Answer the questions in the boxes provided on the question sheets. If you run out of room for an answer, add a page to the end of the document.

Related Readings: `http://pages.cs.wisc.edu/~hasti/cs240/readings/`

Name: Zhuoyan Xu Wisc id: zxu444

Logic

1. Using a truth table, show the equivalence of the following statements.

(a) $P \vee (\neg P \wedge Q) \equiv P \vee Q$

P	Q	$\neg P$	$\neg P \wedge Q$	$P \vee (\neg P \wedge Q)$	$P \vee Q$
T	T	F	F	T	T
T	F	F	F	T	T
F	T	T	T	T	T
F	F	T	F	F	F

$\blacksquare$

(b) $\neg P \vee \neg Q \equiv \neg(P \wedge Q)$

P	Q	$\neg P$	$\neg Q$	$\neg P \vee \neg Q$	$P \wedge Q$	$\neg(P \wedge Q)$
T	T	F	F	F	T	T
T	F	F	T	T	F	F
F	T	T	F	T	F	F
F	F	T	T	T	F	F

(c) $\neg P \lor P \equiv$ true

P	¬P	¬P ∨ P
T	F	T
F	T	T

(d) $P \lor (Q \land R) \equiv (P \lor Q) \land (P \lor R)$

P	Q	R	P ∨ Q	P ∨ R	Q ∧ R	P ∨ (Q ∧ R)	(P ∨ Q) ∧ (P ∨ R)
T	T	T	T	T	T	T	T
T	T	F	T	T	F	T	T
T	F	T	T	T	F	T	T
T	F	F	T	T	F	T	T
F	T	T	T	T	T	T	T
F	T	F	T	F	F	F	F
F	F	T	F	T	F	F	F
F	F	F	F	F	F	F	F

Sets

2. Based on the definitions of the sets A and B, calculate the following: $|A|$, $|B|$, $A \cup B$, $A \cap B$, $A \setminus B$, $B \setminus A$.

   (a) $A = \{1, 2, 6, 10\}$ and $B = \{2, 4, 9, 10\}$

   $|A| = 4$ $|B| = 4$ $A \cup B = \{1, 2, 4, 6, 9, 10\}$
   $A \cap B = \{2, 10\}$
   $A \setminus B = A \cap B^c = \{1, 6\}$
   $B \setminus A = B \cap A^c = \{4, 9\}$

   (b) $A = \{x \mid x \in \mathbb{N}\}$ and $B = \{x \in \mathbb{N} \mid x \text{ is even}\}$

   $|A| = \infty$ $|B| = \infty$ $A \cup B = A$ $A \cap B = B$
   $A \setminus B = A \cap B^c = \{x \in N \mid x \text{ is odd}\}$
   $B \setminus A = B \cap A^c = \phi$

Relations and Functions

3. For each of the following relations, indicate if it is reflexive, antireflexive, symmetric, antisymmetric, or transitive.

   (a) $\{(x, y) : x \leq y\}$

   reflexive. antisymmetric. transitive

   (b) $\{(x, y) : x > y\}$

   antireflexive. antisymmetric. transitive

(c) $\{(x, y) : x < y\}$

antireflexive. antisymmetric. transitive

(d) $\{(x, y) : x = y\}$

reflexive symmetric. transitive

4. For each of the following functions (assume that they are all $f : \mathbb{Z} \to \mathbb{Z}$), indicate if it is surjective (onto), injective (one-to-one), or bijective.

(a) $f(x) = x$

bijective

(b) $f(x) = 2x - 3$

biqective

(c) $f(x) = x^2$

surjective

5. Show that $h(x) = g(f(x))$ is a bijection if $g(x)$ and $f(x)$ are bijections.

Consider $f: \Omega \to A$ $g: A \to B$

$\forall b \in B$. $\exists$ only one $a \in A$ s.t. $g(a) = b$ since g is bijection

For $a \in A$, $\exists$ only one $w \in \Omega$ s.t. $f(w) = a$ since f is bijection

we have $\forall b \in B$, we have unique $w \in \Omega$. s.t. $h(w) = b$.

$h(x)$ is bijection.

