Discrete Structures
CSC160

Instructor: Prakash Neupane
Godawari College
Itahari

Basic Discrete Structures

- Chapter outline:
  - Sets
  - Functions
  - Sequence and Summations

Basic Discrete Structures

- Sets:
  - Sets and Subsets,
  - Power Set,
  - Cartesian Product,
  - Set Operations,
  - Venn Diagram,
  - Inclusion-Exclusion Principle,
  - Computer Representation of Sets

Basic Discrete Structures

- Sets:
  - Fundamental discrete structure on which all other discrete structure are built.
  - Used to groups discrete object together with similar properties
  - For instance:
    - **All the students who are currently enrolled in CSC160 make up a set.**
  - *Definition*:
    - **A set is an unordered collection of objects.**

Basic Discrete Structures

- Sets:
  - For instance:
    - If we are dealing with relations in database then they are sets; ordered collection of elements,
    - Similarly, we can view graph as a set.
  - ***Intuitively,* set is a collection of zero or more objects (or elements or members), the elements need not be ordered.**
  - If we denote set by **S** and some element from the set by e then we say “**e** belongs to **S**” or “**S** contains **e**”

or

in symbol we can write **e** $\in$ **S.**

- Sets:
  - Note that the term ***object*** *has been used without specifying* ***what an object is.***
  - ***Definition:***
    - The objects in s set are called the elements, or members, of the set. A set is said to contain its elements.
  - *For instance,*
    - *$V = \{a, e, i, o, u\}$ is a set of vowels and $\boldsymbol{i} \in \boldsymbol{V}$, if some object doesn't belong to the set we write it as " does not belong to" i.e. say $\boldsymbol{x} \notin \boldsymbol{V}$;*
    - *C = {b, c, d, f, g, h, j, k, l, m, n, p, q, r, s, t, v, w, y, z) is a set of consonants.*

- Subset:
  - Let A and B be sets.
  - We say that A is a subset of B if and only if every element of A is an element of A.
  - We write $A \subseteq B$ to denote the fact that A is a subset of B.
  - For instance:
    - If $A=\{3,5,8\}$ and $B=\{5,8,3,2,6\}$, then $A \subseteq B$
    - If $S=\{3,5,8\}$ and $T=\{5,3,8\}$ then $S \subseteq T$ and $T \subseteq S$

Basic Discrete Structures

- Representations of Sets:
  - Listing of elements:
    - Set of vowels is {a, e, i, o, u}.
  - Set builder form:
    - Here properties of the members of the set is described for
    - e.g. R = {x | x is a real number}
  - Recursive formula:
    - The elements of set are defined using the previous element of the set that is known.
    - e.g. set of natural numbers can be represented as N = $\{x_n = x_{n-1} + 1$, where $x_0 = 0\}$.

- Super Set:
  - The superset is represented by using the symbol "$\supset$".
  - For example, the set A is the superset of set B, then it is symbolically it is represented as $A \supset B$.
  - For example,
    - X = {set of polygons}
    - Y = {set of irregular polygons}
  - Then X is the superset of Y ($X \supset Y$).
  - In other words, we can say that Y is a subset of X and represented as ($Y \subset X$).

- Equal Sets:
  - Two sets A and B are equal if and only if they contain exactly same
  - elements.
  - In other words if $A \subseteq B$ and $B \subseteq A$ then $A = B$.
  - For e.g.
    - $A = \{1, 2, 3, 4, 5\}$ and
    - $B = \{2, 5, 4, 1, 3\}$ are equal sets.

- Empty set:
  - The set that contains no element is called empty set and denoted by $\varnothing$.
  - It is also called null set.
  - We have $\varnothing = \{\ \}$ but $\varnothing \neq \{\varnothing\}$.

- Power Set:
  - Given a set S, power set denoted by P(S) is the set that contains all the subsets of the set S.
  - Symbolically we can write $P(S) = \{x \mid x \subseteq S\}$.
  - For e.g. power set for the set {2, 3} is {∅, {2}, {3}, {2,3}}.
  - The number of elements in the power set of set having n elements is $2^{|n|}$
  - Remember: ∅ is always member of all power set.

- Cardinality:
  - For the set **S**, if there are exactly **n** distinct elements in **S,** where **n** is a number then we say that cardinality of the set **S** is **n** denoted by **|S|.**
  - For e.g.
    - $|\varnothing| = 0$;
    - $|\{a, b, b, c, a\}| = 3$;
    - $|\{\{a, b\}, \{a, b, c\}\}| = 3$
  - If $n \in N$ then the set is finite otherwise, it is infinite.

Basic Discrete Structures

- Order pair:
  - An ordered pair is a pair of elements where the order of the elements matters.
  - It is typically denoted as (a, b), where a is the first element and b is the second element.
  - An ordered pair is not considered a single element of a set in set theory.
  - An ordered pair set is a set that contains multiple ordered pairs, such as {(a, b), (c, d), ...}.
  - The order of the elements within an ordered pair is significant; (a, b) is distinct from (b, a).
  - Ordered pair sets are useful for constructing mathematical structures like Cartesian products or functions.

