CS 577 **Assignment 1 – Discrete Review** Spring 2024

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: Mahir Husain Khan

Wisc id: mhkhan9@wisc.edu
908 479 8843

Logic

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

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

Solution: Truth Table

P	Q	$P \vee Q$	$\neg P$	$\neg P \wedge Q$	$P \vee (\neg P \wedge Q)$
1	1	1	0	0	1
1	0	1	0	0	1
0	1	1	1	1	1
0	0	0	1	0	0

Thus $P \vee Q$ and $P \vee (\neg P \wedge Q)$

$P \vee Q$	$P \vee (\neg P \wedge Q)$
1	1
1	1
1	1
0	0

$\boxed{\therefore P \vee (\neg P \wedge Q) \equiv P \vee Q}$

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

Solution:

P	Q	$\neg P$	$\neg Q$	$\neg P \vee \neg Q$	$P \wedge Q$	$\neg(P \wedge Q)$
1	1	0	0	0	1	0
1	0	0	1	1	0	1
0	1	1	0	1	0	1
0	0	1	1	1	0	1

→ thus $\neg(P \wedge Q)$: 0, 1, 1, 1

and $\neg P \vee \neg Q$: 0, 1, 1, 1

$\boxed{\therefore \neg P \vee \neg Q \equiv \neg(P \wedge Q)}$

(c) $\neg P \vee P \equiv \text{true}$

Solution:

P	$\neg P$	true	$\neg P \vee P$
1	0	1	1
1	0	1	1
0	1	1	1
0	1	1	1

$\rightarrow$ since

$\neg P \vee P$	and	true
1		1
1		1
1		1
1		1

$\hookrightarrow$ $\neg P \vee P \equiv \text{true}$, aka

$\neg P \vee P$ is a tautology

(d) $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$

Solution:

P	Q	R	$Q \wedge R$	$P \vee (Q \wedge R)$	$P \vee Q$	$P \vee R$	$(P \vee Q) \wedge (P \vee R)$
1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	0	0	0	1	1	1	1
0	1	1	1	1	1	1	1
0	1	0	0	0	1	0	0
0	0	1	0	0	0	1	0
0	0	0	0	0	0	0	0

As seen in the truth table above,

$$P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$$

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 $B = \{2, 4, 9, 10\}$

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

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

   Solution: $|A| =$ infinity, $|B| =$ infinity, $A \cup B = \{x \mid x \in \mathbb{N}\}$
   $A \cap B = \{x \in \mathbb{N} \mid x \text{ is even}\}$ $A \setminus B = \{x \in \mathbb{N} \mid x \text{ is odd}\}$
   $B \setminus A = \emptyset$

Relations and Functions

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

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

   $x \le y$
   $y \le x$

   Solution: Reflexive, antisymmetric, transitive

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

   Solution: antireflexive, antisymmetric, transitive

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

Solution: antireflexive, antisymmetric, transitive

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

Solution: reflexive, symmetric, antisymmetric, 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) $f(x) = x$

Solution: bijective

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

Solution: injective, not every integer can be reached.

i.e: $2x - 3 = 8$

$2x = 5$

$x = 2.5 \to$ Not an integer!

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

Solution: $f(x) = x^2 \to x^2 = y$, $x = \sqrt{y} \to$ not an integer $\to$ Thus, $f(x)$ is not onto

also $(-2)^2 = 4$, $2^2 = 4$, thus $f(x)$ is not one to one

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

Solution: if $f(x)$ is a bijection, that means it is both one to one and onto. Furthermore, one to one means that distinct elements in the domain map to distinct elements in the codomain. onto means that every element in the codomain can be reached by at least one element in the domain. Thus, if $f(x)$ is a bijection, then both the statements above are true. The same applies to $g(x)$ meaning that $g(f(x))$ must be bijective since $f(x)$ is bijective.

Thus, since $h(x) = g(f(x))$, then $h(x)$ must also be bijective

Induction

6. Prove the following by induction.
   (a) $\sum_{i=1}^{n} i = n(n+1)/2$

Solution: $\sum_{i=1}^{n}$ → Base case: LHS = $i=1$ RHS = $n=1$ → $\frac{1(1+1)}{2} = \frac{2}{2} = 1$
↳ thus LHS = RHS → thus base case holds!

