**What is Discrete Math**

-Well defined categories

-Int

-Finite Set

-Logic & Methods of Proof

**Proposition**

-A declarative sentence that's true or false, not both.

-Ex:

-3+4=7+0 (true)

-2+2=5 (false)

-Ur mum is fat (true)

-Non-Ex:

-x+5=30 (x=? So neither true or false)

-x is a prime number (neither because x can be multiple numbers)

-What time is it?

-Denote

-True = T = 1

-False = F = 0

**Propositional Variables (aka Sentential Variables)**

-Letters used to represent Proposition

-let s="Seattle"

-let t="bruh"

-s^t (^ is &&) *(more later)*

**1.1 Propositional Logic**

-Atomic Propositions are propositions that cannot be expressed in terms of simpler propositions.

-Negation (Symbol: ¬ or ~)

-Whatever truth value p is, ¬p is opposite.

-Unary Operation, it works with one proposition.

-¬p is a compound proposition. It can be broken down & is dependent on p.

Let $p$ be a proposition. The *negation of* $p$, denoted by $\neg p$ (also denoted by $\overline{p}$), is the statement

"It is not the case that $p$."

The proposition $\neg p$ is read "not $p$." The truth value of the negation of $p$, $\neg p$, is the opposite of the truth value of $p$.

Find the negation of the proposition

"Michael's PC runs Linux"

and express this in simple English.

*Solution:* The negation is

"It is not the case that Michael's PC runs Linux."

This negation can be more simply expressed as

"Michael's PC does not run Linux."

| $p$ | $\neg p$ |
|---|---|
| T | F |
| F | T |

-Conjunction (Symbol: ∧) (AND)

-If both the 2 propositions are true, only then it's true.

-Binary Operator, it works on two propositions

-Ex: Ur mum is fat and ugly

Let $p$ and $q$ be propositions. The *conjunction* of $p$ and $q$, denoted by $p \wedge q$, is the proposition "$p$ and $q$." The conjunction $p \wedge q$ is true when both $p$ and $q$ are true and is false otherwise.

| $p$ | $q$ | $p \wedge q$ |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | F |

-In English, "but" basically means "and". The expectation with "but" is not cared.

-Disjunction / Inclusive Or (Symbol: ∨) (OR)

-True unless both the 2 propositions are false.

-Ex: Students who are fat or ugly can join the bruh community.

Let $p$ and $q$ be propositions. The *disjunction* of $p$ and $q$, denoted by $p \vee q$, is the proposition "$p$ or $q$." The disjunction $p \vee q$ is false when both $p$ and $q$ are false and is true otherwise.

| $p$ | $q$ | $p \vee q$ |
|---|---|---|
| T | T | T |
| T | F | T |
| F | T | T |
| F | F | F |

-Exclusive Or (Symbol: ⊕) (XOR)

-True only when 1 of the 2 propositions is true.

-Ex: You can either pick red pill or blue pill morpheus.

Let $p$ and $q$ be propositions. The *exclusive or* of $p$ and $q$, denoted by $p \oplus q$ (or $p$ XOR $q$), is the proposition that is true when exactly one of $p$ and $q$ is true and is false otherwise.

| $p$ | $q$ | $p \oplus q$ |
|---|---|---|
| T | T | F |
| T | F | T |
| F | T | T |
| F | F | F |

-Conditional Statement / Implication (Symbol: →)

-(p→q) If the hypothesis (p) is true, the conclusion (q) must be true.

-If p is false, then it's true regardless of q.

-Hypothesis/Antecedent → Conclusion/Consequent

-Ex: If I'm elected as chairman of your mum, then I will build back better be like.

-if they aren't elected, then nothing happens.

Let $p$ and $q$ be propositions. The *conditional statement* $p \rightarrow q$ is the proposition "if $p$, then $q$." The conditional statement $p \rightarrow q$ is false when $p$ is true and $q$ is false, and true otherwise. In the conditional statement $p \rightarrow q$, $p$ is called the *hypothesis* (or *antecedent* or *premise*) and $q$ is called the *conclusion* (or *consequence*).

| $p$ | $q$ | $p \rightarrow q$ |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | T |
| F | F | T |

(when p is false, its like: whatever dude, i dont care about q, its true bruh.)

"if $p$, then $q$"
"if $p$, $q$"
"$p$ is sufficient for $q$"
"$q$ if $p$"
"$q$ when $p$"
"a necessary condition for $p$ is $q$"
"$q$ unless $\neg p$"
"$p$ implies $q$"
"$p$ only if $q$"
"a sufficient condition for $q$ is $p$"
"$q$ whenever $p$"
"$q$ is necessary for $p$"
"$q$ follows from $p$"
"$q$ provided that $p$"

("q unless ¬p" is p->q, while "q, unless p is ¬p->q)

-"p only if q"

-"(p)I get wet only if (q)it's raining."

-Under some circumstances, p->q can be false when q is false.

-Weird irl case:

"If Juan has a smartphone, then $2 + 3 = 5$" = T

T or F → T

is true from the definition of a conditional statement, because its conclusion is true. (The truth value of the hypothesis does not matter then.) The conditional statement

"If Juan has a smartphone, then $2 + 3 = 6$" = T

F → F

is true if Juan does not have a smartphone, even though $2 + 3 = 6$ is false.

-Converse, Contrapositive, Inverse

-q→p is the converse of p→q

-¬q→¬p is the contrapositive of p→q (same truth values as p→q, aka p→q & ¬q→¬p are equivalent)

-¬p→¬q = ¬(p→q) is the inverse of p→q (equivalent to q→p)

Find the contrapositive, the converse, and the inverse of the conditional statement

"The home team wins whenever it is raining."

*Solution:* Because "$q$ whenever $p$" is one of the ways to express the conditional statement $p \rightarrow q$, the original statement can be rewritten as

"If it is raining, then the home team wins."

Consequently, the contrapositive of this conditional statement is

"If the home team does not win, then it is not raining."

The converse is

"If the home team wins, then it is raining."

The inverse is

"If it is not raining, then the home team does not win."

Only the contrapositive is equivalent to the original statement. ◀

-not p or q is equivalent (a logical equivalent of conditional)

Note: $\neg p \vee q$ and $p \rightarrow q$ have the same truth value no matter what p and q are.

| $p$ | $q$ | $\neg p$ | $\neg p \vee q$ | $p \rightarrow q$ |
|---|---|---|---|---|
| T | T | F | T | T |
| T | F | F | F | F |
| F | T | T | T | T |
| F | F | T | T | T |

-Biconditional Statement / Bi-implication (Symbol: ↔)

-True if propositions are both T or both F.

Let $p$ and $q$ be propositions. The *biconditional statement* $p \leftrightarrow q$ is the proposition "$p$ if and only if $q$." The biconditional statement $p \leftrightarrow q$ is true when $p$ and $q$ have the same truth values, and is false otherwise. Biconditional statements are also called *bi-implications*.

| $p$ | $q$ | $p \leftrightarrow q$ |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | T |

"$p$ is necessary and sufficient for $q$"
"if $p$ then $q$, and conversely"
"$p$ iff $q$." "$p$ exactly when $q$."

same truth value as $(p \rightarrow q) \wedge (q \rightarrow p)$

-Compound Propositions

-Multiple (atomic) proposition. Formed from existing propositions using logical operators.

