

The background image shows the iconic Flatirons mountain range in Boulder, Colorado. The mountains are composed of light-colored, layered rock and are partially covered with green pine forests. In the foreground, there is a grassy, open field with a few small trees and shrubs. A group of people can be seen walking along a path on the left side of the field.

CSCI 1102 Computer Science 2

Meeting 15: Tuesday 3/23/2021
Order & Equality



John von Neumann

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 v}{\partial y^2} = \lambda$$
$$-u^2) \frac{\partial u}{\partial x} - u v \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)$$
$$+ (v^2 - u^2) \frac{\partial v}{\partial y} + \frac{\partial^2 v}{\partial y^2}$$

THE
SPA THESE EQUATIONS

SPACE

COMPONENTS

COMP

THE

PRI

PRESS

DEI

DENSIT

Y

Y

Y

Y

Y

Y

Y

Y

Y

Y

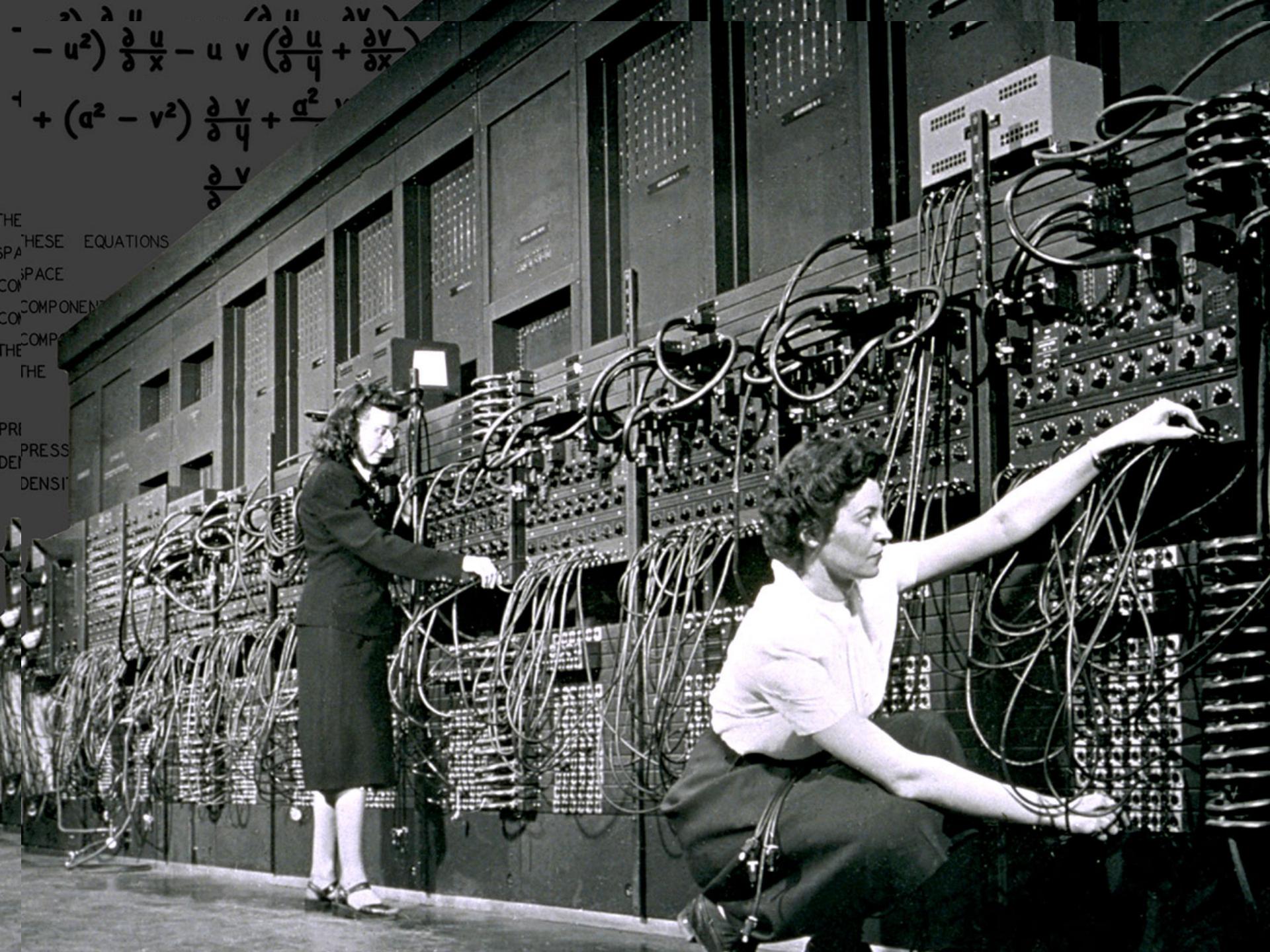
Y

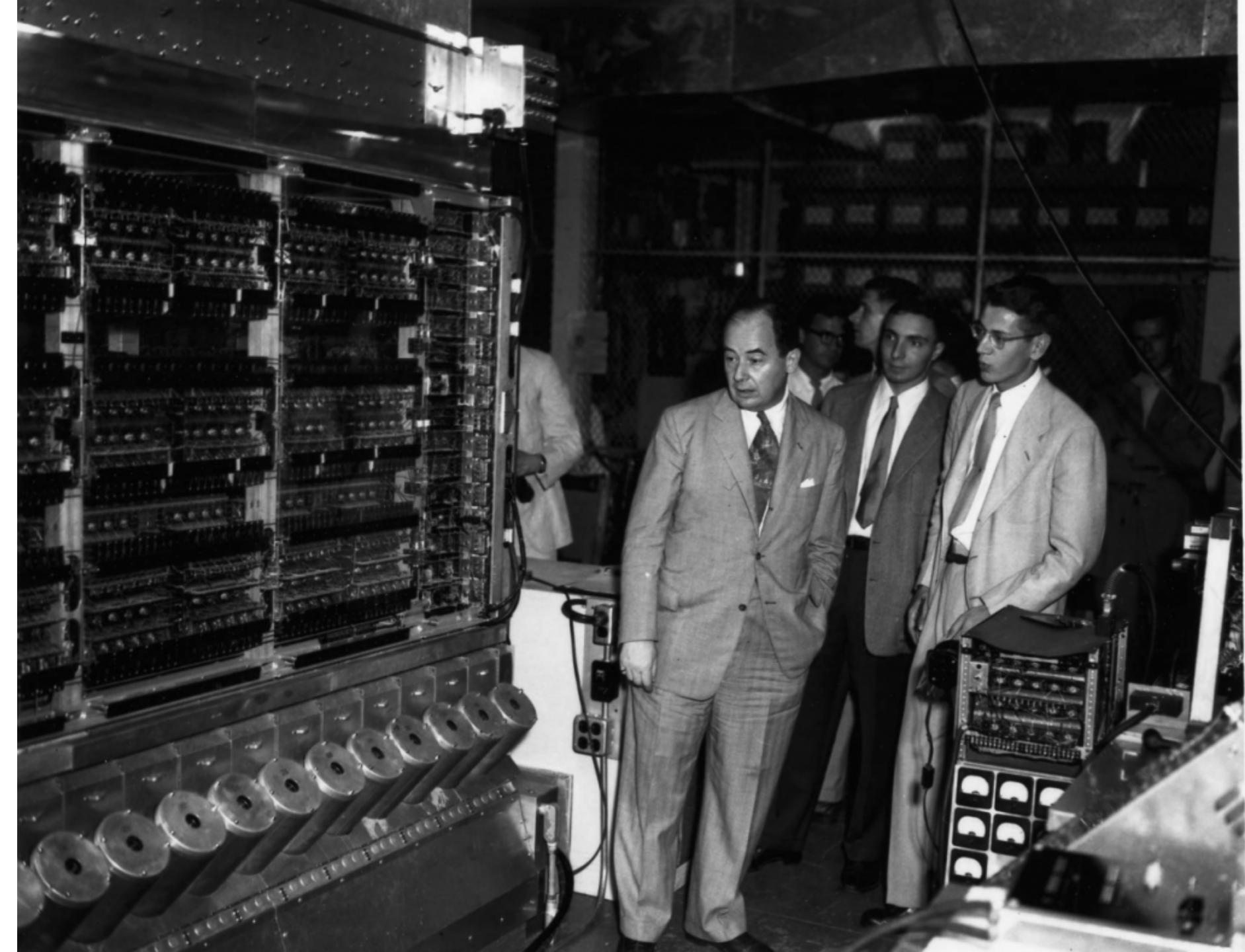
Y

Y

Y

Y





Von Neumann w Richard Feynman & Stanislaw Ulam in Los Alamos



Order & Equality

The Natural Order

mutation *new Comparator()* *null*
hashCode *<=* *Compare*