Inductive hypothesis: Assume $n=k$ holds, i.e.: $\sum_{i=1}^{k} = 1+2+\ldots+k = \frac{k(k+1)}{2} = \frac{k^2+k}{2}$

Inductive step: $\sum_{i=1}^{k+1} k+1 = \frac{(k+1)(k+2)}{2}$

$1+2+\ldots+k+k+1 = \frac{k^2+3k+2}{2} = \frac{k^2+k+2k+2}{2} = \frac{k^2+k}{2} + \frac{2k+2}{2}$

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

↳ From inductive hypothesis we know $\sum_{i=1}^{k} i = \frac{k(k+1)}{2}$ is true

$\therefore \sum_{i=1}^{k} k+1$ → thus same as LHS, thus $k+1$ holds

$\therefore \sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ holds by induction

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

Solution: $\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$. Base case: $i=1$ → LHS 1^2

RHS $\frac{1(1+1)(2\cdot1+1)}{6}$ $\frac{1(2)(3)}{6} = \frac{6}{6} = 1$ → thus LHS = RHS → thus base case holds

Inductive hypothesis: Assume true for $\sum_{i=1}^{k} 1^2+2^2+3^2\ldots+k^2 = \frac{k(k+1)(2k+1)}{6}$ → Assume this holds. $k(k+1)(2k+1) = (k^2+k)(2k+1) = \frac{2k^3+3k^2+k}{6}$

Inductive step: $\sum_{i=1}^{k+1} 1^2+2^2+3^2+\ldots k^2+(k+1)^2 \Rightarrow (k+1)(k+2)(2(k+1)+1)$ → $\frac{(k+1)(k+2)(2k+3)}{6}$

$= \frac{(k^2+3k+2)(2k+3)}{6} = \frac{2k^3+3k^2+6k^2+9k+4k+6}{6} = \frac{(2k^3+3k^2+k)+(6k^2+12k+6)}{6}$

$= \frac{(2k^3+3k^2+k)}{6} + \frac{(6k^2+12k+6)}{6} = \sum_{i=1}^{n} k^2 + \frac{(6k^2+12k+6)}{6} = \sum_{i=1}^{n} k^2 + (k^2+2k+1)$

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

$= \sum_{i=1}^{n} k^2 + (k+1)^2 = \sum_{i=1}^{n} (k+1)^2$ → Thus $\sum_{i=1}^{k+1} (k+1)^2 = (k+1)(k+2)(2k+3)$ holds by induction.

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

Solution: Base case: Assume $P(1)$ holds: $1^3 = \frac{1^2(1+1)^2}{4} = \frac{1\cdot4}{4} = \frac{4}{4} = 1$ RHS
Thus LHS = RHS, thus $P(1)$ holds ← LHS

Inductive hypothesis: Assume $1^3+2^3+\ldots+k^3 = \sum_{i=1}^{k} k^3 = \frac{k^2(k+1)^2}{4}$ holds (i.e: $P(k)$)

Inductive step: show $1^3+2^3+\ldots+k^3+(k+1)^3 = \frac{(k+1)^2(k+2)^2}{4}$ → $\frac{(k+2)(k+2)(k+1)^2}{4} = \frac{(k^2+4k+4)(k+1)^2}{4}$
$(\sum_{i=1}^{k+1} (k+1)^3)$

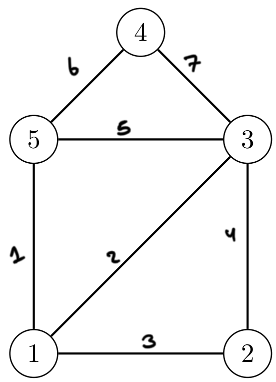
$= \frac{(k^2)(k+1)^2(4k+4)}{4} = \frac{k^2(k+1)^2}{4} + \frac{(4k+4)(k+1)^2}{4} = \sum_{i=1}^{n} k^3 + \frac{(4k+4)(k+1)^2}{4}$ → inductive hypothesis →

$= \sum_{i=1}^{n} k^3 + \frac{(4k+4)(k^2+2k+1)}{4} = \sum_{i=1}^{n} k^3 + \frac{4k^3+8k^2+4k+4k^2+8k+4}{4} = \frac{4k^3+12k^2+12k+4}{4} = k^3+3k^2+3k+1$

↳ $\sum_{i=1}^{n} k^3 + (k+1)^3$ → $\sum_{i=1}^{n} (k+1)^3$ → thus by induction we have proved that $\sum (k+1)^3 = \frac{(k+1)^2(k+2)^2}{4}$ holds!