-Ex:

**TABLE 7 The Truth Table of $(p \vee \neg q) \rightarrow (p \wedge q)$.**

| $p$ | $q$ | $\neg q$ | $p \vee \neg q$ | $p \wedge q$ | $(p \vee \neg q) \rightarrow (p \wedge q)$ |
|---|---|---|---|---|---|
| T | T | F | T | T | T |
| T | F | T | T | F | F |
| F | T | F | F | F | T |
| F | F | T | T | F | F |

-Precedence of Logical Operators

-Eww, PEMDAS be like of operators discrete mafs

-BEFORE not, Quantifiers have the highest precedence

| *Operator* | *Precedence* |
|---|---|
| $\neg$ | 1 |
| $\wedge$ | 2 |
| $\vee$ | 3 |
| $\rightarrow$ | 4 |
| $\leftrightarrow$ | 5 |

-When conditionals are chained one after another (like p -> q -> r), there is no precedence. In this class it's Left to Right ((p -> q) -> r).

-Ex: $p \vee q \wedge r = p \vee (q \wedge r)$, $p \implies q \vee r = p \implies (q \vee r)$

-Bit Strings OR, AND, XOR

-If you have 2 bit strings, you can do these operations on them through each column

```
01 1011 0110
11 0001 1101
-------------
11 1011 1111   bitwise OR ∨
01 0001 0100   bitwise AND ∧
10 1010 1011   bitwise XOR ⊕
```

**10/4:**

-Truth table practice

| $p$ | $q$ | $\neg p$ | $p \wedge q$ | $\neg p \vee q$ | $(p \wedge q) \implies (\neg p \vee q)$ |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 |

-3 proposition truth table template

| $p$ | $q$ | $r$ |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 1 |
| 0 | 1 | 1 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

**10/6:**

-What is a truth assignment?

-Assigning T or F to a proposition. So it's the left column showing all the possible combo of truth values for each proposition.

-Statement to Truth Table Exercise

A. *If* I get up early today *and* it is sunny today, *then* it will not be a bad day.

p = I get up early today.

q = It's sunny today

r = It'll be a bad day

$(p \wedge q) \implies \neg r$

| $p$ | $q$ | $r$ | $p \wedge q$ | $\neg r$ | $(p \wedge q) \implies \neg r$ |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 |

B. It's a bad day today *only if either* I do not get up early today *or* it is not sunny today (*or both*)

$r \implies \neg p \vee \neg q$

| $p$ | $q$ | $r$ | $\neg p$ | $\neg q$ | $\vee$ | $\implies$ |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 |

-They are logically equivalent, same truth values.

**Propositional Equivalence**

-Tautology

-A compound statement that's always true under all truth assignments.

-Logically Equivalent

-Compound propositions that have the same truth values as each other.

-Compound propositions (let's say p & q) are logically equivalent if the p <--> q is a tautology.

-Denote by p≡q

-Fat Laws table algebra be like

| *Equivalence* | *Name* |
|---|---|
| $p \wedge \mathbf{T} \equiv p$<br>$p \vee \mathbf{F} \equiv p$ | Identity laws |
| $p \vee \mathbf{T} \equiv \mathbf{T}$<br>$p \wedge \mathbf{F} \equiv \mathbf{F}$ | Domination laws |
| $p \vee p \equiv p$<br>$p \wedge p \equiv p$ | Idempotent laws |
| $\neg(\neg p) \equiv p$ | Double negation law |
| $p \vee q \equiv q \vee p$<br>$p \wedge q \equiv q \wedge p$ | Commutative laws |
| $(p \vee q) \vee r \equiv p \vee (q \vee r)$<br>$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$ | Associative laws |
| $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$<br>$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$ | Distributive laws |
| $\neg(p \wedge q) \equiv \neg p \vee \neg q$<br>$\neg(p \vee q) \equiv \neg p \wedge \neg q$ | De Morgan's laws |
| $p \vee (p \wedge q) \equiv p$<br>$p \wedge (p \vee q) \equiv p$ | Absorption laws |
| $p \vee \neg p \equiv \mathbf{T}$<br>$p \wedge \neg p \equiv \mathbf{F}$ | Negation laws |

| |
|---|
| $p \to q \equiv \neg p \vee q$ |
| $p \to q \equiv \neg q \to \neg p$ |
| $p \vee q \equiv \neg p \to q$ |
| $p \wedge q \equiv \neg(p \to \neg q)$ |
| $\neg(p \to q) \equiv p \wedge \neg q$ |
| $(p \to q) \wedge (p \to r) \equiv p \to (q \wedge r)$ |
| $(p \to r) \wedge (q \to r) \equiv (p \vee q) \to r$ |
| $(p \to q) \vee (p \to r) \equiv p \to (q \vee r)$ |
| $(p \to r) \vee (q \to r) \equiv (p \wedge q) \to r$ |

| |
|---|
| $p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$ |
| $p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$ |
| $p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$ |
| $\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$ |

| *Operator* | *Precedence* |
|---|---|
| $\neg$ | 1 |
| $\wedge$<br>$\vee$ | 2<br>3 |
| $\to$<br>$\leftrightarrow$ | 4<br>5 |

-De Morgan's on more than 2 propositions

$\neg(a \vee b \wedge c) \equiv \neg a \wedge \neg b \vee \neg c$

-Negate and flip a cuh.

-Not And laws maybe idk

$$\neg(p \wedge q) \equiv p \underline{\Delta} q$$

$$\neg p \equiv p \underline{\Delta} p$$

-Denoting Repeating / Generalize Or & And

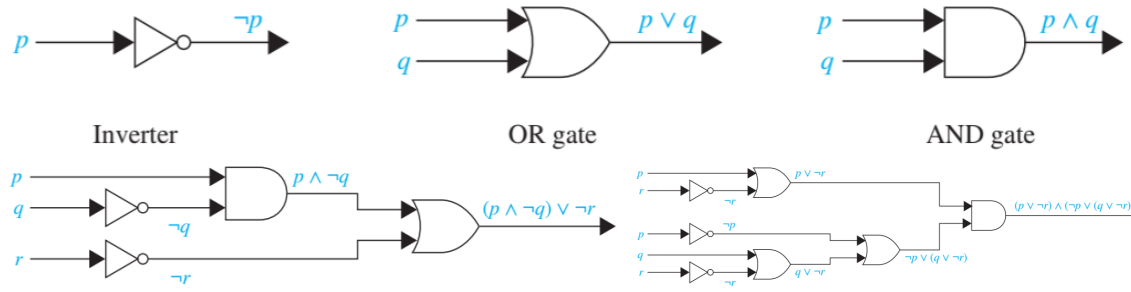
$\bigvee_{j=1}^{n} p_j$ for $p_1 \vee p_2 \vee \cdots \vee p_n$

$\bigwedge_{j=1}^{n} p_j$ for $p_1 \wedge p_2 \wedge \cdots \wedge p_n$

-De Morgan's Law

$$\neg\left(\bigvee_{j=1}^{n} p_j\right) \equiv \bigwedge_{j=1}^{n} \neg p_j \text{ and } \neg\left(\bigwedge_{j=1}^{n} p_j\right) \equiv \bigvee_{j=1}^{n} \neg p_j$$

**Circuits irl application**




**Satisfiability**

-Satistiable

-A compound that has at least a truth assignment such that the proposition is true.

-Application: Circuit Validation

-Run the output of the model vs the implementation of a circuit into XOR. If XOR is ever true, there's a problem.

-Unsatisfiable

-The opposite of a tautology

-aka Contradiction

A compound proposition that is always true, no matter what the truth values of the propositional variables that occur in it, is called a *tautology*. A compound proposition that is always false is called a *contradiction*. A compound proposition that is neither a tautology nor a contradiction is called a *contingency*.

-Contingency is neither a tautology or contradiction (the common one)

**Predicates**

-A predicate is a property that the subject of the statement can have. Like, x (subject) has fur (property (of a furry🤮)

-Let say x+5=30

-x is the subject of the statement

-We can give the statement "x+5=30" a name. So, let P(x) be a predicate of the subject x of the statement.