 == *compareTo*
equals

Sets



Sets

$$A = \{\clubsuit, \diamondsuit, \heartsuit, \spadesuit\}$$

$$B = \{\otimes, \oplus\}$$

Statements about Containment

$$\clubsuit \in \{\clubsuit, \diamondsuit, \heartsuit, \spadesuit\}$$

$$\otimes \notin \{\clubsuit, \diamondsuit, \heartsuit, \spadesuit\}$$

General Statements

Variables and their Quantifiers

x, y, z for variables (which *vary* over sets)

$\forall x \in A . \textit{statement}$

$\exists x \in A . \textit{statement}$

$\forall x \in \{\clubsuit, \diamondsuit, \heartsuit, \spadesuit\}.$ x is a card suit.

\clubsuit is a card suit *and*

\diamondsuit is a card suit *and*

\heartsuit is a card suit *and*

\spadesuit is a card suit.

$\forall x \in \{\otimes, \oplus\}$. x is a card suit.

\otimes is a card suit *and*

\oplus is a card suit.

Statements can be true or false.

$\exists x \in \{\otimes, \oplus\}$. x is a card suit.

\otimes is a card suit *or*

\oplus is a card suit.

Statements can be true or false.

Set Comprehensions

- $\{ x \mid \text{statement} \}$ means set of all x such that statement is true;