Basic Discrete Structures

- Cartesian Product:
  - The Cartesian product of two sets A and B, denoted as $A \times B$, is a set of all possible ordered pairs (a, b), where a belongs to set A and b belongs to set B. In other words, it is a combination of all elements from both sets.
  - Mathematically, the Cartesian product of A and B can be defined as:
    - $A \times B = \{(a, b) \mid a \in A \text{ and } b \in B\}$
  - Here's an example to illustrate the Cartesian product:
    - Let $A = \{1, 2\}$ and $B = \{a, b\}$.
    - $A \times B = \{(1, a), (1, b), (2, a), (2, b)\}$
  - In this example, the Cartesian product of sets A and B generates a new set with four ordered pairs: (1, a), (1, b), (2, a), and (2, b).
  - Each ordered pair represents a combination of an element from set A and an element from set B.
  - The Cartesian product is a fundamental concept in set theory and has applications in various areas, such as mathematics, computer science, and database systems.

- Venn Diagram:
  - A visual representation of the relationships between different sets.
  - Venn diagrams are widely used in various fields, including mathematics, logic, statistics, and data analysis.
  - They are helpful for organizing and analyzing data, solving problems involving set operations, and demonstrating logical relationships.

- Set Operations:
  - Set operations are fundamental operations that can be performed on sets to manipulate and combine their elements.
  - The basic set operations include,
    - Union,
    - Intersection,
    - Difference, and
    - Complement.

- Set Operations:
  - Union (∪):
    - The union of two sets A and B, denoted as $A \cup B$, is a set that contains all elements that are present in either set A or set B or in both.
    - Example:
      - $A = \{1, 2, 3\}$
      - $B = \{3, 4, 5\}$
      - $A \cup B = \{1, 2, 3, 4, 5\}$

- Set Operations:
  - Intersection (∩):
    - The intersection of two sets A and B, denoted as $A \cap B$, is a set that contains all elements that are common to both set A and set B.
    - Example:
      - $A = \{1, 2, 3\}$
      - $B = \{3, 4, 5\}$
      - $A \cap B = \{3\}$

- Set Operations:
  - Difference (set minus) (−):
    - The difference of two sets A and B, denoted as A - B, is a set that contains all elements that are present in set A but not in set B.
    - Example:
      - A = {1, 2, 3}
      - B = {3, 4, 5}
      - A - B = {1, 2}

- Set Operations:
  - Complement ('):
  - The complement of a set A, denoted as A', is a set that contains all elements that are not present in set A, within the universal set U.
    - Example:
      - U = {1, 2, 3, 4, 5}
      - A = {3}
      - A' = {1, 2, 4, 5}

- Set Identities :
  - Set identities are fundamental relationships or properties that hold true for sets and their operations. These identities help simplify set expressions and establish connections between different set operations. Here are some common set identities:
  - Identity Law:
    - $A \cup \varnothing = A$ (Union with the empty set is the set itself)
    - $A \cap U = A$ (Intersection with the universal set is the set itself)
  - Domination Law:
    - $A \cup U = U$ (Union with the universal set is the universal set itself)
    - $A \cap \varnothing = \varnothing$ (Intersection with the empty set is the empty set itself)
  - Idempotent Law:
    - $A \cup A = A$ (Union of a set with itself is the set itself)
    - $A \cap A = A$ (Intersection of a set with itself is the set itself)

- Set Identities :
  - Commutative Law:
    - $A \cup B = B \cup A$ (Union is commutative)
    - $A \cap B = B \cap A$ (Intersection is commutative)
  - Associative Law:
    - $(A \cup B) \cup C = A \cup (B \cup C)$ (Associativity of union)
    - $(A \cap B) \cap C = A \cap (B \cap C)$ (Associativity of intersection)
  - Distributive Law:
    - $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ (Union distributes over intersection)
    - $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ (Intersection distributes over union)
  - De Morgan's Laws:
    - $(A \cup B)' = A' \cap B'$ (Complement of the union)
    - $(A \cap B)' = A' \cup B'$ (Complement of the intersection)

- Let's verify De Morgan's Laws using an example. Consider two sets: A = {1, 2, 3} and B = {2, 3, 4}. We will verify De Morgan's Laws for these sets.
  - Complement of the Union (First De Morgan's Law):
  - $(A \cup B)' = A' \cap B'$
  - Using the sets A and B, let's calculate the left-hand side (LHS) and the right-hand side (RHS) of the equation and check if they are equal.
    - LHS: $(A \cup B)' = (\{1, 2, 3\} \cup \{2, 3, 4\})'$
    - $= \{1, 2, 3, 4\}'$ (Taking the union of A and B)
    - $= \{1, 2, 3, 4\}'$ (The set itself)
    - $= \{\}$ (Empty set)

    - RHS: $A' \cap B' = \{1\}' \cap \{2, 3, 4\}'$
    - $= \varnothing \cap \{2, 3, 4\}$ (Complement of A and B)
    - $= \{\}$ (Empty set)
  - Since the LHS and RHS both result in the empty set {}, they are equal. Thus, the first De Morgan's Law holds true for the given example.