-Is P(17) true? P(17) is false, so no.

-Is P(25) true? Yes.

-We can think of predicates as propositional functions that map values (or subjects) to truth values.

-Subjects of predicates can be of any type.

-Ex: Let P(x) be the statement "x is older than 21 years."

-P(Tom Hanks) is T.

-P(Greta Thunberg) is F.

-Multi-variable Predicates.

-Ex: Let Q(x, y) be the statement "x is odd and y is odd."

-Ex: Let F(x, y) be the statement "x is a friend of y.:

-F(x, y) -> F(y, x)

-Generalize

-Ex: $P(x_1, x_2 \ldots, x_n)$ is a propositional function that maps a tuple (the x_1, 2 ...n) to true or false

-The x goes to n. These are called n-place/n-ary predicates.

**Quantification**

-Universal Quantification (∀)

-"For all" (every possible subject in the domain is true until a counterexample is found to falsify it)

The *universal quantification* of $P(x)$ is the statement

"$P(x)$ for all values of $x$ in the domain."

The notation $\forall x P(x)$ denotes the universal quantification of $P(x)$. Here $\forall$ is called the **universal quantifier**. We read $\forall x P(x)$ as "for all $x P(x)$" or "for every $x P(x)$." An element for which $P(x)$ is false is called a **counterexample** to $\forall x P(x)$.

| *Statement* | *When True?* | *When False?* |
|---|---|---|
| $\forall x P(x)$ | $P(x)$ is true for every $x$. | There is an $x$ for which $P(x)$ is false. |
| $\exists x P(x)$ | There is an $x$ for which $P(x)$ is true. | $P(x)$ is false for every $x$. |

-Ex: We make statements such as

1. All actors have a fine arts degree.
2. All computer programmers have a CS degree.
3. For every integer x, (x^2) > 0

-Domain of discourse, universe of discourse, domain

-The set of all possible variables of a variable.

-The universal quantification of P(x) is the statement "P(x) for all values x in the domain"

1. All actors have a fine arts degree.

-∀x P(x)

-The domain of x are actors

-P(x) is "x has a fine arts degree".

3. For every integer x, (x^2) > 0

-∀x G(x)

-The domain of x are integers.

-G(x) is “x^2 > 0”
-A Counter Example i
-Specifying domain is important.
-Ex: Let P(x) be the statement “x^2 >= x”
-Suppose the domain of x in P(x) are all integers.
-∀x P(x) is true.
-Suppose the domain of x in P(x) are all real number.
-∀x P(x) is false.
-Counterexample: x is ½.

-Existential Quantification ( ∃ )
-”For some” (True when at least one subject is found in the domain to be true.)

The *existential quantification* of $P(x)$ is the proposition

“There exists an element $x$ in the domain such that $P(x)$.”

We use the notation $\exists x P(x)$ for the existential quantification of $P(x)$. Here $\exists$ is called the *existential quantifier*.

-Uniqueness Quantification (!∃ , ∃!, ∃_1)
-”For one” (only one possible subject in the domain must be true, else false)

the one that is most often seen is the **uniqueness quantifier**, denoted by $\exists!$ or $\exists_1$. The notation $\exists! x P(x)$ [or $\exists_1 x P(x)$] states “There exists a unique $x$ such that $P(x)$ is true.” (Other phrases for uniqueness quantification include “there is exactly one” and “there is one and only one.”) For instance, $\exists! x(x - 1 = 0)$,

-Quantifier is like a loop
-So like for “for all”, you can think of it like a loop that checks all values in the array if it’s true.
-Quantification over a finite domain
-Universal (same as generalizing and)

When the domain of a quantifier is finite, that is, when all its elements can be listed, quantified statements can be expressed using propositional logic. In particular, when the elements of the domain are $x_1, x_2, \ldots, x_n$, where $n$ is a positive integer, the universal quantification $\forall x P(x)$ is the same as the conjunction

$$P(x_1) \wedge P(x_2) \wedge \cdots \wedge P(x_n),$$

**EXAMPLE 15** What is the truth value of $\forall x P(x)$, where $P(x)$ is the statement “$x^2 < 10$” and the domain consists of the positive integers not exceeding 4?

*Solution:* The statement $\forall x P(x)$ is the same as the conjunction

$$P(1) \wedge P(2) \wedge P(3) \wedge P(4),$$

because the domain consists of the integers 1, 2, 3, and 4. Because $P(4)$, which is the statement “$4^2 < 10$,” is false, it follows that $\forall x P(x)$ is false. ◀

-Existential (same as generalizing or)

Similarly, when the elements of the domain are $x_1, x_2, \ldots, x_n$, where $n$ is a positive integer, the existential quantification $\exists x P(x)$ is the same as the disjunction

$$P(x_1) \vee P(x_2) \vee \cdots \vee P(x_n)$$

**EXAMPLE 16** What is the truth value of $\exists x P(x)$, where $P(x)$ is the statement “$x^2 > 10$” and the universe of discourse consists of the positive integers not exceeding 4?

*Solution:* Because the domain is $\{1, 2, 3, 4\}$, the proposition $\exists x P(x)$ is the same as the disjunction

$$P(1) \vee P(2) \vee P(3) \vee P(4).$$

Because $P(4)$, which is the statement “$4^2 > 10$,” is true, it follows that $\exists x P(x)$ is true. ◀

-Quantification over a restricted domain
-After the quantifier, you can specify a domain with an inequality.

What do the statements $\forall x < 0\,(x^2 > 0)$, $\forall y \neq 0\,(y^3 \neq 0)$, and $\exists z > 0\,(z^2 = 2)$ mean, where the domain in each case consists of the real numbers?

*Solution:* The statement $\forall x < 0\ (x^2 > 0)$ states that for every real number $x$ with $x < 0$, $x^2 > 0$. That is, it states “The square of a negative real number is positive.” This statement is the same as $\forall x(x < 0 \rightarrow x^2 > 0)$.

The statement $\forall y \neq 0\,(y^3 \neq 0)$ states that for every real number $y$ with $y \neq 0$, we have $y^3 \neq 0$. That is, it states “The cube of every nonzero real number is nonzero.” This statement is equivalent to $\forall y(y \neq 0 \rightarrow y^3 \neq 0)$.

Finally, the statement $\exists z > 0\,(z^2 = 2)$ states that there exists a real number $z$ with $z > 0$ such that $z^2 = 2$. That is, it states “There is a positive square root of 2.” This statement is equivalent to $\exists z(z > 0 \wedge z^2 = 2)$. ◀

-Equivalency be like

$\forall x < 0\,(x^2 > 0) \equiv \forall x(L(x) \rightarrow G(x))$ (where L(x) is “x<0”, G(x) is “x^2>0”)

-Multiple Quantifiers for different subject
-So for P(x,y)
-You can do ∀x∀y
-Or you can do ∀x∃y
-Or ∀x>0∃y
-switch ordering, etc.
-Scope, Bound/Free Variables
-An example is ∀xP(x) ∨ Q(x) ≡ (∀xP(x)) ∨ Q(x), which is ≠ ∀x(P(x) ∨ Q(x))
-P(x) is bounded by the for all x, since it is in the parenthesis and is in scope.
-Q(x) is free.

-Careful for: $\exists x P(x) \wedge \exists x Q(x) \not\equiv \exists x (P(x) \wedge Q(x))$

-First says there's a subject in x that makes P(x) true and another subject in x (could be same or dif) that makes Q(x) true.

-Second says there's a subject in x that makes both P(x) and Q(x) true.

-For these, if the 2nd is true, then the 1st is true.

-Logical Equivalent for Quantifiers

-2 statements involving predicates and/or quantifiers are logically equivalent iff they have the same truth value no matter the domain or meaning of predicates.

