

For bugs that are low on 1, 2 but high on 3, we may ignore them on the initial deployment

4) Track peak RAM memory usage to make sure it meets requirement

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, relocatable 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 SJF page replacement algorithm.

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

(f) Sequential 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
$\forall x[F(x) \rightarrow \forall y(E(y, x) \rightarrow B(y))]$?

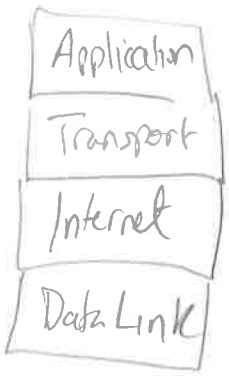
(a) All fish eat bears.
(b) Every fish is eaten by some bear.
(c) Bears only eat fish.
(d) Every bear eats fish.
(e) Only bears eat fish.

6. Networking and Communications

Point gained: 4

(a) Draw a diagram showing layers of the Internet Protocol Stack and briefly discuss role of each layer. 3

(b) Describe functions of each layer when a file is transferred from a source to destination using (*file transfer protocol* (FTP). 1



All of these layers act as interfaces for the layers above and below them. The protocols are instantiations of the layers.

The application layer contains protocols such as HTTP and FTP and is responsible for converting data in user facing applications into a format usable by and readable from the transport layer.

The transport layer provides reliable connection oriented services on top of an unreliable connectionless internet where packets can be lost, expired, or corrupted. The transport layer can do error (checksum) checking, sequencing of data, and flow control between connection points.

The internet layer is like the postal system it encapsulates the transport segments in packets and routes them through a vast array of routers to get to destination. This layer will control internet congestion and route path finding. It will break up and reassemble packets depending on available data rates.

The data Link layer is concerned with the local network before interfacing to the internet Ethernet is an example protocol.

FTP sets up two connection oriented services between user-client. One is for communication the other for data transfer. Once the user communicates they want the file, the server will start sequencing file data and send it through IP packets. The user's transport layer will confirm it has received all the data, resequencing the segments as necessary.

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.

point gained: 10

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

It makes sense to me to sort one of the lists so we can use binary search. Given the suggested $O()$ complexity:

sort B in $m \log m$ time, use mergesort if space permits or quicksort otherwise

Now, for each element in A: $O(n)$

BinarySearch(B, A_i) $O(\lg m)$

sorted array — target element

if BinarySearch == False (not found)

then A_i not in B

total time

$O(\underbrace{m \log m}_{\text{sort B}} + \overbrace{n}^{\text{each element in A}} \overbrace{\log m}^{\text{Binary Search B for } A_i})$

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.

AM
AS
AAM
aEbAA
abaEaE
abaa

AM
A
bAA
baEaE
baaBaaB
baaaBBaaaBB
baaabEbEaaabEbE
baaabbaaabb

AM
AS
AAM
bAA bAA
baEaE baEaE
baaBaaBaaBaaB
baabEaabEaabEaabE
baabaabaabaab

aE
abA
abaE
ababA

a) It generates a string of any length of as and bs excluding the empty string and 'b'. It can finish with $\$\$$ if G is the true start symbol. There is at least one 'a'.

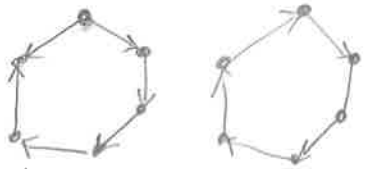
b.

c. I'm not sure what LL(1) means. It perhaps has something to do with the bijectivity of the language. Meaning there are multiple parse trees that can result in same string. It may not be LL(1) because it looks like there are multiple ways to get the same string.

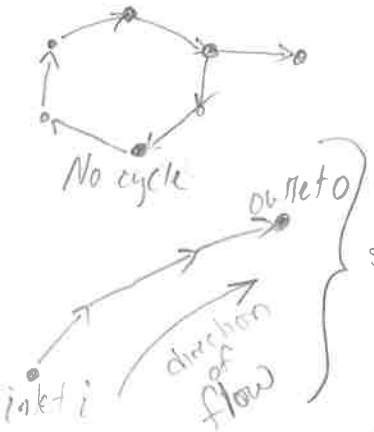
9. Discrete Structures

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).

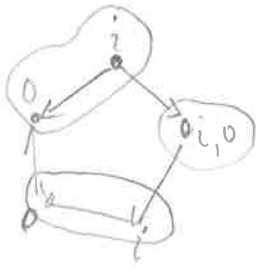
I will show a constructive proof by developing an algorithm that runs in polynomial time. The problem does not state whether the graph is directed or undirected. I will assume both are possible. I will also assume "at most two adjacent nodes" means that there are a maximum of two reachable nodes in the adjacency list of each node.



We take a safe-edge with disjoint sets approach, we will build sets of edges where the directionality is maintained, We can use depth-first-search in $O(V+E)$ polynomial time to build our sets.



Each vertex starts with attributes V.i and V.o set to True to indicate they could be the inlet or outlet of a set. Each vertex starts as the head of a linked list.




```
For each v in G.V
    if v.merged == False
        Safeedgesearch(v)

Safeedgesearch(v)
    For each neighbor in v.neighbor
        if(neighbor.merged == False)
            v.i, v.o = False
            v.head.i = True //set inlet for set
            Union(neighbor, v.head)   // neighbor's head becomes v's head
            neighbor.o = True //set outlet for set
            v.merged = True
            neighbor.merged = True
            Safeedgesearch(neighbor)   // recursive call
```

Check cycle
Foreach v, check if neighbor matches i,o
merge sets by matching inlets to outlets. Then check if the length of the largest set is equal to E, ie |V|=|E| in which case a cycle exists, otherwise it does not.