- Let's verify De Morgan's Laws using an example. Consider two sets: A = {1, 2, 3} and B = {2, 3, 4}. We will verify De Morgan's Laws for these sets.
  - Complement of the Intersection (Second De Morgan's Law):
  - $(A \cap B)' = A' \cup B'$
    - Again, let's calculate the LHS and RHS of the equation and check their equality.
  - LHS: $(A \cap B)' = (\{1, 2, 3\} \cap \{2, 3, 4\})'$
    - $= \{2, 3\}'$ (Taking the intersection of A and B)
    - $= \{1, 4\}$ (Complement of the intersection)
  - RHS: $A' \cup B' = \{1\}' \cup \{2, 3, 4\}'$
    - $= \varnothing \cup \{1, 2, 3, 4\}$ (Complement of A and B)
    - $= \{1, 2, 3, 4\}$ (Universal set)
  - In this case, the LHS ({1, 4}) and the RHS ({1, 2, 3, 4}) are not equal, indicating that the second De Morgan's Law does not hold true for the given example.

Basic Discrete Structures

- Inclusion-Exclusion Principle:
  - The Inclusion-Exclusion Principle is a counting technique used in combinatorics and set theory to calculate the size or cardinality of the union of multiple sets.
  - It provides a formula for calculating the number of elements in the union of sets while accounting for overlaps and intersections.
  - The principle states that for any finite sets $A_1, A_2, \ldots, A_n$, the size of their union ($A_1 \square A_2 \square \ldots \square A_n$) can be calculated as follows:
    - $\mathbf{|A_1 \cup A_2 \cup \ldots \cup A_n| = |A_1| + |A_2| + \ldots + |A_n| - |A_1 \cap A_2| - |A_1 \cap A_3| - \ldots - |A_{n-1} \cap A_n| + |A_1 \cap A_2 \cap A_3| + \ldots + (-1)^\wedge(n-1)\ |A_1 \cap A_2 \cap \ldots \cap A_n|}$
  - In this formula, $|A|$ represents the cardinality or the number of elements in set A, and $\cap$ denotes the intersection of sets.

- Inclusion-Exclusion Principle:
  - The Inclusion-Exclusion Principle is useful for solving problems involving counting, probability, and set operations, where we need to determine the size of unions of sets while considering their overlaps.
  - By applying this principle, we can accurately calculate the size of the union of multiple sets, accounting for all possible intersections and overlaps among them

- Inclusion-Exclusion Principle:
  - Problem: In a programming competition, there are 50 participants. Out of these participants, 30 know Java, 25 know Python, and 20 know C++. Additionally, 15 participants know both Java and Python, 10 participants know both Python and C++, and 5 participants know all three languages. How many participants know at least one of the three languages?

- Inclusion-Exclusion Principle:
  - Solution:
    - Let's use the Inclusion-Exclusion Principle to solve this problem.
      - Let, A = Participants who know Java
      - B = Participants who know Python
      - C = Participants who know C++
    - We are asked to find the number of participants who know at least one of the three languages, which is equivalent to finding the size of the union (A ∪ B ∪ C).
    - Using the information given:
      - |A| = 30 (participants who know Java)
      - |B| = 25 (participants who know Python)
      - |C| = 20 (participants who know C++)

- Inclusion-Exclusion Principle:
  - Solution:
    - $|A \cap B| = 15$ (participants who know both Java and Python)
    - $|B \cap C| = 10$ (participants who know both Python and C++)
    - $|A \cap C| = ?$ (participants who know both Java and C++)

    - We can find $|A \cap C|$ using the principle of inclusion-exclusion. Since 5 participants know all three languages, we subtract this from the total intersection:
      - $|A \cap C| = |A| + |C| - |A \cap C \cap B| = 30 + 20 - 5 = 45 - 5 = 40.$
    - Now, we can calculate the size of the union:
      - $|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |A \cap C| + |A \cap B \cap C|$
      - $= 30 + 25 + 20 - 15 - 10 - 40 + 5$
      - $= 35$
    - Therefore, there are 35 participants who know at least one of the three languages (Java, Python, or C++).

- Computer Representation of Sets:
  - In computer science, sets can be represented using bits, particularly using bit vectors or bit arrays.
  - Each element in the set is associated with a bit position in the bit vector, and the corresponding bit is set to 1 if the element is present in the set or 0 if it is not.