Induction

6. Prove the following by induction.

(a) $\sum_{i=1}^{n} i = n(n+1)/2$

(i) $n = 0$ $\quad 0 = 0$ $\quad \square$

(ii) Suppose $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

$$\sum_{i=1}^{n+1} i = \frac{n(n+1)}{2} + (n+1) = \frac{(n+1)(n+2)}{2}$$

then we have $\forall n$,

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

(b) $\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$

$2n^2+2n+1$

$2n \quad 3$

$n \quad 2$

(i) $n=0$ $\quad 0=0$

$2n \quad 1$

$n \quad 1$

(ii) Suppose $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$

$$\sum_{i=1}^{n+1} i^2 = \frac{1}{6} n(n+1)(2n+1) + (n+1)^2 = \frac{1}{6}(n+1)(2n^2+n+6n+6)$$

$$= \frac{1}{6}(n+1)(2n^2+7n+6) = \frac{1}{6}(n+1)$$

$$= \frac{1}{6}(n+1)(n+2)(2n+3)$$

(c) $\sum_{i=1}^{n} i^3 = n^2(n+1)^2/4$

(i) $n=0$ $\quad 0=0$

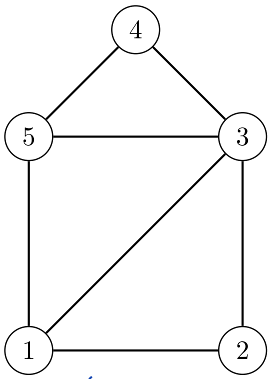
(ii) Suppose $\sum_{i=1}^{n} i^3 = \frac{1}{4} n^2(n+1)^2$

$$\sum_{i=1}^{n+1} i^2 = \frac{1}{4} n^2(n+1)^2 + (n+1)^3 = \frac{1}{4}(n+1)^2(n^2+4n+4)$$

$$= \frac{1}{4}(n+1)^2(n+2n)^2$$

Graphs and Trees

7. Give the adjacency matrix, adjacency list, edge list, and incidence matrix for the following graph.



adjacency matrix:

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

adjacency list.

1	→ 2	→ 3	→ 5	/
2	→ 1	→ 3	/	
3	→ 1	→ 2	→ 5	→ 4 /
4	→ 3	→ 5		
5	→ 1	→ 3	→ 4	/

edge list:

(1,2), (1,3), (1,5), (2,3), (3,4), (3,5), (4,5)

incidence matrix

	(1,2)	(1,3)	(1,5)	(2,3)	(3,4)	(3,5)	(4,5)
1	1	1	1				
2	1			1			
3		1		1	1	1	
4					1		1
5			1			1	1

8. How many edges are there is a complete graph of size n? Prove by induction.

$N = \frac{n(n-1)}{2}$ is the number of edges with size n complete graph

we prove by induction. $n=1$. $N=0$

consider for size n. we have $N = \frac{n(n-1)}{2}$

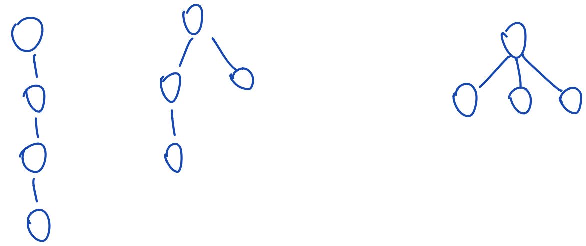
for size $n+1$, we add a vertex to size n graph.

since it's complete, we have:

$$N = \frac{n(n-1)}{2} + n = \frac{n(n+1)}{2} \text{ for size } n+1 \text{ complete graph}$$

then we prove $N = \frac{n(n-1)}{2}$ ∎

9. Draw all possible (unlabelled) trees with 4 nodes.



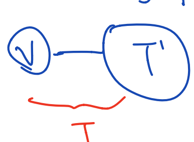
10. Show by induction that, for all trees, $|E| = |V| - 1$.

we prove by induction:

(i) $|V| = 1$, we have $|E| = 0$

(ii) Suppose: $|V| = n \geq 2$ $|E| = n-1$

consider $\forall$ tree T with $n+1$ vertices. then T has at least 2 leaves. we get a new graph T' by removing one of leaves v from T :



then we have T' is a connected graph and doesn't contain simple cycle since T' is a subgraph of tree T.

Then we have T' is a tree by definition.