Example

$$\mathbb{N} = \{ k \mid k = 0 \text{ or } \exists k' \in \mathbb{N} \text{ such that } k = k' + 1 \}$$

$$\text{Evens} = \{ x \mid x \in \mathbb{N} \text{ and } \exists y \in \mathbb{N} \text{ such that } x = 2y \}$$

Tuples

(x_1, \dots, x_k) k -tuples.

Subset & Set Equality

$A \subseteq B$ means $\forall x \in A. x \in B;$

$A = B$ means $A \subseteq B$ and $B \subseteq A.$

Building Sets

- *Union* : $A_1 \cup \dots \cup A_n = \{a \mid a \in A_i \text{ for some } i \in \{1, \dots, n\}\}$;
- *Intersection* : $A_1 \cap \dots \cap A_n = \{a \mid a \in A_i \text{ for every } i \in \{1, \dots, n\}\}$;
- *Disjoint Union* : $A_1 + \dots + A_n = \{(i, a) \mid a \in A_i \text{ for some } i \in \{1, \dots, n\}\}$;
- *Product* : $A_1 \times \dots \times A_n = \{(a_1, \dots, a_n) \mid a_i \in A_i\}$; 
- *Sequences* : Let A be a set of symbols and let ϵ denote the empty sequence.

$$A^* = \{w \mid w = \epsilon \text{ or } w = aw' \text{ with } a \in A \text{ and } w' \in A^*\}$$

Example: $\{a, b\}^* = \{\epsilon, a\epsilon, b\epsilon, aa\epsilon, ab\epsilon, \dots\}$