Basic Discrete Structures

- Computer Representation of Sets: To illustrate the computer representation of sets using bits:
  - Let's consider a set of integers from 1 to 8: {1, 2, 3, 4, 5, 6, 7, 8}.
  - We will use a bit vector of size 8, where each bit represents the presence or absence of an element in the set.
    - Initialize the bit vector as: **00000000**
  - To represent the set, we set the corresponding bit positions to 1 for the elements present in the set:
    - If we include elements 1, 3, 4, and 8 in the set, the bit vector will be updated as: **10110001**
    - If we include elements 2, 5, 6, and 7 in the set, the bit vector will be updated as: **11111111**

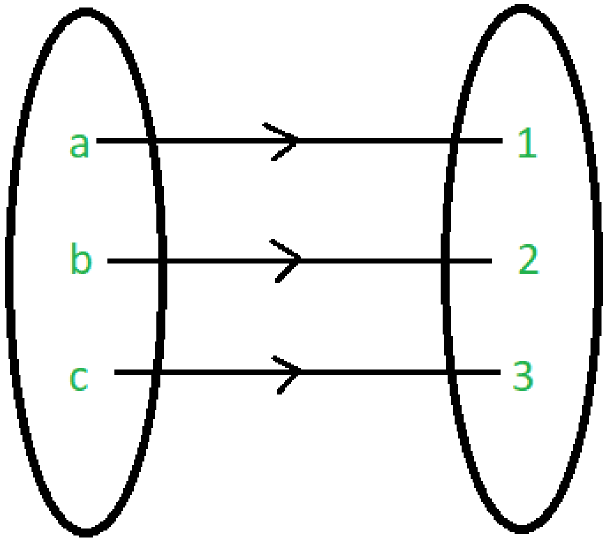
- Functions:
  - A function $f : A \rightarrow B$ consists of:
    - the set A, which is called the domain of f ,
    - the set B, which is called the range / codomain of f ,
    - a rule which assigns to every element $a \in A$ an element $b \in B$, which we denote by f (a).
    - If A is function from A to B we write $f: A \rightarrow B$
  - Codomain: Set of values that could possibly come out
  - Range: Set of values that actually do come out

- Functions:
  - Pre-image and Image of a function::
    - ***A function f: A ⇢ B such that for each a ∈ A, there exists a unique b ∈ B such that (a, b) ∈ R then, a is called the pre-image of f and b is called the image of f.***
    - For example, let's consider a function f: A ⇢ B, where A is the set of natural numbers (A = {1, 2, 3, ...}) and B is the set of even numbers (B = {2, 4, 6, ...}). We can define the relation R such that (a, b) ∈ R if and only if a is an odd number and b is the next even number greater than a.
    - With this definition, for each odd number a in A, there exists a unique even number b in B such that (a, b) ∈ R. The pre-image a is the odd number, and the image b is the next even number. For example, f(3) = 4, f(7) = 8, and so on.

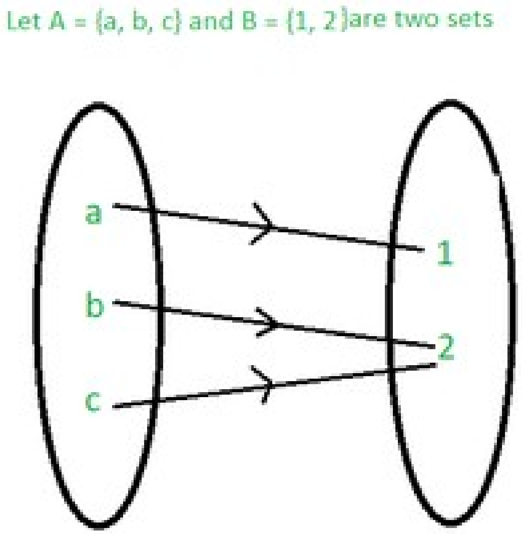
Basic Discrete Structures

- Functions:
  - One to One function (Injective Function):
    - one element of the domain is connected to one element of the codomain.
    - f: A ⇢ B is said to be a one-one (injective) function if different elements of A have different images in B.
  -

Let A = {a, b, c} and B = {1, 2, 3}are two sets

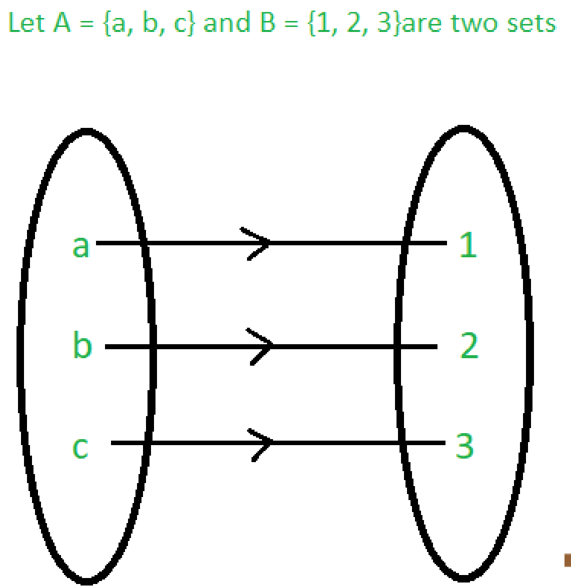


- Functions:
  - Onto function(Surjective Function):
    - A function f: A -> B is said to be onto (surjective) function if every element of B is an image of some element of A i.e. f(A) = B or range of f is the codomain of f.
    - A function in which every element of the codomain has one pre-image.
  -