-That means you can have 2 different predicates that are paired with dif operators or quantifiers to get the same truth table.

-Ex:

$$\forall x(P(x) \wedge Q(x)) \equiv \forall x P(x) \wedge \forall x Q(x)$$
$$\equiv \forall x P(x) \wedge \forall y Q(y)$$

-Doesn't matter what subject or domain crap. Just if they have the same truth value.

$$\forall x(P(x) \vee Q(x)) \neq \forall x P(x) \vee Q(x)$$

-Left side means if each x has both properties. Right means if some x has P, and another Q, or has both.

-In this case if right side is true, left is true.

$$\forall x\, P(x) \equiv \neg(\exists x\, (\neg P(x)))$$

**Negating Quantified Expressions**

$$\neg \forall x\, P(x) \equiv \exists x \neg P(x)$$

$$\neg \exists x\, P(x) \equiv \forall x \neg P(x)$$

-Proof (usage of Generalize and De Morgan's)

$$\neg \forall x\, P(x) \equiv \neg(P(x_1) \wedge P(x_2) \wedge \ldots \wedge P(x_n)) \quad \text{(defn of } \forall\text{)}$$
$$\equiv \neg \bigwedge_{i=1}^{n} P(x_i) \quad \text{(defn of } \wedge\text{)}$$
$$\equiv \bigvee_{i=1}^{n} \neg P(x_i) \quad \text{(De Morgan's)}$$
$$\equiv \neg P(x_1) \vee \neg P(x_2) \vee \ldots \vee \neg P(x_n) \quad \text{(defn of } \vee\text{)}$$
$$\equiv \exists x \neg P(x) \quad \text{(defn of } \exists\text{)}$$

-Engrish

example: Every student in TCSS 321 got 100% on Midterm 1.

The negation is

It is not the case that every student in TCSS 321 got 100% on Midterm 1.

This is the same as

There is a student in TCSS 321 who did not get 100% on Midterm 1.

-Big Ex w/ multiple quantifiers:

example:

For all $x$, $y$, and $z$, if $x<y$ and $y<z$, then $x<z$.

$$\forall x \forall y \forall z ((L(x,y) \wedge L(y,z)) \rightarrow L(x,z))$$

where $L(a,b)$ is the statement "$a$ is less than $b$."

How do we express the negation?

$$\neg \forall x \forall y \forall z ((L(x,y) \wedge L(y,z)) \rightarrow L(x,z))$$
$$\equiv \exists x \neg \forall y \forall z ((L(x,y) \wedge L(y,z)) \rightarrow L(x,z))$$
$$\equiv \exists x \exists y \neg \forall z ((L(x,y) \wedge L(y,z)) \rightarrow L(x,z))$$
$$\equiv \exists x \exists y \exists z \neg ((L(x,y) \wedge L(y,z)) \rightarrow L(x,z))$$
$$\equiv \exists x \exists y \exists z \neg (\neg (L(x,y) \wedge L(y,z)) \vee L(x,z)) \quad p \rightarrow q \equiv \neg p \vee q$$
$$\equiv \exists x \exists y \exists z \neg ((\neg L(x,y) \vee \neg L(y,z)) \vee L(x,z)) \quad \text{De Morgan's law}$$
$$\equiv \exists x \exists y \exists z (L(x,y) \wedge L(y,z) \wedge \neg L(x,z)) \quad \text{De Morgan's law}$$

**Translating English to Quantifiers**

-Ex:

-Every student in this class has taken a class using Java.

-(aka) For every student x in this class, x has taken a class using Java.

-Domain: Students in this class

-J(x): “x has taken a class using Java.”

-∀x J(x)

-Domain: Students

-C(x): “x is in this class.”

-J(x): “x has taken a class using Java.”

-∀x (C(x)->J(x)) (conditional because we don’t care if they are not in the class)

-Domain: People

-S(x): “x is a student.”

-C(x): “x is in this class.”

-J(x): “x has taken a class using Java.”

-∀x (S(x) -> (C(x) -> J(x)))

-∀x ((S(x) ∧ C(x)) -> J(x))

-Ex:

-There is a student in this class using Java but also can write programs in Pascal.

-Domain: Students

-C(x): “x is in this class.”

-J(x): “x can program in Java.”

-P(x): “x can program in Pascal.”

-∀x(C(x) -> (J(x) ∧ P(x))

**Quantifier Tautology Stuff**

-Is $\forall x P(x) \rightarrow \exists x P(x)$ a tautology?

-Surprisingly, it’s not: when the domain is empty.

-There is no counter to the universal, and no example for existence. T -> F ≡ T

-Is $\forall x P(x) \vee \exists x P(x)$ a tautology?

-No. It’s when all subjects are false to P(x).

-Is $(\forall x P(x) \vee \exists x P(x)) \vee (\forall x P(x) \rightarrow \exists x P(x))$ a tautology?

-Yea. When all subjects are false, the conditional is true.

-When proving if a statement with quantification is a tautology:

-True when the domain is empty?

-True when the domain is not empty?

**Nested Quantifiers**

-Quantifiers that nested within each other. (Even without parenthesis)

-Ex:

$\forall x \forall y (x + y = y + x)$ (true)

$\forall x \forall y \forall z ((x + y) + z = x + (y + z))$ (true)

$\forall x \exists y (y = x^2)$ (true, there always exists a y that is x^2. You can find a y for each x that equals x^2.)

$\forall x \exists y (x = y^2)$ (false, when x is negative y is gg to try and equal.)

$\exists x \forall y (y = x^2)$ (false, not all y satisfy x^2.)

$\exists x \forall y (y^2 = x)$ (false, same as above.)

$\exists x \forall y (y \neq x^2)$ (false, when x=0 & y=0.)

$\forall x \forall y (y^2 = x)$ (false, not all satisfies the other. x=0 & y=1)

-The order of nested quantifier ACTUALLY MATTERS!

-AxEy is not the same as EyAx.

-It depends on the statement you are trying to make. Really isn’t a rule.

-Nested loop analogy be like

-Table of cases with 2 variables

**TABLE 1 Quantifications of Two Variables.**

| *Statement* | *When True?* | *When False?* |
|---|---|---|
| $\forall x \forall y P(x, y)$<br>$\forall y \forall x P(x, y)$ | $P(x, y)$ is true for every pair $x, y$. | There is a pair $x, y$ for which $P(x, y)$ is false. |
| $\forall x \exists y P(x, y)$ | For every $x$ there is a $y$ for which $P(x, y)$ is true. | There is an $x$ such that $P(x, y)$ is false for every $y$. |
| $\exists x \forall y P(x, y)$ | There is an $x$ for which $P(x, y)$ is true for every $y$. | For every $x$ there is a $y$ for which $P(x, y)$ is false. |
| $\exists x \exists y P(x, y)$<br>$\exists y \exists x P(x, y)$ | There is a pair $x, y$ for which $P(x, y)$ is true. | $P(x, y)$ is false for every pair $x, y$. |

**Translating English to Nested Quantifiers**

-For all x, y, and z, if x<y and y<z, then x<z.

$\forall x, y, z ((L(x, y) \wedge L(y, z)) \rightarrow L(x, z))$

-where L(x,y): x is less than y.”