Relations

- $R \subseteq A_1 \times \dots \times A_n$ is called an n-ary *relation* on sets A_1, \dots, A_n .
- When R is an n-ary relation on sets A_1, \dots, A_n and $A_1 = \dots = A_n$ we say that R is an n-ary relation on A_1 ;
- We're especially interested in binary relations where $n = 2$.
- $\text{DomDef}(R) = \{a \in A \mid \text{for some } b \in B, (a, b) \in R\}$

$$A = \{\clubsuit, \diamondsuit, \heartsuit, \spadesuit\} \quad B = \{\otimes, \oplus\}$$

$$\begin{aligned} R_1 = A \times B = & \{ \\ & (\clubsuit, \otimes), (\diamondsuit, \otimes), (\heartsuit, \otimes), (\spadesuit, \otimes), \\ & (\clubsuit, \oplus), (\diamondsuit, \oplus), (\heartsuit, \oplus), (\spadesuit, \oplus) \\ & \} \end{aligned}$$

$$R_2 = \{\}.$$

$$R_3 = \{(\clubsuit, \otimes), (\spadesuit, \oplus)\}$$

$$R_4 = \{(\clubsuit, \otimes), (\diamondsuit, \otimes), (\heartsuit, \otimes), (\spadesuit, \otimes)\}$$

Partial Maps (aka Partial Functions)

Let R be a binary relation on A, B .

R is a *partial map* from A to B if and only iff

$\forall a \in A. b, b' \in B. \text{if } (a, b) \in R \text{ and } (a, b') \in R \text{ then } b = b'$.

$$R_1 = A \times B = \{$$

$$(\clubsuit, \otimes), (\diamondsuit, \otimes), (\heartsuit, \otimes), (\spadesuit, \otimes),$$

$$(\clubsuit, \oplus), (\diamondsuit, \oplus), (\heartsuit, \oplus), (\spadesuit, \oplus)$$

$$\}$$
$$R_2 = \{\}.$$

Partial maps
from A to B?

$$R_3 = \{(\clubsuit, \otimes), (\spadesuit, \oplus)\}$$
$$R_4 = \{(\clubsuit, \otimes), (\diamondsuit, \otimes), (\heartsuit, \otimes), (\spadesuit, \otimes)\}$$

$$R_1 = A \times B = \{$$
$$(\clubsuit, \otimes), (\diamondsuit, \otimes), (\heartsuit, \otimes), (\spadesuit, \otimes),$$
$$(\clubsuit, \oplus), (\diamondsuit, \oplus), (\heartsuit, \oplus), (\spadesuit, \oplus)$$
$$\}$$
$$R_2 = \{\}.$$

Only R_2 , R_3 and R_4 are partial maps from A to B.

$$R_3 = \{(\clubsuit, \otimes), (\spadesuit, \oplus)\}$$
$$R_4 = \{(\clubsuit, \otimes), (\diamondsuit, \otimes), (\heartsuit, \otimes), (\spadesuit, \otimes)\}$$

Notation:

- We usually use f, g, h, \dots for partial maps;
- We use Euler's notation $f(a)$ to denote the unique $b \in B$ such that $(a, b) \in f$ or the special "undefined" symbol \perp if there is no such b .

Total Map

Let f be a partial map from A to B . Then f is a *total map* from A to B iff $\text{DomDef}(f) = A$. In the examples above, only R4 is a total map from A to B .

Preorders

Let R be a binary relation on A . I.e., $R \subseteq A \times A$.

- R is *reflexive* iff $\forall x \in A. (x, x) \in R$;
- R is *transitive* iff
 $\forall x, y, z \in A$. If $(x, y) \in R$ and $(y, z) \in R$ then $(x, z) \in R$.

A binary relation that is both reflexive and transitive is called a *preorder*.