Basic Discrete Structures

- Functions:
  - One-One Correspondent function(Bijective Function or One-One Onto Function):
    - A function which is both one-one and onto (both injective and surjective) is called one-one correspondent(bijective) function.
  -

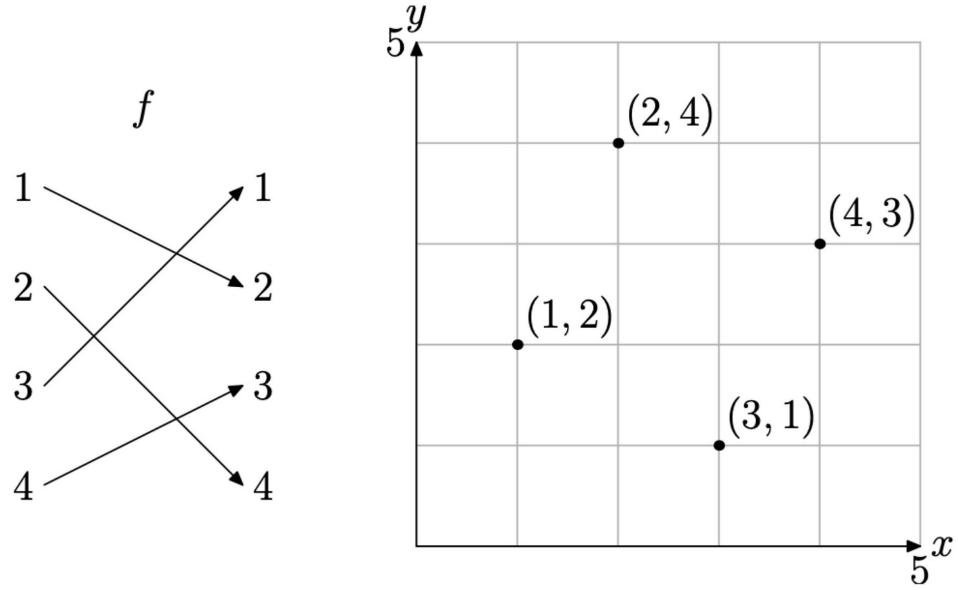


- Functions:
  - Inverse of a function
    - Let f: A ⇢ B be a bijection then, a function g: B ⇢ A which associates each element b ∈ B to a different element a ∈ A such that f(a) = b is called the inverse of f.
    - $f(a) = b \leftrightarrow g(b) = a$
  -

- Functions:
  - Composition of functions :
    - Let f: A ⇢ B and g: B ⇢ C be two functions then, a function gof: A ⇢ C is defined by
      - $(gof)(x) = g(f(x))$, for all $x \in A$ is called the composition of f and g.
  -

Basic Discrete Structures

- Functions:
  - Graph of a functions :
    - The graph of a function is the collection of all ordered pairs of the function.
    - These are usually represented as points in a Cartesian coordinate system.
  - As an example, consider the function: f={(1,2),(2,4),(3,1),(4,3)}
    - Fig (a): Mapping diagram
    - Fig (b): Graph of a function



(a) (b)

- Functions:
  - Graph of a functions :
    - **The graph of f is the set of all ordered pairs (x,f(x)) so that x is in the domain of f. In symbols,**
      - **Graph of f={(x,f(x)):x is in the domain of f.}**

- Functions for CS:
  - **Ceiling and Floor Function:**
    - The ceiling and floor functions are mathematical functions that allow you to round a number up or down to the nearest integer, respectively.

    - Notation: $\lceil x \rceil$ is ceiling function, $\lfloor x \rfloor$ is floor function where x is variable.
    - Let x be a real number then:
      - $\lfloor x \rfloor$ called the floor function of x, assigns to the real number x the largest integer that is less than or equal to x.
      - This function rounds x down to the closest integer less than or equal to x.
      - The floor function is also called the greatest integer function.

      - $\lceil x \rceil$ called the ceiling function of x, assigns to the real number x the smallest integer that is greater than or equal to x.
      - This function rounds x up to the closest integer less than or equal to x.