-Negation

$[\forall x, y, z ((L(x, y) \wedge L(y, z)) \rightarrow L(x, z))] \equiv \exists x, y, z \neg ((L(x, y) \wedge L(y, z)) \rightarrow L(x, z))$

-More Nested Quantifiers

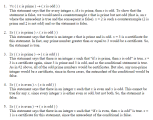
$\forall x((\text{x is prime}) \rightarrow (\text{x is odd}))$ (false, x=2)

$\exists x((\text{x is prime}) \wedge (\text{x is odd}))$ (true)

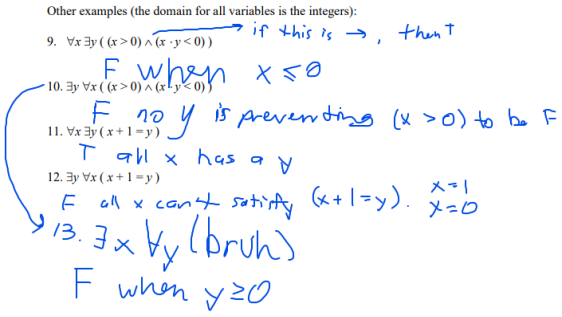
$\exists x((\text{x is prime}) \rightarrow (\text{x is odd}))$ (true)

$\exists x((\text{x is even}) \wedge (\text{x is odd}))$ (bruh false)

$\exists x((\text{x is } even) \rightarrow (\text{x is odd}))$ ()



-Extra Practice



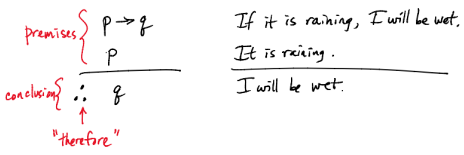

**Rules of Inference**

-Argument

-A sequence of statements that ends with a conclusion.

-Premise

-A statement that is assumed to be true.



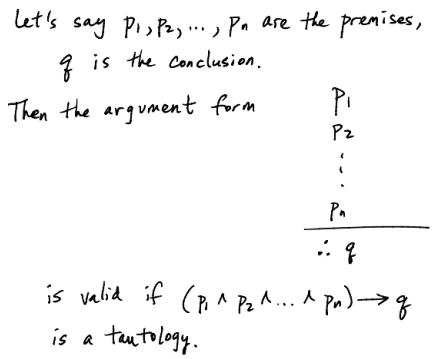

-Validating an Argument

-It's valid when the conclusion of the argument must be true if all the premises are true.

-We can prove it by showing that a compound statement of all premises implying the conclusion is a tautology.

-Back to the rain & wet example, it is a tautology when ((p -> q) ∧ p) -> q making it valid.

-More general explanation of validation: "all premises implying the conclusion must be a tautology"




-It can also be valid when a premise is clearly false.

-If the earth explodes, then the economy will be great.(p -> q)

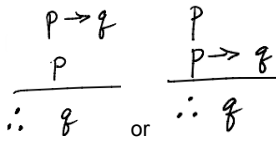
-The earth explodes. (p)

-Therefore the economy will be great. (∴q)

-Argument Form

-The argument form is when you strip away the text

-So with the rain & wet example:

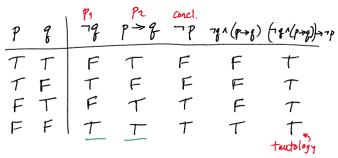



(it was modus ponens)

- "Or" is because communicative by conjunction in def of validation. In >2 premises, you can arrange them however you want.

-Fat All them Rules of Inference

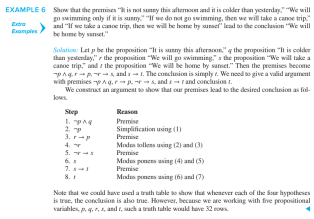
| Rule of Inference | Tautology | Name |
|---|---|---|
| $p$<br>$p \to q$<br>$\therefore q$ | $(p \wedge (p \to q)) \to q$ | Modus ponens |
| $\neg q$<br>$p \to q$<br>$\therefore \neg p$ | $(\neg q \wedge (p \to q)) \to \neg p$ | Modus tollens |
| $p \to q$<br>$q \to r$<br>$\therefore p \to r$ | $((p \to q) \wedge (q \to r)) \to (p \to r)$ | Hypothetical syllogism |
| $p \vee q$<br>$\neg p$<br>$\therefore q$ | $((p \vee q) \wedge \neg p) \to q$ | Disjunctive syllogism |
| $p$<br>$\therefore p \vee q$ | $p \to (p \vee q)$ | Addition |
| $p \wedge q$<br>$\therefore p$ | $(p \wedge q) \to p$ | Simplification |
| $p$<br>$q$<br>$\therefore p \wedge q$ | $((p) \wedge (q)) \to (p \wedge q)$ | Conjunction |
| $p \vee q$<br>$\neg p \vee r$<br>$\therefore q \vee r$ | $((p \vee q) \wedge (\neg p \vee r)) \to (q \vee r)$ | Resolution |



(modus tollens truth table)

-When constructing an argument from English, you should try to simplify each premise ASAP. Label your lines so you can prove simplification.

| Step | Reason |
|---|---|
| 1. $\neg p \wedge q$ | Premise |
| 2. $\neg p$ | Simplification using (1) |
| 3. $r \to p$ | Premise |
| 4. $\neg r$ | Modus tollens using (2) and (3) |
| 5. $\neg r \to s$ | Premise |
| 6. $s$ | Modus ponens using (4) and (5) |
| 7. $s \to t$ | Premise |
| 8. $t$ | Modus ponens using (6) and (7) |


EXAMPLE 6 Extra Examples

Show that the premises "It is not sunny this afternoon and it is colder than yesterday," "We will go swimming only if it is sunny," "If we do not go swimming, then we will take a canoe trip," and "If we take a canoe trip, then we will be home by sunset" lead to the conclusion "We will be home by sunset."

*Solution:* Let $p$ be the proposition "It is sunny this afternoon," $q$ the proposition "It is colder than yesterday," $r$ the proposition "We will go swimming," $s$ the proposition "We will take a canoe trip," and $t$ the proposition "We will be home by sunset." Then the premises become $\neg p \wedge q$, $r \to p$, $\neg r \to s$, and $s \to t$. The conclusion is simply $t$. We need to give a valid argument with premises $\neg p \wedge q$, $r \to p$, $\neg r \to s$, and $s \to t$ and conclusion $t$.

We construct an argument to show that our premises lead to the desired conclusion as follows.

| Step | Reason |
|---|---|
| 1. $\neg p \wedge q$ | Premise |
| 2. $\neg p$ | Simplification using (1) |
| 3. $r \to p$ | Premise |
| 4. $\neg r$ | Modus tollens using (2) and (3) |
| 5. $\neg r \to s$ | Premise |
| 6. $s$ | Modus ponens using (4) and (5) |
| 7. $s \to t$ | Premise |
| 8. $t$ | Modus ponens using (6) and (7) |

Note that we could have used a truth table to show that whenever each of the four hypotheses is true, the conclusion is also true. However, because we are working with five propositional variables, $p$, $q$, $r$, $s$, and $t$, such a truth table would have 32 rows.

-You can also introduce a premise with a single used proposition to simplify and prove???

-Quantifiers Rules of Inference

| Rule of Inference | Name |
|---|---|
| $\forall x P(x)$<br>$\therefore P(c)$ | Universal instantiation |
| $P(c)$ for an arbitrary $c$<br>$\therefore \forall x P(x)$ | Universal generalization |
| $\exists x P(x)$<br>$\therefore P(c)$ for some element $c$ | Existential instantiation |
| $P(c)$ for some element $c$<br>$\therefore \exists x P(x)$ | Existential generalization |

-c is a possible x value. Because every x is true, there is some arbitrary c. Because some x is true, there is a particular c.

-Ex:

-All humans are mortal.

-Socrates is human.

-Therefore Socrates is mortal.

-1: Ax(H(x) -> M(x))

-2: H(Socrates)

-3: H(Socrates) -> M(Socrates) (Universal Instantiation, 1)

-4: M(Socrates) (Modus Ponens, 3)

**Proofs if p->q**

-Even vs Odd Integer Proof (Direct Proof)

-The integer n is even if there is an integer k such that n=2k.