Graphs and Trees

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



Solution:

adjacency matrix:

	1	2	3	4	5
1	0	1	1	0	1
2	1	0	1	0	0
3	1	1	0	1	1
4	0	0	1	0	1
5	1	0	1	1	0

Adjacency list:

1: 2, 3, 5
2: 1, 3
3: 1, 2, 4, 5
4: 5, 3
5: 1, 3, 4

Edge List:

$\{\{1,2\}, \{1,3\}, \{1,5\}, \{2,3\}, \{3,4\}, \{3,5\}, \{5,4\}\}$

Incidence Matrix:

edges / vertices

	1	2	3	4	5	6	7
1	1	1	1	0	0	0	0
2	0	0	1	1	0	0	0
3	0	1	0	1	1	0	1
4	0	0	0	0	0	1	1
5	1	0	0	0	1	1	0

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

Solution: The formula for calculating how many edges there are in a complete graph is: $\frac{n\cdot(n-1)}{2}$

Base Case 1: The number of edges in a complete graph with 1 vertex (K_1) is 0 edges.
Putting this into the equation: $\frac{1(1-1)}{2} = \frac{0}{2} = 0$ –> Holds for Base case 1

Base Case 2: The # of edges in a K_2 graph is 1 edge. Equation: $\frac{2\cdot(2-1)}{2} = \frac{2\cdot 1}{2} = 1$ edge
holds for base case 2

Inductive Hypothesis: Assume the formula calculates how many edges there are in a complete graph with k vertices, (i.e: K_k) correctly. $\frac{k\cdot(k-1)}{2} = \frac{k^2-k}{2}$

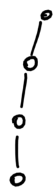
Inductive step: A complete graph with $k+1$ vertices, i.e: K_{k+1} –> we know from the inductive hypothesis that $\frac{k\cdot(k-1)}{2}$ correctly calculates how many edges there are in a K_k complete graph. Thus if we add one more vertex, then it must connect to the other k vertices. Plugging this into the inductive hypothesis, we get $\frac{k\cdot(k-1)}{2} + k = \frac{k\cdot(k-1)+2k}{2} = \frac{k^2-k+2k}{2}$

$= \frac{k^2+k}{2}$ –> this can be simplified into –> $\frac{(k)(k+1)}{2}$ –> this is the same as plugging $(k+1)$ into the formula $\left(\frac{(k+1)(k)}{2}\right)$

thus by induction, the formula $\frac{(n)(n-1)}{2}$ can find number of edges for complete graphs with n vertices

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

Solution:



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

Solution: Base Case: 1 vertex tree, which is the smallest possible tree. This tree has 0 edges since there are no vertices to connect to. The formula corroborates this, as $|V| - 1 = 1 - 1 = 0 = |E|$

Inductive Hypothesis: In a tree with k vertices, assume the formula $|E| = |V| - 1$ holds true, thus $|E| = k - 1$ is assumed to be true

Inductive Step: In a tree with k+1 vertices, how many edges would the tree have? Well we know from the inductive hypothesis that $|E| = k - 1$ holds true. If one more vertex is added, then only 1 more edge can be added. This is because the new vertex can only be connected to one other vertex, which is connected by only one edge. Thus, $|E| = k - 1 + 1 = |E| = k$

Putting k+1 into the formula (as $|V| = k+1$) $\rightarrow |E| = k+1 - 1$ $\boxed{= |E| = k}$

Thus as shown by induction, we can see that the formula $|E| = |V| - 1$ holds for all trees.

Counting

11. How many 3 digit pin codes are there?

Solution: $10 \cdot 10 \cdot 10 = 1000$ pin codes

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
⋮

Solution:
1 = 1
2 3 = 5
4 5 6 = 15
7 8 9 10 = 34
11 12 13 14 15 = 65

11 13
12 14
23 + 27 = 50
15
65

1, 5, 15, 34, 65

4 10 19 31

6 9 12

3 3 → difference of 3 each time

start term = 1

starts at 6 : 6 = 3n
2 = n

6 + 3n

???

couldnt get it to work ← through geometric series

Tried finding the difference and calulating the expression

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 hand be drawn from a standard deck.