- Functions for CS:
  - **Ceiling and Floor Function:**

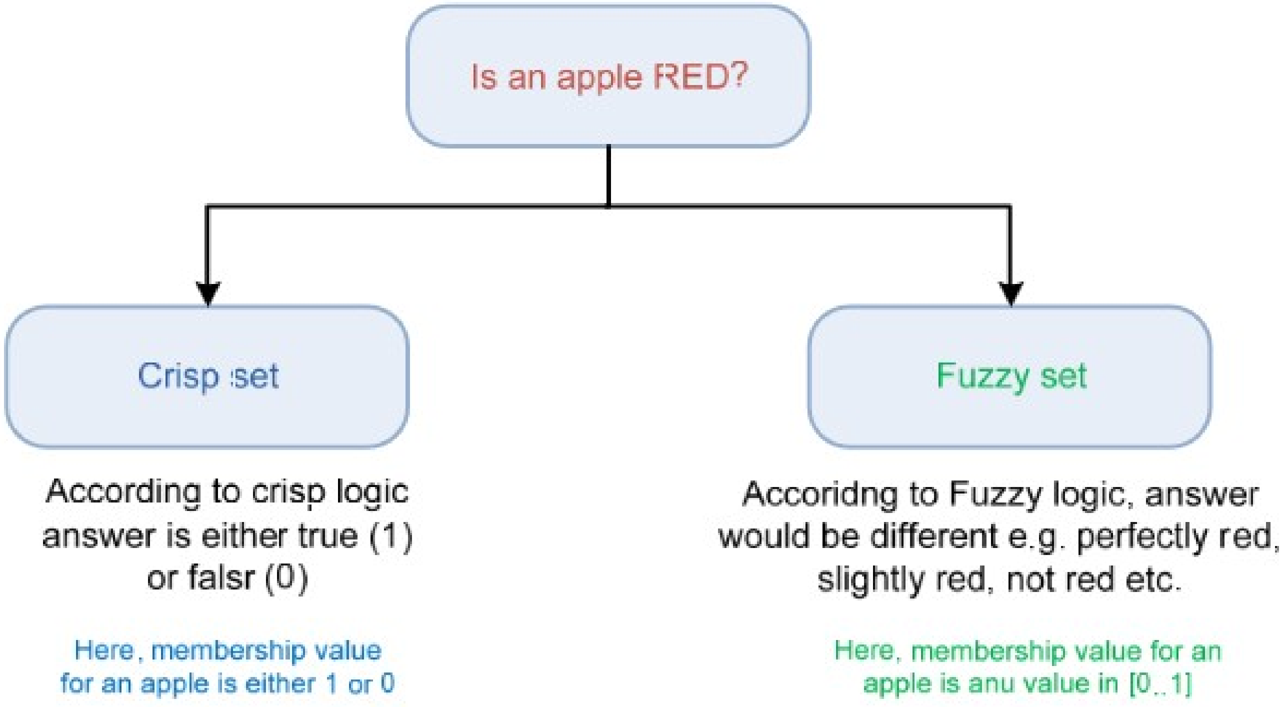
```
#include <stdio.h>
#include <math.h>
 int main()
{
   float val;
    float fVal,cVal;

printf("Enter a float value: ");
scanf("%f",&val);

fVal=floor(val);
cVal =ceil(val);
printf("floor value:%f \nceil value:%f\n",fVal,cVal);
    return 0;
}
```

- Functions:
  - **Boolean and Exponential Function:**
    - A boolean function operates on boolean variables, true (1) or false (0).
    - Boolean functions take boolean inputs and produce boolean outputs.
    - They are commonly used in logic circuits, computer programming, and digital systems. Boolean functions can be expressed using logical operators such as AND, OR, NOT, and XOR.
    - These functions are defined by truth tables that specify the output value for each combination of input values.
    - Boolean functions can be combined using logical operators to create more complex functions.

- Functions:
  - **Boolean and Exponential Function:**
    - Exponential functions are a type of mathematical function that involves a base raised to the power of a variable.
    - In discrete structures, exponential functions often refer to functions of the form f(x) = a^x, where 'a' is a constant and 'x' is a discrete variable taking integer values.
    - Exponential functions grow or decay rapidly as the variable 'x' increases or decreases.
    - These functions have many applications in fields such as mathematics, physics, finance, and computer science.
    - Exponential functions in discrete structures are particularly relevant in the study of algorithms and complexity theory.
    - Exponential growth or decay rates can have significant implications for the efficiency or complexity of algorithms.
    - For example, an algorithm with an exponential time complexity will generally become impractical for large input sizes.

Basic Discrete Structures

- Fuzzy Sets:
  - **Classical Sets vs Fuzzy Sets:**



- Fuzzy Sets:
  - **Classical Sets vs Fuzzy Sets:**
    - The generalization is performed as follows:
    - For any crisp set A, it is possible to define a Characteristic function or membership function $\mu A = \{0, 1\}$.i.e. the characteristic function takes either of the values 0 Or 1 in the classical set.
    - For a fuzzy set, the characteristic function can take any value between zero and one.

Basic Discrete Structures

- Fuzzy Sets:
  - Fuzzy Set Theory was formalized by Professor Lofti Zadeh at the University of California in 1965.
  - Definitions:
    - Formally, let X be a universal set, and a fuzzy set A in X is characterized by a membership function μA(x), which assigns a membership grade to each element x in X.
    - The membership function represents the degree to which x belongs to the fuzzy set A
  - The membership function μA(x) takes values in the interval [0, 1], where 0 indicates no membership and 1 indicates full membership. Intermediate values represent degrees of membership between 0 and 1, reflecting the extent to which an element partially belongs to the fuzzy set.