Preorders

$\{(\clubsuit, \clubsuit), (\diamondsuit, \diamondsuit), (\heartsuit, \heartsuit), (\spadesuit, \spadesuit)\}$

$\{(\clubsuit, \clubsuit), (\diamondsuit, \diamondsuit), (\heartsuit, \heartsuit), (\spadesuit, \spadesuit), (\clubsuit, \spadesuit)\}$

$\{(\clubsuit, \clubsuit), (\diamondsuit, \diamondsuit), (\heartsuit, \heartsuit), (\spadesuit, \spadesuit), (\clubsuit, \spadesuit), (\spadesuit, \heartsuit)\}$

Preorders

$\{(\clubsuit, \clubsuit), (\diamondsuit, \diamondsuit), (\heartsuit, \heartsuit), (\spadesuit, \spadesuit)\}$ preorder

$\{(\clubsuit, \clubsuit), (\diamondsuit, \diamondsuit), (\heartsuit, \heartsuit), (\spadesuit, \spadesuit), (\clubsuit, \spadesuit)\}$ preorder

$\{(\clubsuit, \clubsuit), (\diamondsuit, \diamondsuit), (\heartsuit, \heartsuit), (\spadesuit, \spadesuit), (\clubsuit, \spadesuit), (\spadesuit, \heartsuit)\}$ not a preorder

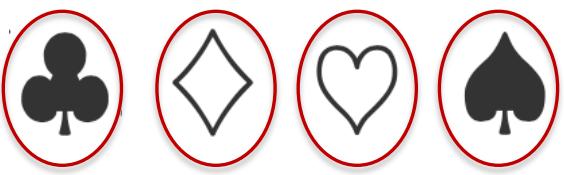
Equivalence Relations & Partial Orders

Let R be a binary relation on A .

- R is *symmetric* iff $\forall x, y \in A$. if $(x, y) \in R$ then $(y, x) \in R$;
- R is *antisymmetric* iff
 $\forall x, y \in A$. if $(x, y) \in R$ and $(y, x) \in R$ then $x = y$.

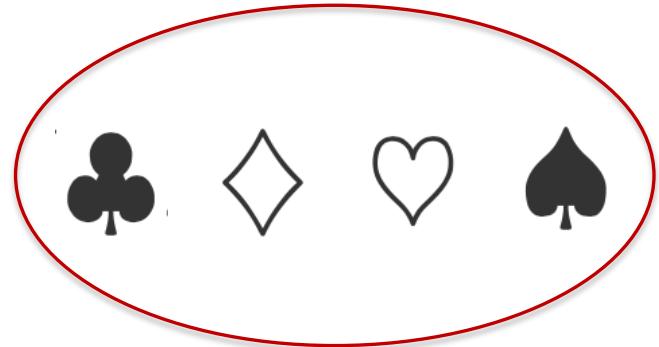
A symmetric preorder is called an *equivalence relation*.

An antisymmetric preorder is called a *partial order*.

$\{(\clubsuit, \clubsuit), (\diamondsuit, \diamondsuit), (\heartsuit, \heartsuit), (\spadesuit, \spadesuit)\}$ 

{
 $(\clubsuit, \clubsuit), (\clubsuit, \diamondsuit), (\clubsuit, \heartsuit), (\clubsuit, \spadesuit),$
 $(\diamondsuit, \clubsuit), (\diamondsuit, \diamondsuit), (\diamondsuit, \heartsuit), (\diamondsuit, \spadesuit),$
 $(\heartsuit, \clubsuit), (\heartsuit, \diamondsuit), (\heartsuit, \heartsuit), (\heartsuit, \spadesuit),$
 $(\spadesuit, \clubsuit), (\spadesuit, \diamondsuit), (\spadesuit, \heartsuit), (\spadesuit, \spadesuit)$

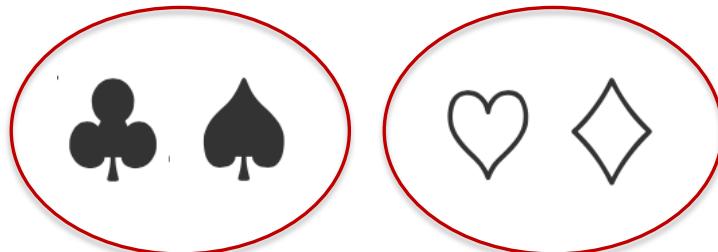
}



Three equivalence relations on A.

