A relation can be thought of as a table that lists the relationship of elements to other elements.

The most common relation in CS is binary relation, which relates two elements at one time.

Mathematically, a binary relation $R$ from a set $X$ to set $Y$ is a subset of the Cartesian product $X \times Y$. We say $(x, y) \in R$, which could be written as $xRy$ (meaning that $x$ is related to $y$). We say $X$ as the domain of the relation $R$, while $Y$ as the range of the relation $R$.

If $X = Y$, we will call $R$ as a binary relation on $X$.

**Definition 1:** A relation $R$ on $X$ is reflexive, if and only if $(x, x) \in R$ for every $x \in X$.

**Definition 2:** A relation $R$ on $X$ is symmetric, if and only if for each $(x, y) \in R$, then $(y, x) \in R$.

**Definition 3:** A relation $R$ on $X$ is transitive, if and only if for each $(x, y) \in R$ and $(y, z) \in R$, then $(x, z) \in R$.

We could also define anti-reflexive, anti-symmetric and anti-transitive when NO element in the relation satisfying the corresponding property. Please make sure that you understand that anti is different from not.

The most important relation is equivalence relation, when a relation is reflexive, symmetric and transitive. Based on an equivalence relation, a set may be partitioned.

Relations are partial orders if they are reflexive, anti-symmetric and transitive. For a partial ordered relation $R$ over $D$, two elements $x, y \in D$, they are either comparable or incomparable. If every pair of elements are comparable, then the partial order becomes total order.

**Inverse** of a relation is defined as:

$$R^{-1} = \{(y, x) | (x, y) \in R\}$$

Suppose $R_1$ is a relation from $X$ to $Y$ and $R_2$ is a relation from $Y$ to $Z$, the the composition of $R_1$ and $R_2$ is defined as:
How could we represent a relation in a program?

One naive solution is by using matrices. If we give an order for the elements in both the domain $X$ and range $Y$. We could have have a boolean matrix $c$ for the relation $R$, such that $x[i,j] = true$ if and only if $(x_i, y_j) \in R$.

Given a matrix for a relation, how could we determine:

1. The relation is reflexive;
2. The relation is symmetric;
3. The relation is transitive;
4. The relation is a equivalence relation;
5. The relation is a partial order;

How could we compute the composite of two relations?

A function is a special type of relation. A relation $f$ from $X$ to $Y$ is a function if and only if:

1. The domain of $f$ is $X$;
2. If $(x, y), (x, y') \in f$, then $y = y'$.

Disjoint Set (Partition)

We know that an equivalence relation $R$ defines a partition of a set $X$. In fact, a partition gives us a set of subset $X$, and each pair of the subsets are disjoint. Based on this idea, we define disjoint set with the following operations.

1. makeSet(int n): make a disjoint set, which contains $n$ subsets with one element;
2. union(x,y): union the set that $x$ belongs and the set $y$ belongs;
3. find(x): find the representative of element $x$;