- Fuzzy Sets:
  - Membership function:
    - Suppose we have a fuzzy set A representing the concept of "tallness" in a group of people.
    - The universal set, X, in this case, would be the set of all individuals in the group.
    - The membership function, μA(x), assigns a membership grade to each person indicating their degree of tallness.
    - Here's an example of a membership function for the fuzzy set A:
    - μA(x) =
      - 0.1 if x is very short
      - 0.4 if x is moderately short
      - 0.7 if x is average height
      - 0.9 if x is moderately tall
      - 0.6 if x is very tall

- Fuzzy Sets:
  - Membership function:
    - In the above example, the membership function assigns different membership grades to individuals based on their height.
    - The grades range from 0 to 1, indicating the degree of tallness for each person.
    - For instance, if we consider a person named ABC, who is moderately tall, his membership grade in the fuzzy set A would be 0.9, indicating a high degree of tallness.
    - On the other hand, if we consider a person named XYZ, who is moderately short, her membership grade in the fuzzy set A would be 0.4, indicating a moderate degree of tallness.
    - It's important to note that these membership grades are subjective and can vary depending on the context and the criteria used to assess tallness.

- Fuzzy Set operations:
  - Fuzzy set operations allow for the combination and manipulation of fuzzy sets. The most commonly used fuzzy set operations are:
    - Union,
    - Intersection, and
    - Complement
  - Union:
    - The union of two fuzzy sets A and B, denoted as A ∪ B, represents the set of elements that belong to either A or B or both.
    - The membership grade in the union is determined by taking the maximum membership grade at each point.
      - **$\mu(A \cup B)(x) = \max(\mu A(x), \mu B(x))$**

- Fuzzy Set operations:
  - Union: Example
    - Consider two fuzzy sets A and B representing "tallness" and "athleticism," respectively, in a group of individuals.
    - Let's define their membership functions:
      - μA(x) =
        - 0.2 if x is moderately tall
        - 0.8 if x is very tall
      - μB(x) =
        - 0.5 if x is moderately athletic
        - 0.9 if x is highly athletic
    - To calculate the union, A ∪ B, we take the maximum membership grade at each point:
      - $\mu(A \cup B)(x) = \max(\mu A(x), \mu B(x))$
    - For example, for a person who is both moderately tall and moderately athletic, we evaluate: $\mu(A \cup B)(\text{Person}) = \max(0.2, 0.5) = 0.5$

Basic Discrete Structures

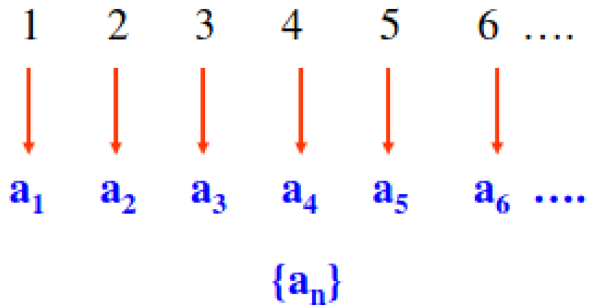
- Fuzzy Set operations:
- Intersection:
  - The intersection of two fuzzy sets A and B, denoted as A ∩ B, represents the set of elements that belong to both A and B.
  - The membership grade in the intersection is determined by taking the minimum membership grade at each point.
    - **$\mu(A \cap B)(x) = \min(\mu A(x), \mu B(x))$**
  - Example:
    - Using the same fuzzy sets A and B from the previous example, we calculate the intersection, A ∩ B:
    - $\mu(A \cap B)(x) = \min(\mu A(x), \mu B(x))$
  - For example, for a person, who is moderately tall and moderately athletic, we evaluate:
    - $\mu(A \cap B)(\text{Person}) = \min(0.2, 0.5) = 0.2$

- Fuzzy Set operations:
  - Complement:
    - The complement of a fuzzy set A, denoted as A', represents the set of elements that do not belong to A.
    - The membership grade in the complement is determined by subtracting the membership grade from 1.
      - **$\mu A'(x) = 1 - \mu A(x)$**

Basic Discrete Structures

- Fuzzy Set operations:
  - Complement:
    - Example: Let's consider the fuzzy set A representing "shortness" in a group of individuals:
      - μA(x) =
        - 0.3 if x is moderately short
        - 0.7 if x is very short
      - To calculate the complement, A', we subtract the membership grade from 1:
        - μA'(x) = 1 - μA(x)
    - For example, for a person who is moderately short, we evaluate:
        - μA'(Person) = 1 - 0.3 = 0.7

- Sequences and summations:
  - Sequences:
    - A sequence is a function from a subset of the set of integers (typically the set {0,1,2,...} or the set {1,2,3,...} to a set S.
    - We use the notation a_n to denote the image of the integer n.
    - We call a_n term of the sequence.
    - **Notation**:
    - $\{a_n\}$ is used to represent the sequence (note {} is the same notation used for sets, so be careful).
    - $\{a_n\}$ represents the ordered list a_1 , a_2 , a_3 , ...