{
 $(\clubsuit, \clubsuit), (\clubsuit, \spadesuit), (\spadesuit, \clubsuit), (\spadesuit, \spadesuit),$
 $(\diamondsuit, \diamondsuit), (\diamondsuit, \heartsuit), (\heartsuit, \diamondsuit), (\heartsuit, \heartsuit)$

}



$\{(\clubsuit, \clubsuit), (\diamondsuit, \diamondsuit), (\heartsuit, \heartsuit), (\spadesuit, \spadesuit)\}$ 

{
(\clubsuit, \clubsuit), (\clubsuit, \diamondsuit), (\clubsuit, \heartsuit), (\clubsuit, \spadesuit),
($\diamondsuit, \diamondsuit$), (\diamondsuit, \heartsuit), (\diamondsuit, \spadesuit),
(\heartsuit, \heartsuit), (\heartsuit, \spadesuit),
(\spadesuit, \spadesuit)
}

Two partial orders on A.



{
(\clubsuit, \clubsuit), (\clubsuit, \spadesuit), (\spadesuit, \clubsuit), (\spadesuit, \spadesuit),
($\diamondsuit, \diamondsuit$), (\diamondsuit, \heartsuit), (\heartsuit, \diamondsuit), (\heartsuit, \heartsuit)
}

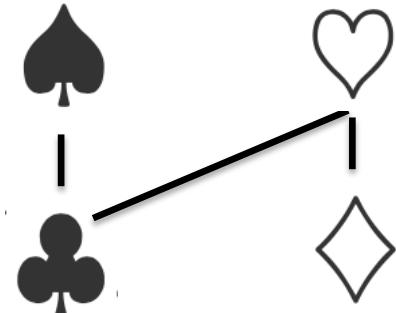
Not a partial order.

$\{(\clubsuit, \clubsuit), (\diamondsuit, \diamondsuit), (\heartsuit, \heartsuit), (\spadesuit, \spadesuit)\}$ 

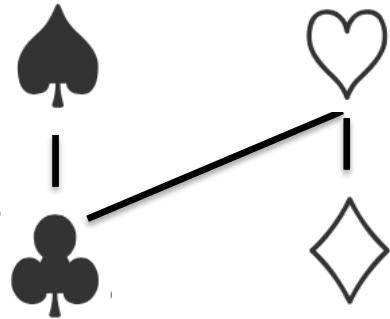
{
 (\clubsuit, \clubsuit), (\clubsuit, \diamondsuit), (\clubsuit, \heartsuit), (\clubsuit, \spadesuit),
 ($\diamondsuit, \diamondsuit$), (\diamondsuit, \heartsuit), (\diamondsuit, \spadesuit),
 (\heartsuit, \heartsuit), (\heartsuit, \spadesuit),
 (\spadesuit, \spadesuit)
}

Three partial orders on A.

{
 (\clubsuit, \clubsuit), (\clubsuit, \spadesuit), (\clubsuit, \heartsuit), (\spadesuit, \spadesuit),
 ($\diamondsuit, \diamondsuit$), (\diamondsuit, \heartsuit), (\heartsuit, \heartsuit)
}



Total Order



Diamond isn't comparable
to either spade or club.



Good to go.

Java

- **equals** – should define an equivalence relation, for items x and y of the same type.
 - If x isn't null, y can't be null;
 - `x.equals(x)`
 - If `x.equals(y)` then `y.equals(x)`
 - If `x.equals(y)` and `y.equals(z)` then `x.equals(z)`.

Java

- **equals** must be in synch with **hashCode**: any two items x and y such that $x.equals(y)$ is true, it must be the case that

$$x.hashCode() == y.hashCode()$$

- NB: it isn't the case that if $x.equals(y)$ is false, then

$$x.hashCode() \neq y.hashCode()$$

Java

- `compareTo` – should define a total order;
- `compareTo` should be *consistent* with `equals`:
for items `x` and `y` of the same type, it should
be the case that:

`x.compareTo(y) == 0` if and only if `x.equals(y)`