Ph.D. Comprehensive Examination

Computer Science Department
University of Miami

August 19, 2022

Student Name: Melanie Xia Student Number: C12043533

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

① Sorted Array

a. Search : O(n) if linear search
O(log n) if binary search

b. Insert : O(n) assuming the original array is full and you need to create a new array with the new element

c. Successor : O(n) or O(log n) depending on search method

d. Predecessor : O(n) or O(log n) depending on search method

② Ordered doubly linked list

a. Search : O(n)

b. Insert : O(1) because you just rearrange the pointers

c. Successor : O(n)

d. Predecessor : O(n)

③ Min-Heap

a. Search : O(n) because you have to look through entire heap. values added at the first available space if larger than parent nodes

b. Insert : O(n)

c. Successor : O(n)

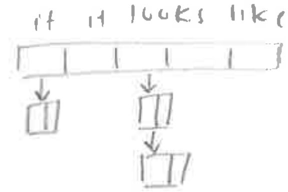
d. Predecessor : O(n)

④ Hash Table

a. Search :

O(1) if it looks like

O(n) if it looks like



b. Insert :

O(1) or O(n) depending on coll

c. Successor : O(n)

d. Predecessor : O(n)

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

call y directly to run swap

b. call-by-reference

call f( ) to run swap

c. call-by-value-result

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:
① Taking the functions and pseudocode and turning it into runnable code (variables probably include math...)
If your project requires a database, such as SQL, table joins are also math (e.g. intersection of databases, union)

(b) Functional Specification
① Team needs to determine how much software and other resources will cost. This might be based on a tentative number of users that the clients expect to have.

(e) Testing and Evaluation
① Check for bugs, discrepancies, etc.
- If you are creating a video game, and everyone forgets to limit the number of players allowed for a multiplayer game, too many people entering server can cause it to crash.
- Stress testing
- In web development, check the bandwidth

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**

(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

   $\forall x[F(x) \to \forall y(E(y, x) \to B(y))]$?

   (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.

$B(y)$: y is a bear $E(y,x) \to B(y)$: bear eats fish
$F(x)$: x is a fish

$\forall$fish[fish $\to$ $\forall$bear(bear eats 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) ① Application : client communicates with a host server

- HTTP : Hypertext Transfer Protocol
- FTP : File Transfer protocol
- DNS : Domain Name Server

② Transport : Client and host agree upon a certain set of rules (e.g. size of packages). Data collected at sockets, transferred to network layer

- TCP : reliable, connection oriented
  ↳ checks to ensure that data has been received
- UDP : faster, but not connection oriented
  ↳ no checks to ensure data was received

③ Network : data is split up into packages and transferred

- IPv4
- IPv6

} every device that is connected to the internet is assigned a unique IP address. IPv4 was running out of IPs, so IPv6 was created

④ Link / physical : where the hardware lies. The packages are split so that they can be transported through physical components (e.g. ethernet)

Diagram:

Application
Transport
Network
Link / Physical

(b) Application Layer : Initiate FTP?

Transport Layer : FTP uses TCP (3-handshake) → establish connected

↳ → SYN
← SYN-ACK
→ SYN

Network Layer : File transfer between client and host
FTP is plain-text transfer, but for more security use FTPS

Link Layer : Data or file packages sent through hardware to devices

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.

Given two lists $A = a_1, \ldots, a_n$ and $B = b_1, \ldots, b_m$
return the list of elements that are in A but not B
Time complexity: $O(n \log m + m \log m)$

What gives $\log n$ time?
Merge sort ← generic n
BST
↳ constructing a BST = $O(n \log n)$ time
searching through a BST = $O(\log n)$, but do that for however many arguments

Algorithm:
① Construct a Binary Search Tree for the elements in B.
BST: 1st B element at root
As you go through B list, if element is less than root (or another node you're comparing it to), go to left. If greater, go to right. Keep comparing until you reach null and then insert
$O(m \log m)$

② Iterate through every element in A.
Search the BST (which would only be half of it)
If element is not found, append it to a list [that you created] for elements that are not found in B.

$O(n \log m)$

8. **Automata and language theory**

   Consider the following grammar:

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

AM → Aε → aE → aε
→ bAAS → bAAA → baaa
AM → Aε → aE → aaB → aabε
→ abA → aba
→ abAA → abaaε
→ aaB → aaabbε