Basic Discrete Structures

- Sequences and summations:
  - Sequences: Examples
    - $a^n = n^2$, where n = 1,2,3… What are the elements of the sequence?
      - ………………………..
    - $a^n = (-1)^n$, where n=0,1,2,3,… Elements of the sequence?
      - ……………………..
    - $a^n = 2^n$, where n=0,1,2,3,… Elements of the sequence?
      - ………………………..

Basic Discrete Structures

- Sequences and summations:
  - Sequences: Examples
    - $a^n = n^2$, where n = 1,2,3… What are the elements of the sequence?
      - 1, 4, 9, 16, 25, ...
    - $a^n = (-1)^n$, where n=0,1,2,3,… Elements of the sequence?
      - 1, -1, 1, -1, 1, ...
    - $a^n = 2^n$, where n=0,1,2,3,… Elements of the sequence?
      - 1, 2, 4, 8, 16, 32, ..

Basic Discrete Structures

- Sequences and summations:
  - Sequences: Arithmetic progression
    - An arithmetic progression is a sequence of the form
      - **a, a+d,a+2d, …, a+nd**
    - where **a** is the **initial term** and **d** is **common difference**, such that both belong to **R**.
  - Example:
    - $s_n = -1+4n$ for n=0,1,2,3, …
    - members: -1, 3, 7, 11, …

- Sequences and summations:
  - Sequences: Geometric progression
    - A geometric progression is a sequence of the form:
      - **$a, ar, ar^2, \ldots, ar^k,$**
    - where **a** is the **initial term**, and **r** is the **common ratio** and both **a** and **r** belong to **R.**
  - Example**:**
    - $a_n = (\frac{1}{2})^n$ for $n = 0,1,2,3, \ldots$
    - Members: 1, ½, ¼, 1/8, ……..

- Sequences and summations:
  - Summations:
    - Summation of the terms of a sequence:

$$\sum_{j=m}^{n} a_j = a_m + a_{m+1} + \ldots + a_n$$

    - The variable j is referred to as the index of summation.
      - m is the lower limit and
      - n is the upper limit of the summation.

Basic Discrete Structures

- Sequences and summations:
  - Summations: Examples

- **1) Sum the first 7 terms of {n²} where n=1,2,3, … .**
- $$\sum_{j=1}^{7} a_j = \sum_{j=1}^{7} j^2 = 1 + 4 + 16 + 25 + 36 + 49 = 140$$

- **2) What is the value of**

$$\sum_{k=4}^{8} a_j = \sum_{k=4}^{8} (-1)^j = 1 + (-1) + 1 + (-1) + 1 = 1$$

Basic Discrete Structures

- Sequences and summations:
  - Single and Double Summation

$$S = \sum_{j=1}^{5}(2+j3) =$$
$$= \sum_{j=1}^{5}2 + \sum_{j=1}^{5}j3 =$$
$$= 2\sum_{j=1}^{5}1 + 3\sum_{j=1}^{5}j =$$
$$= 2*5 + 3\sum_{j=1}^{5}j =$$
$$= 10 + 3\frac{(5+1)}{2}*5 =$$
$$= 10 + 45 = 55$$

$$S = \sum_{i=1}^{4}\sum_{j=1}^{2}(2i-j) =$$
$$= \sum_{i=1}^{4}\left[\sum_{j=1}^{2}2i - \sum_{j=1}^{2}j\right] =$$
$$= \sum_{i=1}^{4}\left[2i\sum_{j=1}^{2}1 - \sum_{j=1}^{2}j\right] =$$
$$= \sum_{i=1}^{4}\left[2i*2 - \sum_{j=1}^{2}j\right] =$$
$$= \sum_{i=1}^{4}[2i*2-3] =$$
$$= \sum_{i=1}^{4}4i - \sum_{i=1}^{4}3 =$$
$$= 4\sum_{i=1}^{4}i - 3\sum_{i=1}^{4}1 = \quad 4*10 - 3*4 = 28$$

Basic Discrete Structures

- Theory Assignment:

1. Define disjunction and conjunction with suitable examples. asked in 2068

2. Consider a set U = { 1,2,3,4,5,6,7,8,9,10}. What will be the computer representation for set containing the numbers which are multiple of 3 not exceeding 6? Describe injective, surjective and bijective function with examples. asked in 2075

4. Prove that $\overline{A \cap B} = \overline{A} \cup \overline{B}$ by using set builder notation. How sets are represented by using bit string? Why it is preferred over unordered representation of sets? asked in Model Question

5. How can you relate domain and co-domain of functions with functions in programming language? Discuss composite and inverse of function with suitable examples. asked in Model Question

6. Which of the following are posets? asked in 2076

a. $(Z, =)$
b. $(Z, \neq)$
c. $(Z, \subseteq)$

12. Define ceiling and floor function. Why do we need Inclusion - Exclusion principle? Make it clear withsuitable example. asked in 2076

- References:
  - *Kenneth H. Rosen, Discrete mathematics and its applications, Seventh Edition McGraw Hill Publication, 2012.*