Since T' has n vertices. we have $|V_{T'}| = n$ $|E_{T'}| = n-1$

then for T: $|V_T| = n+1$ $|E_T| = n$ $\Rightarrow$ $|V_T| = |E_T| + 1$ ∎

Counting

11. How many 3 digit pin codes are there?

$10 \times 10 \times 10 = 1000$

12. What is the expression for the sum of the ith line (indexing starts at 1) of the following:

1
2 3
4 5 6
7 8 9 10
⋮

the sum of 1-th line to i-th line:

$1+2+\cdots+\frac{i(i+1)}{2} = \frac{1}{2}\left[\frac{i(i+1)}{2}\cdot\left(\frac{i(i+1)}{2}+1\right)\right] = \frac{1}{8}\, i(i+1)\cdot\left[i(i+1)+2\right]$

the sum of i-th line:

$\frac{1}{8}\left\{i(i+1)\left[i(i+1)+2\right] - i(i-1)\cdot\left[i(i-1)+2\right]\right\}$

13 13 13 13

5

13. A standard deck of 52 cards has 4 suits, and each suit has card number 1 (ace) to 10, a jack, a queen, and a king. A standard poker hand has 5 cards. For the following, how many ways can the described had be drawn from a standard deck.

(a) A royal flush: all 5 cards have the same suit and are 10, jack, queen, king, ace.

5

(b) A straight flush: all 5 cards have the same suit and are in sequence, but not a royal flush.

$5 \times 9 = 45$

(c) A flush: all 5 cards have the same suit, but not a royal or straight flush.

$4\binom{13}{5} = 4\times\frac{13\times12\times11\times10\times9}{5\times4\times3\times2} = 13\times12\times11\times3 = 5148$

(d) Only one pair:

$13\binom{4}{2}\cdot\binom{12}{3}\times4 = \frac{13\times4\times3}{2}\times\frac{12\times11\times10}{3\times2}\times4$

$= 68640$

Proofs

14. Show that $2x$ is even for all $x \in \mathbb{N}$.
    (a) By direct proof.

since $\frac{2x}{2} = x \in N \subseteq Z$

then $2x$ is even

    (b) By contradiction.

If $2x$ is odd, then $\exists k \in Z$ s.t. $2x = 2k+1$

then we have $x = \frac{2x}{2} = \frac{2k+1}{2} = k + \frac{1}{2} \notin N$.

contradict with $x \in N$.

we have $2x$ is even

15. For all $x, y \in \mathbb{R}$, show that $|x + y| \leq |x| + |y|$. (Hint: use proof by cases.)

Case 1: $x+y \geq 0$

1.1: $x \geq 0, y \geq 0$. $x+y \leq x+y$

1.2: $x \geq 0, y < 0$. $x+y \leq x-y$

1.3. $x < 0, y \geq 0$ $x+y \leq -x+y$ $\square$

case 2: $x+y < 0$

2.1 $x \geq 0, y < 0$ $-x-y \leq x-y$

2.2. $x < 0, y \geq 0$ $-x-y \leq -x+y$

2.3. $x < 0, y < 0$ $-x-y \leq -x-y$ $\square$

$\blacksquare$

Program Correctness (and invariants)

16. For the following algorithms, describe the loop invariant(s) and prove that they are sound and complete.

(a)

Algorithm 1: findMin

Input: a: A non-empty array of integers (indexed starting at 1)
Output: The smallest element in the array

```
begin
    min ← ∞
    for i ← 1 to len(a) do
        if a[i] < min then
            min ← a[i]
        end
    end
    return min
end
```

Consider $a = (a_1, a_2, \cdots a_m)$ where m: length(a)

we first state loop invariant:

After k loop of (a), we have $min = \min(a_1, a_2, \cdots a_k)$

for every $k \in \{1, \cdots$ length(a)$\}$

we prove by induction: for $k=0$ it holds

consider $k=n$ holds. $min = \min\{a_1 \cdots a_n\}$ after n loops

for $k=n+1$. if $a_{n+1} < min$. $min \leftarrow a_{n+1}$

we have $min = \min\{a_1, \cdots a_{n+1}\}$ after $n+1$ loops

then we prove the partial correctness □ . ①

Since a is non-empty array of integers.

the loop will terminates when $i = n$

we have the termination of algorithm □ ②

then we prove correctness of algorithm □

Algorithm 2: InsertionSort

Input: a: A non-empty array of integers (indexed starting at 1)
Output: a sorted from largest to smallest

```
begin
    for i ← 2 to len(a) do
        val ← a[i]
        for j ← 1 to i − 1 do
            if val > a[j] then
                shift a[j..i − 1] to a[j + 1..i]
                a[j] ← val
                break
            end
        end
    end
    return a
end
```

(b)

a:
$i-1$
val
j

Consider $a = (a_1, \cdots a_m)$ m = length

we first state loop invariant:

After k main loops (loops on i). we have: first k entries in a

$a_{sub} = (a_1, \cdots a_k)$ ordered from largest to smallest.

we prove by induction.

when $k=2$. if $a_1 > a_2$, shift first 2 entries. else remain the same

we have (a_1, a_2) ordered after first loop

suppose $k=n$ $(a_1, \cdots a_n)$ ordered. for $k=n+1$. $val \leftarrow a_{n+1}$

for $j = 1 \ldots n$

① if $\exists\, l$ s.t. $val > a_l$. let $l := \underset{j}{\arg\min}\ val > a_j$. then $val > \max_{j \geq l} a_j$

insert val right before a_l will have a sorted first $n+1$ elements.

② $\forall j \in \{1, \ldots n\}$ $val \leq a_j$ the items remain the same.

we have sorted first $n+1$ elements

then we have partial correctness □

then we prove termination:

Since a is non-empty array of integers.

the loop will terminates when $i = n$

we have the termination of algorithm □ ②

then we prove correctness of algorithm $\boxed{\text{III}}$

Recurrences

17. Solve the following recurrences.

(a) $c_0 = 1; c_n = c_{n-1} + 4$

$$
\begin{aligned}
C_n &= C_{n-1} + 4 \\
&= C_{n-2} + 4 \times 2 \\
&= \cdots \\
&= C_0 + 4n \\
&= 4n + 1
\end{aligned}
$$

(b) $d_0 = 4; d_n = 3 \cdot d_{n-1}$

$$
\begin{aligned}
d_n &= 3 \cdot d_{n-1} \\
&= 3^2 d_{n-2} \\
&= \cdots \\
&= 3^n d_0 \\
&= 4 \times 3^n
\end{aligned}
$$

(c) $T(1) = 1; T(n) = 2T(n/2) + n$ (An upper bound is sufficient.)

$$
\begin{aligned}
T(n) &= 2T(\tfrac{n}{2}) + n \\
&= 2\left(2T(n/2^2) + \frac{n}{2}\right) + n \\
&= 2^2 T(n/2^2) + 2n \\
&= 2^2\left(2T(n/2^3) + \frac{n}{2^2}\right) + 2n \\
&= 2^3 T(n/2^3) + 3n \\
&= \cdots \\
&= 2^{\log_2 n} T(n/n) + \log_2 n \cdot n \\
&= n(\log_2 n + 1) = O(n \log n)
\end{aligned}
$$

$k = \log_2 n$

(d) $f(1) = 1; f(n) = \sum_1^{n-1} (i \cdot f(i))$
(If you are having trouble with this one. Look for a hint to be posted on Piazza)

$$
f(n) - f(n-1) = \sum_{i=1}^{n-1} i f(i) - \sum_{i=1}^{n-2} i f_i = (n-1) \cdot f(n-1)
$$

$$
\begin{aligned}
f(n) &= n f(n-1) \\
&= n \cdot (n-1) \cdot f(n-2) \\
&= n! \cdot f(1) \\
&= n!
\end{aligned}
$$

Coding Question

Most assignments will have a coding question. You can code in C, C++, C#, Java, or Python. You will submit a Makefile and a source code file.

Makefile: In the Makefile, there needs to be a build command and a run command. Below is a sample Makefile for a C++ program. You will find this Makefile in assignment details. Download the sample Makefile and edit it for your chosen programming language and code.

```
#Build commands to copy:
#Replace g++ -o HelloWorld HelloWord.cpp below with the appropriate command.
#Java:
#       javac source_file.java
#Python:
#       echo "Nothing to compile."
#C#:
#       mcs -out:exec_name source_file.cs
#C:
#       gcc -o exec_name source_file.c
#C++:
#       g++ -o exec_name source_file.cpp
build:
        g++ -o HelloWorld HelloWorld.cpp

#Run commands to copy:
#Replace ./HelloWorld below with the appropriate command.
#Java:
#       java source_file
#Python 3:
#       python3 source_file.py
#C#:
#       mono exec_name
#C/C++:
#       ./exec_name
run:
        ./HelloWorld
```

HelloWorld Program Details The input will start with a positive integer, giving the number of instances that follow. For each instance, there will be a string. For each string s, the program should output Hello, s! on its own line.

A sample input is the following:

```
3
World
Marc
Owen
```

The output for the sample input should be the following:

```
Hello, World!
Hello, Marc!
Hello, Owen!
```