-The integer n is odd if there is an integer k such that n=2k+1

-Claim: If n is an odd integer, then n^2 is odd. (false btw)

-(aka) For all integers n, if n is an odd integer, then n^2 is odd.

Let $P(n)$ be "$n$ is odd."
Let $Q(n)$ be "$n^2$ is odd." $\forall n\ (P(n) \to Q(n))$

-The Proof:

-Suppose n is odd. (Assume that the hypothesis is TRUE when claim is p->q)

-Then n=2k+1 for some integer k.

-We need to show that $n^2 = (2k+1)^2 = 4k^2+4k+1$ is odd.

-Well, $n^2 = k^2 + 4k + 1 = 2(2k^2+2k)+1$

-Since $2k^2+2$ is an integer, then n^2 is odd.■

-Proof line by line.

-1. integer n is odd (premise)

-2. n=2k+1 for some integer k (def of odd, (1)

-3. $n^2=(2k+1)^2=4k^2+4k+1=2(2k^2+2k)+1$ (2, algebra)

-4. n^2 is odd. (def of odd, 3)

-So, if n is odd, then n^2 is odd.■

-Proof by Contraposition

-Based on the fact that p->q ≡ ¬q->¬p and start with ¬q (contrapositive def)

-Example:

-Claim: If n is an integer and 3n+2 is odd (p), then n is odd (q).

-Direct Proof failure attempt😞:

-3n+2 is odd (premise)

-3n+2=2k+1 for some integer k (def of odd)

-3n+1=2k (algebra)

-n=(2k-1)/3 [bruh moment, don't work]

-Contraposition, Proof big juice😗😎:

-Suppose n is even. (premise, assuming ¬q)

-Then n=2k for some integer k (def of even)

-3n+2=3(2k)+2=6k+2=2(3k+1) (algebra)

-3n+2 is even (def of even)

-Therefore if n is even, then 3n+2 is even.

-aka, Therefore if 3n+2 is odd, then n is odd. (contrapositive)

-Proof by Contradiction

-Assume that the claim is false (¬p->q) and reach a contradiction. (aka Take the claim, negate the claim, prove it can't be true, therefore the orig is true.)

-So if "p", suppose "¬p", blah blah statements, reach a contradiction p∧¬q

-¬p->(p∧¬q)

-Proof Ex:

-Claim: If 3n+2 is odd, then n is odd.

-Let p be "3n+2 is odd".

-Let q be "n is odd".

-(By way of contradiction) Suppose p->q (orig claim) is false.

-Then ¬(p->q)≡(p∧¬q) is true.

-"3n+2" is odd and n is even should be true since we are supposing a contradiction.

$$\begin{aligned} 3n+2 &= 3(2k)+2 \\ &= 6k+2 \\ &= 2(3k+1) \end{aligned}$$

-So, "3n+2" is even, contradicting "3n+2 is odd" (p).

-Therefore the original claim is true.

-Proof Ex:

-Claim: At least 4 of any 22 days must fall on the same day of the week.

-p->q

-p: 22 days

-(Proof idea): Suppose the claim is not true, then each of the 7 days of the week occurs at most 3 times in the 22 days. But, that only accounts for 21 days, contradicting the fact that there are 22 days to account for. So, the original claim is true.

-Proof by Cases

$$\begin{array}{l} p_1 \vee p_2 \vee p_3 \vee \ldots \vee p_n \\ p_1 \rightarrow a \\ p_2 \rightarrow a \\ \vdots \\ \underline{p_2 \rightarrow a} \\ \therefore a \end{array}$$

-Ex: For all int n, n^2 >= 0

-Proof: break up ">="

-case 1: n>0

-case 2: n=0

-case 3: n<0

-All are true, therefore the claim is true.

-Proof by Exhaustion

-Like bruteforce, examine every single element in the domain.

-Vacuous proof

-Where the hypothesis is false, so the claim p->q is true

-Ex: The empty set is a subset of a set S. The empty set has no element, so by defn of subset, the hypothesis is false, making the whole conditional true.

**Mistakes in Proof**

-Improper use of a fallacy

-Affirming the conclusion. (looks like some fake modus tollens)

p->q

q

∴p

-Denying Hypothesis (looks like fake modus ponens)

p->q
¬p
∴q

-Begging the question, which is using what you are trying to show to help you prove it.

**Logical Equivalences and Recommended Proof Technique Table**

| Statement Form | Proof Approach |
|---|---|
| $\forall x.\ P$ | ***Direct proof:*** Pick an arbitrary $x$, then prove $P$ is true for that choice of $x$.<br><br>***By contradiction:*** Suppose for the sake of contradiction that there is some $x$ where $P$ is false. Then derive a contradiction. |
| $\exists x.\ P$ | ***Direct proof:*** Do some exploring and find a choice of $x$ where $P$ is true. Then, write a proof explaining why $P$ is true in that case.<br><br>***By contradiction:*** Suppose for the sake of contradiction that $P$ is always false and derive a contradiction. |
| $\neg P$ | ***Direct proof:*** Simplify your formula by pushing the negation deeper, then apply the appropriate rule.<br><br>***By contradiction:*** Suppose for the sake of contradiction that $P$ is true, then derive a contradiction. |
| $P \wedge Q$ | ***Direct proof:*** Prove each of $P$ and $Q$ independently.<br><br>***By contradiction:*** Assume $\neg P \vee \neg Q$. Then, try to derive a contradiction. |
| $P \vee Q$ | ***Direct proof:*** Prove that $\neg P \rightarrow Q$, or prove that $\neg Q \rightarrow P$.<br><br>***By contradiction:*** Assume $\neg P \wedge \neg Q$. Then, try to derive a contradiction. |
| $P \rightarrow Q$ | ***Direct proof:*** Assume $P$ is true, then prove $Q$.<br><br>***By contradiction:*** Assume $P$ is true and $Q$ is false, then derive a contradiction.<br><br>***By contrapositive:*** Assume $\neg Q$, then prove $\neg P$. |
| $P \leftrightarrow Q$ | Prove both $P \rightarrow Q$ and $Q \rightarrow P$. |

**Sets**

-Set

-A set is an unordered collection of distinct objects called elements/members of the set.

-A set is said to contain its elements.

-We write a∈A to denote that 'a' is an element of set A. (a∉A is not in the set)

> A *set* is an unordered collection of distinct objects, called *elements* or *members* of the set. A set is said to *contain* its elements. We write $a \in A$ to denote that $a$ is an element of the set $A$. The notation $a \notin A$ denotes that $a$ is not an element of the set $A$.

-How to describe a set?

-Roster Method (array be like)

A = {1,2,3,4,5,6}

A = {2,4,6,8,...,100} (ellipses indicates pattern)

-Set Builder Notation

P = {x | x is perfect square} (aka "Let P be the set of all x where x is a prefect square")

-We can assign "x is a perfect square" into a predicate, like Q(x).

= {x | x=y^2, for some integer}

= {x | ∃y(x=y^2)} (where y is an int)

-Common Sets

$\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ natural numbers

$\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ integers

$\mathbb{Z}^+ = \{1, 2, 3, \ldots\}$ positive integers

$\mathbb{Q} = \{ p/q \mid p \in \mathbb{Z}, q \in \mathbb{Z}, \text{ and } q \neq 0 \}$ rational numbers

$\mathbb{R}$ is the real numbers

$\mathbb{R}^+$, positive real numbers

$\mathbb{C}$, complex numbers

(irrational: fancy P)

-You can define your own.

-Empty Set

-The set of no elements

$\emptyset$ or $\{\}$

-Sets can be elements in another set, so an empty set counts as an element still.