AM → Aε → aE → aaB
→ aaaBB
→ aaabb

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

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

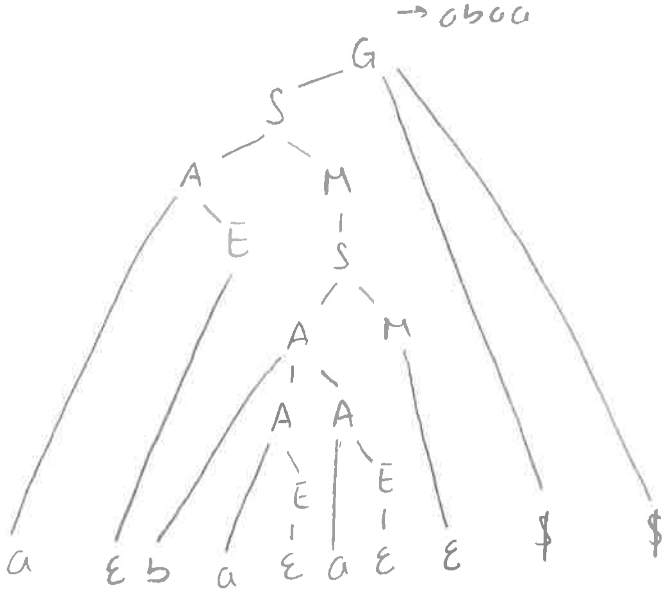
   (c) Is the grammar LL(1)? If so, show the parse table; if not, identify a prediction conflict.

a. Valid : {a, baaa, baa, aab, aba, abaa, aaabb, aaab}

At least one a and less than a bs

~~L = {a^i b^j a^k | i , j < i+k}~~

b. S → AM → AS
AS → AAM → aEbAA → abaεaε
→ abaa



c. The grammar is not context free.

Use the pumping lemma.
Split into 5 parts
$|uv^ixy^iz| \geq p$
where
$|vxy| < p$
$|vy| > 0$

If pumping lemma is true, there must be a pumping length p.

Let $w = a^p b^p a^p$
$u = a^p$
$v = b^\alpha \rightarrow b^{i\alpha}$
$x = b$
$y = b^{p-\alpha} \rightarrow b^{i(p-\alpha)}$
$z = a^p$

$uv^ixy^iz = a^p b^{i\alpha} b b^{i(p-\alpha)} a^p$
So
$i\alpha + 1 + i(p-\alpha) < p+p$
$\cancel{i\alpha} + 1 + ip - \cancel{i\alpha} < p+p$
$1 + ip < 2p$
$ip < 2p-1$
$i < \frac{2p-1}{p}$
if $i \geq \frac{2p-1}{p}$, w won't be in the language

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).

HAMPATH is NP. Decidable problems have a yes or no answer.

NP: decidable problems that can be solved with a nondeterministic polynomial machine.
↳ accept or reject
↳ always accepts if a path to accept exists
↳ problems are not solved in poly. time, but can be verified in poly. time

Prove problems are in P by giving a poly. algorithm

· Start at node a

repeat until there are no more unmarked nodes { Look at every node, if the node is not visited and an edge exists from (a, b). Go to node b.

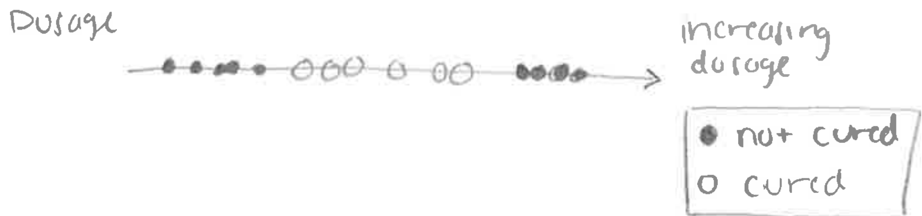
If the last edge connects to a, reject. Else, accept

10. **Other Topics**

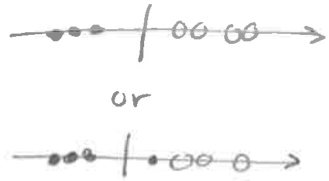
Give a detailed explanation of any one approach to machine learning. Give a substantial example that illustrates the technical operation of the approach, and demonstrates interesting knowledge learned.

SVM

Let's say that we are trying to find the amount of medication dosage that is able to cure patients of a certain condition.

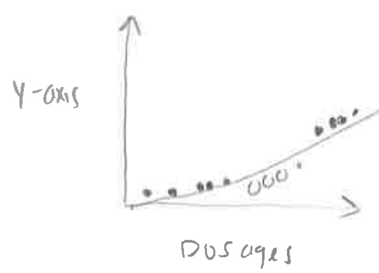


What an interesting distribution! Just by looking at the data, there is no clear cut threshold between cured and not cured patients. By clear cut I mean a distribution like this



SVC with cross-validation can allow for some misclassification

The data can be moved to a higher dimension as such by taking the square of every value



(Not drawn to scale :))

Now we have a distinction between cured and not cured dosages!

The SVC can be created by using a kernel where you specify the number of dimensions your data is in.

e.g. data is in 2D, so SVC becomes a hyperplane