The total time should be $O(\underbrace{V+E}_{\text{DFS}} + \underbrace{V}_{\text{unions}} + \underbrace{V}_{\text{cycle check}})$

10. **Other Topics**

point gained: 10

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.

Lets take the problem of binary classification using logistic regression. We are given N feature vectors x_i $i=1\ldots N$ with corresponding correct labels y_i $i=1\ldots N$ where $y_i \in [0,1]$. Each feature vector itself has D features so $x_i = [x_{i1}, x_{i2}, \ldots x_{iD}]$.

For logistic regression, our hypothesis is $\ln\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_{i1} + \ldots \beta_D x_{iD}$ where the weight vector $\beta = [\beta_0\ \beta_1 \ldots \beta_D]$ is the weight vector we want to learn. Note we apply $\{1\} \cup \{x_i\}$ for the β_0 bias term. p_i is the probability that x_i belongs to class 1 and thus $1-p_i$ is the probability that x_i belongs to class 0

$$z_i = \beta^T x_i$$

$$\ln\left(\frac{p_i}{1-p_i}\right) = z_i$$

Create a loss function over all our samples

$$\min\left[J = -\sum_{i=1}^{N}\left[y_i \ln p_i + (1-y_i)\ln(1-p_i)\right]\right] \quad \text{cross entropy form}$$

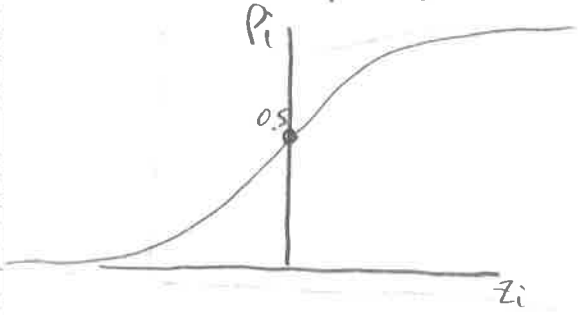
$$p_i = (1-p_i)e^{z_i}$$

$$p_i = e^{z_i} - p_i e^{z_i}$$

$$p_i(1+e^{z_i}) = e^{z_i}$$

$$p_i = \frac{e^{z_i}}{1+e^{z_i}} \cdot \frac{e^{-z_i}}{e^{-z_i}}$$

$$p_i = \frac{1}{1+e^{-z_i}}$$



Interesting knowledge...

J can be rewritten

$$\max\left[J = \sum_{i=1}^{N} \ln p_i^{y_i} + \ln(1-p_i)^{(1-y_i)}\right]$$

$$\max\left[J = \prod_{i=1}^{N} \ln p_i^{y_i}(1-p_i)^{(1-y_i)}\right]$$

minimizing our cross entropy is same as maximizing log likelihood of the data we see

we train our weights w/ stochastic gradient descent $\frac{\partial J}{\partial p_i}\frac{\partial p_i}{\partial \beta_i}$. I will come back if I have more time

$$\frac{\partial J}{\partial p_i} = -\left[\sum_{i=1}^{N} y_i\left(\frac{1}{p_i}\right) - (1-y_i)\left(\frac{1}{1-p_i}\right)\right]$$

$$p_i = \frac{1}{1+e^{-z_i}}$$

$$\frac{\partial p_i}{\partial \beta_{ij}} = \frac{\partial}{\partial \beta_{ij}}\left[\frac{1}{1+e^{-f(\beta_{ij})}}\right] = -\left(1+e^{-f(\beta_{ij})}\right)^{-2}\left(-e^{-f(\beta_{ij})}\right)\frac{\partial f(\beta_{ij})}{\partial \beta_{ij}}$$ (X_{ij})

$$= \frac{e^{-f(\beta_{ij})}}{\left(1+e^{-f(\beta_{ij})}\right)^2} = \frac{1-1+e^{-f(\beta_{ij})}}{\left(1+e^{-f(\beta_{ij})}\right)^2}$$

$$= \frac{1+e^{-z_i}}{\left(1+e^{-z_i}\right)^2} - \frac{1}{\left(1+e^{-z_i}\right)^2} = \frac{1}{\left(1+e^{-z_i}\right)} - \frac{1}{\left(1+e^{-z_i}\right)^2}$$

$$= \frac{1}{\left(1+e^{-z_i}\right)}\left(1 - \frac{1}{1+e^{-z_i}}\right)$$

$$= p_i(1-p_i)x_{ij}$$

So $$\frac{\partial J}{\partial \beta_{ij}} = -\left[\sum_{i=1}^{N} y_i\left(\frac{1}{p_i}\right) - (1-y_i)\left(\frac{1}{1-p_i}\right)\right] p_i(1-p_i)x_{ij}$$

$$= -\left[\sum_{i=1}^{N} y_i(1-p_i)x_{ij} - (1-y_i)p_i x_{ij}\right]$$

Then for gradient descent (minibatch, in this case)

$$\beta_{ij} \leftarrow \beta_{ij} - \alpha\frac{\partial J}{\partial \beta_{ij}}$$

(α: learning rate)