$\{\}$ ↑ 0 elements

$\{\emptyset\} = \{\{\}\}$ ↑ 1 element

-Subset

-Set A is a subset of set B iff every element of A is also an of B. (A⊆B)

-Set B is the superset of set A (B⊇A)

> The set $A$ is a *subset* of $B$, and $B$ is a *superset* of $A$, if and only if every element of $A$ is also an element of $B$. We use the notation $A \subseteq B$ to indicate that $A$ is a subset of the set $B$. If, instead, we want to stress that $B$ is a superset of $A$, we use the equivalent notation $B \supseteq A$. (So, $A \subseteq B$ and $B \supseteq A$ are equivalent statements.)

$\forall x(x \in A \rightarrow x \in B)$

*Showing that A is a Subset of B* To show that $A \subseteq B$, show that if $x$ belongs to $A$ then $x$ also belongs to $B$.

*Showing that A is Not a Subset of B* To show that $A \not\subseteq B$, find a single $x \in A$ such that $x \notin B$.

-Example:

$\mathbb{Z} \subseteq \mathbb{Q}$

-This is true because **Z**/1 = an element in **Q.**

-Proper Subset

-Set A is a subset of set B iff every element of A is also an of B, but B has at least an element not in A. (A⊂B).

-Equivalent

-2 sets are equal iff they have the same elements.

Two sets are *equal* if and only if they have the same elements. Therefore, if $A$ and $B$ are sets, then $A$ and $B$ are equal if and only if $\forall x(x \in A \leftrightarrow x \in B)$. We write $A = B$ if $A$ and $B$ are equal sets.

-Claim: If A⊆B and B⊇A, then A=B.

-Proof (Contraposition):

-Suppose A≠B

-Then there must be an element of A that is not in B or vice versa.

A⊈B or B⊈A

¬(A⊆B) ∨ ¬(B⊆A)

¬((A⊆B) ∧ (B⊆A))

-So, if A≠B, then it is not the case that A⊆B and B⊆A

-Therefore by contraposition, if A⊆B and B⊆A then A=B.■

-Proof (Direct):

$\forall x((x \in A) \iff (x \in B))$

$\forall x((x \in B) \iff (x \in A))$

$\forall x\ (x \in A \leftrightarrow x \in B)$ (subset def)

-Therefore, it is by definition that it is equal.

-Proof for Converse?

-Membership Tables

-Similar to truth tables, but shows if an element from one set is related to another. Useful to prove equality.

-Ex: Use a membership table to show that $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$.

| $A$ | $B$ | $C$ | $B \cup C$ | $A \cap (B \cup C)$ | $A \cap B$ | $A \cap C$ | $(A \cap B) \cup (A \cap C)$ |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

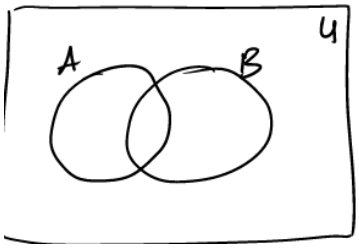
-Union acts like OR, and Intersection like AND.

-Methods of Proving Set Identities

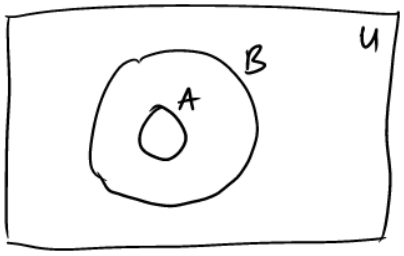
| *Description* | *Method* |
|---|---|
| Subset method | Show that each side of the identity is a subset of the other side. |
| Membership table | For each possible combination of the atomic sets, show that an element in exactly these atomic sets must either belong to both sides or belong to neither side |
| Apply existing identities | Start with one side, transform it into the other side using a sequence of steps by applying an established identity. |

-Venn Diagrams

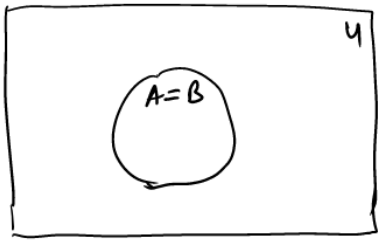
-General Situation



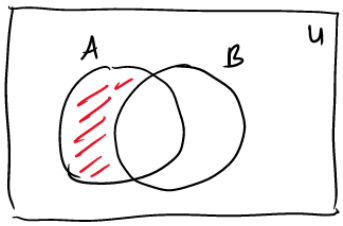

-A subset of B




-A=B




-Elements in A but not in B (but you can specify if shading means something else)




-Size of a Set

-A set S is **finite** if there are exactly n distinct elements in S, where n is a non-negative integer.

-n is the cardinality of S, denoted |S|

-A set S is infinite if it is not finite.

-Powerset

-The powerset of a set S is the set of all subsets of S, denoted $\mathcal{P}(S)$

-Ex:

$\mathcal{P}(\{3,4\}) = \{\emptyset, \{3\}, \{4\}, \{3,4\}\}$

$\mathcal{P}(\{0,1,2\}) = \{\emptyset, \{0\}, \{1\}, \{2\}, \{0,1\}, \{0,2\}, \{1,2\}, \{0,1,2\}\}$ (These are not permutation, not ordered)

$\mathcal{P}(\emptyset) = \{\emptyset\}$ Note that $\emptyset \subseteq \emptyset$.

$\mathcal{P}(\mathbb{Z}) = \{x \mid x \text{ is a subset of } \mathbb{Z}\}$

$=$ the set of all subsets of $\mathbb{Z}$.

-The cardinality of P(S) (aka |P(S)|) is 2^n, where n is the number of elements in Set S.

-If S is infinite, then $|P(S)| = 2^{|S|}$

-$P(\mathbb{Z}) = \{x \mid x \text{ is a subset of } \mathbb{Z}\}$ = the set of all subsets of **Z** (x⊆**Z**)

-2 weird set theorem

For every set $S$, $(i)$ $\emptyset \subseteq S$ and $(ii)$ $S \subseteq S$.

-i. Vacuous proof

-ii. S ≡ S, so subset ez

-Restricting domain of quantifier shorthand

Short hand:

$\forall x \in \mathbb{Z}$ ( )

-Programming & Set (Bit Vector)

-You represent a set with a bit array. The “1” bit at the n index means the element corresponding to that index is in the set.

-Multiset

-$\{v_1 \cdot d_1, v_2 \cdot d_2, \dots\}$

-The v_1 value is repeated d_1 amount of times.

-{1,1,1} = {1} because repeat is the same value of element.

**Cartesian Product**

-An ordered n-tuple

-$(a_1, a_2, \dots, a_n)$

-is a collection where a_1 is the first element, a_2 is the second, …, and a_n is the nth element.

-The Cartesian product of sets A and B

Let $A$ and $B$ be sets. The *Cartesian product* of $A$ and $B$, denoted by $A \times B$, is the set of all ordered pairs $(a, b)$, where $a \in A$ and $b \in B$. Hence,

$$A \times B = \{(a, b) \mid a \in A \wedge b \in B\}.$$

The *Cartesian product* of the sets $A_1, A_2, \dots, A_n$, denoted by $A_1 \times A_2 \times \cdots \times A_n$, is the set of ordered $n$-tuples $(a_1, a_2, \dots, a_n)$, where $a_i$ belongs to $A_i$ for $i = 1, 2, \dots, n$. In other words,

$$A_1 \times A_2 \times \cdots \times A_n = \{(a_1, a_2, \dots, a_n) \mid a_i \in A_i \text{ for } i = 1, 2, \dots, n\}.$$

$A \times B = \{(a,b) \mid a \in A \text{ and } b \in B\}$

(a,b): all objects of this form; a∈A and b∈B: that satisfy this condition or predicate

-is the set of all ordered pairs (a,b), where a∈A & b∈B

-The cardinality of AxB (aka |AxB|) is the amount of elements in A * the amount of elements in B.