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

Solution: only 4 possible royal flushes

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

1 2 3 4 5 6 7 8 9 10 J Q K

Solution:
1 − 5 5 − 9 9 − K
2 − 6 6 − 10
3 − 7 7 − J
4 − 8 8 − Q

→ (9 possible straight flushes) × 4 = 36 ways

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

Solution: $\binom{13}{5} \times 4 = 5148$ ways

$5148 - 36 - 4 = 5108$ ways

Royal flush = 4 ways

straight flush = 36 ways

(d) Only one pair (2 of the 5 cards have the same number/rank, while the remaining 3 cards all have different numbers/ranks):

Solution: $\underbrace{\binom{52}{1}+\binom{3}{1}}_{\text{Same number}}+\binom{50}{3}$

$= 52 + 3 + 19600 = 19655$

Proofs

14. Show that $2x$ is even for all $x \in \mathbb{N}$.

(a) By direct proof.

Solution: An even number is a number that can be expressed in the form "2K" for some integer K.
Now considering $2x$, where $x \in \mathbb{N}$, we can express $2x$ as $2 \cdot K$ where $K = x$. This is because since x is a natural number, it is also an integer by definition.
Thus by definition, $2x$ is even for all $x \in \mathbb{N}$, proven by direct proof.

(b) By contradiction.

Solution: Assume $2x$ is not even for all $x \in \mathbb{N}$. By definition, an even number can be expressed as $2k$ for some $K \in \mathbb{Z}$
Thus, if $2x$ is not even, it cannot be expressed in the form $2K$, and we know if a number is not even then it must be odd, expressed in the form $2K+1$, $K \in \mathbb{Z}$.
Therefore if $2x$ is not even (thus odd), it must be able to be expressed in the form $2K+1$
Thus $\rightarrow$ $2x = 2K+1$
$x = K + \frac{1}{2}$ $\rightarrow$ However, x is a natural number, thus it cannot equal $K + \frac{1}{2}$ since $\frac{1}{2}$ is not a natural number.
Thus, $2x$ must be an even number, as proven by contradiction.

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

Solution: We will split this proof into 3 cases.
Case 1: $x \geq 0, y \geq 0$
$|x+y| = x+y$, $|x|+|y| = x+y$, thus $\boxed{x+y = x+y}$
Case 2: $x \geq 0, y < 0$:
$y = -y$, $x = x$
$\hookrightarrow$ sub-case 1: $x - y \geq 0$ $\rightarrow$ $|x+y| < |x| + |y|$, thus $\boxed{x+y \leq x+y}$ since $y \leq 0$
$\hookrightarrow$ sub-case 2: $x - y < 0$ $\rightarrow$ $-(x+y) \leq x+y$, thus $\boxed{-(x+y) < x+y}$ $\rightarrow$ because $-x-y < x-y$
Case 3: $x < 0, y \geq 0$:
same proof as case 2
Case 4: $x < 0, y < 0$:
$x = -x$, $y = -y$
$\hookrightarrow$ $-x-y \leq |-x| + (-y)$
$\boxed{-x-y < x+y}$ $\Rightarrow$ Thus for all cases, $|x+y| \leq |x| + |y|$ holds!

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
```

Solution: The loop invariant for this algorithm is that at the start of each iteration of the loop, the variable "min" contains the smallest value in the array "a" from index 1 to $i-1$

will prove the soundness and completeness of this invariant through induction.

Base Case:
Before the first iteration of the loop, the "min" variable is set to ∞, meaning it is larger than all possible values in array "a". Thus, the invariant holds.

Inductive hypothesis: Assume Invariant holds on the K^{th} iteration of the loop, meaning the variable "min" contains the smallest value in the array from index 1 to $K-1$

Inductive Step: In the $(K+1)^{st}$ iteration of the loop, we need to show the invariant holds for "min" from $a[1]$ to $a[K]$.

There are 2 cases to consider:

1) If $a[K] \geq min$: In this case, "min" is left unchanged, so by the inductive hypothesis, "min" still contains the smallest value in the array from index 1 to K.

2) If $a[K] < min$: In this case, min is updated to $a[K]$. Since $a[K]$ is less than any other element in array a from index 1 to K-1 (by IH), min now contains the smallest value and thus the invariant holds!

In both cases the invariant holds, showing soundness

When the loop terminates, $i = len(a) + 1$. By the loop invariant, "min" contains the smallest value in the array from index 1 to len(a), which is the entire array. Therefore, the algorithm correctly returns the smallest element in the array.

↳ Hence invariant shows completeness!

(b)

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
```

Solution: The loop invariant in this problem is that the start of each outer loop iteration contains the elements of a in descending order.

Base Case: The first iteration of the loop at i=2 puts the element at a[i] at a[j], where j=1, therefore the array is sorted in this iteration.

Inductive Hypothesis: Assume the loop invariant holds for some i=k and j=m.

Inductive step: At the start of the k+1st iteration, a[k+1] is compared to a[m+1] (where j=m+1). If a[k+1] > a[m+1], the elements from a[1] to a[m+1] are shifted to the right until a[1...m] > a[k+1] showing descending order.
If a[m+1] > a[k+1], the current order holds since it is already in descending order.
In both cases, the element in a[k+1] is placed in the correct area according to the descending order, Thus the invariant is shown to be sound

The outer loop terminates when i = len(a) + 1, which should be at the end of the array, and thus the array will have been successfully sorted into descending order.

Thus this invariant is both sound and complete.

Recurrences

17. Solve the following recurrences. Show work and do not use the master theorem.

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

Solution:

$C_0 = 1$ 4
$C_1 = 5$ 4
$C_2 = 9$ 4
$C_3 = 13$ 4
$C_4 = 17$ 4
$C_5 = 21$

$\hookrightarrow 4n + 1 \longrightarrow$ Proposed solution

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

Solution:

$d_0 = 4$
$d_1 = 3 \cdot 4 = 12$
$d_2 = 3 \cdot 12 = 36$
$d_3 = 3 \cdot 36 = 108$
$d_4 = 3 \cdot 108 = 324$
$d_5 = 3 \cdot 324 = 972$

$8 \rightarrow d_0 + 2d_0 = 12$
$\times 3$
$24 \rightarrow d_1 + 2d_1 = 12 + 24 = 36$
$\times 3$
$72 \rightarrow d_2 + 2d_2 = 36 + 72 = 108$
$\times 3$
216
$\times 3$
748

$a_n = 4 \cdot 3^n$

$a_1 = 4 \cdot 3^0$
$a_2 = 4 \cdot 3^1 = 12$
$a_3 = 4 \cdot 3^2 = 36$
$a_4 = 4 \cdot 27$

Proposed solution : $4 \cdot 3^n$

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

Solution: Recursion Tree

Level 0 $T(n) = 2T(n/2) + n$ → The height of the tree is $\log_2 n$ because we divide by 2 on each level. So there are $\log_2 n + 1$ levels in total.

Level 1 : $2T(n/2) = 4T(n/4) + n$

Level 2 : $4T(n/4) = 8T(n/8) + n$

$\vdots$

At each level i, the cost is n and there are 2^i nodes so the total cost at level i is $2^i \cdot n$

Total cost of all levels = sum of cost at each level : $T(n) = \sum_{i=0}^{\log_2 n} 2^i \cdot n$

$= n \sum^{\log_2 n} 2^i$

→ geometric series → sum = $\frac{r^k - 1}{r-1} = \frac{2^{\log_2 n+1} - 1}{2-1} \cdot n$

$= n \cdot (2n-1) = 2n^2 - n$ → $\therefore$ $T(n) = 2n^2 - n$

(d) $f(1) = 1; f(n) = \sum_1^{n-1} (i \cdot f(i))$
(Hint: compute $f(n+1) - f(n)$ for $n > 1$)

Solution:

$f(n+1) = \sum_{i=1}^{n} i \cdot f(i)$ $\quad$ $f(n) = \sum_{i=1}^{n-1} i \cdot f(i)$

$f(n+1) - f(n) = \sum_{i=1}^{n} i \cdot f(i) - \sum i \cdot f(i)$

$\therefore f(n+1) - f(n) = n \cdot f(n)$

$f(n+1) = n+1 \cdot f(n)$

$f(1) = 1$

$f(2) = 2 \cdot 1 = 2$

$f(3) = 3 \cdot 2 = 6$

$f(4) = 4 \cdot 6 = 24$

$f(5) = 5 \cdot 24 = 120$

→ $f(n) = n! \; (\forall n \geq 1)$

Coding Question: Hello World

Most assignments will have a coding question. You can code in C, C++, C#, Java, Python, or Rust. 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
#Rust:
#       rustc source_file.rs

build:
        g++ -o HelloWorld HelloWord.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
#Rust:
#       ./source_file

run:
        ./HelloWorld
```

18. **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!
```