-Ex:

Let $A = \{s, t, u\}$

$B = \{0, 1\}$

$A \times B = \{(s,0), (s,1), (t,0), (t,1), (u,0), (u,1)\}$

-$A \times \emptyset = \emptyset$

-$A \times B = B \times A$ is true only when A=B, false when A⊆B (since ordering matters), false when B is Ø, false when A={0} and B={1}.

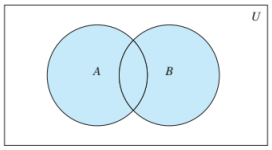
**Set Operators**

-Union: A ∪ B

-A set that contains elements that are in either A or B.

Let $A$ and $B$ be sets. The *union* of the sets $A$ and $B$, denoted by $A \cup B$, is the set that contains those elements that are either in $A$ or in $B$, or in both.

$$A \cup B = \{x \mid x \in A \vee x \in B\}$$



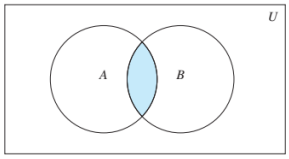
$A \cup B$ is shaded.

-Intersection: A ∩ B

-A set that contains elements that are in both A and B

Let $A$ and $B$ be sets. The *intersection* of the sets $A$ and $B$, denoted by $A \cap B$, is the set containing those elements in both $A$ and $B$.

$$A \cap B = \{x \mid x \in A \wedge x \in B\}$$



$A \cap B$ is shaded.

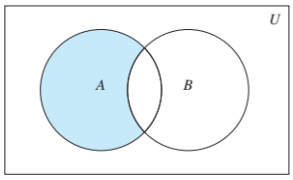
-Disjoint is when A ∩ B = null set. (aka no similar elements between the 2)

-Set Difference: A - B

-A set that contains elements that are in A but not in B

Let $A$ and $B$ be sets. The *difference* of $A$ and $B$, denoted by $A - B$, is the set containing those elements that are in $A$ but not in $B$. The difference of $A$ and $B$ is also called the *complement of $B$ with respect to $A$*.

$$A - B = \{x \mid x \in A \wedge x \notin B\}$$



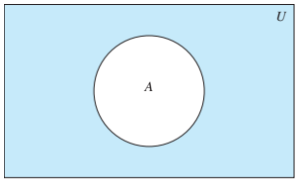
$A - B$ is shaded.

-Complement: $\overline{A}$ (alternatively, A')

-Given a universal set U, the complement of A, denoted as $\overline{A}$, is (U - A).

Let $U$ be the universal set. The *complement* of the set $A$, denoted by $A$, is the complement of $A$ with respect to $U$. Therefore, the complement of the set $A$ is $U - A$.

$$\overline{A} = \{x \in U \mid x \notin A\}$$



$\overline{A}$ is shaded.

**Set Identities (Fat)**

| Identity | Name |
|---|---|
| $A \cap U = A$<br>$A \cup \emptyset = A$ | Identity laws |
| $A \cup U = U$<br>$A \cap \emptyset = \emptyset$ | Domination laws |
| $A \cup A = A$<br>$A \cap A = A$ | Idempotent laws |
| $\overline{(\overline{A})} = A$ | Complementation law |
| $A \cup B = B \cup A$<br>$A \cap B = B \cap A$ | Commutative laws |
| $A \cup (B \cup C) = (A \cup B) \cup C$<br>$A \cap (B \cap C) = (A \cap B) \cap C$ | Associative laws |
| $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$<br>$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ | Distributive laws |
| $\overline{A \cap B} = \overline{A} \cup \overline{B}$<br>$\overline{A \cup B} = \overline{A} \cap \overline{B}$ | De Morgan's laws |
| $A \cup (A \cap B) = A$<br>$A \cap (A \cup B) = A$ | Absorption laws |
| $A \cup \overline{A} = U$<br>$A \cap \overline{A} = \emptyset$ | Complement laws |

**TABLE 2** A Membership Table for the Distributive Property.

| $A$ | $B$ | $C$ | $B \cup C$ | $A \cap (B \cup C)$ | $A \cap B$ | $A \cap C$ | $(A \cap B) \cup (A \cap C)$ |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

-Prove that $\overline{A \cup B} = \overline{A} \cap \overline{B}$

-Claim: For any sets A and B, $\overline{A \cup B} = \overline{A} \cap \overline{B}$

-Proof: We show $\overline{A \cup B} \subseteq \overline{A} \cap \overline{B}$ and $\overline{A} \cap \overline{B} \subseteq \overline{A \cup B}$

-We first show that $\overline{A \cup B} \subseteq \overline{A} \cap \overline{B}$

-Suppose x∈ $\overline{A \cup B}$

-So x∉ $A \cup B$ by def of complement.

-This means that is it not true that x∈ $A \cup B$.

-So, it is not true that x∈A or x∈B by def of union.

-That is ¬((x∈A) ∨ (x∈B)) is true.

-By De Morgan's ¬(x∈A) ∨ ¬(x∈B) ≡ ((x∉A) ∧ (x∉B))

-So, x∉A and x∉B, (def of negation)

-meaning $x \in \overline{A}$ and $x \in \overline{B}$

-By def of intersection $x \in \overline{A} \cap \overline{B}$

-Therefore $\overline{A} \cap \overline{B} \subseteq A \cup B$

-Secondly, we show that $\overline{A} \cap \overline{B} \subseteq \overline{A \cup B}$

-Bruh/

**Set Generalize**

$$\bigcup_{i=1}^{n} A_i = A_1 \cup A_2 \cup A_3 \cup \ldots \cup A_n$$

$$\bigcap_{i=1}^{n} A_i = A_1 \cap A_2 \cap A_3 \cap \ldots \cap A_n$$

-Ex:

$A_i = \{1, 2, 3, \ldots, i\}$ for $i = 1, 2, 3, \ldots, i$

$$\bigcup_{i=1}^{n} A_i = \{1\} \cup \{1, 2\} \cup \{1, 2, 3\} \cup \ldots \cup \{1, 2, 3, \ldots, n\} = \{1, 2, 3, \ldots, n\}$$

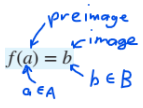
$$\bigcap_{i=1}^{n} A_i = \{1\} \cap \{1, 2\} \cap \{1, 2, 3\} \cap \ldots \cap \{1, 2, 3, \ldots, n\} = \{1\}$$

**Function Sequences**

-Wat the heo is the function?

Let $A$ and $B$ be nonempty sets. A *function* $f$ from $A$ to $B$ is an assignment of exactly one element of $B$ to each element of $A$. We write $f(a) = b$ if $b$ is the unique element of $B$ assigned by the function $f$ to the element $a$ of $A$. If $f$ is a function from $A$ to $B$, we write $f : A \to B$.

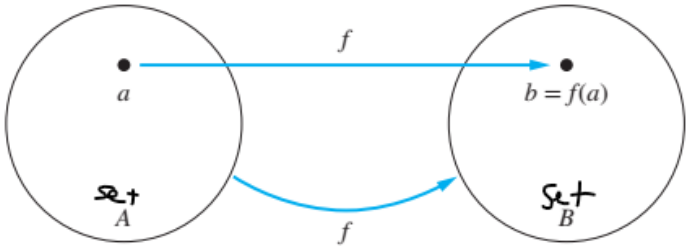
If $f$ is a function from $A$ to $B$, we say that $A$ is the *domain* of $f$ and $B$ is the *codomain* of $f$. If $f(a) = b$, we say that $b$ is the *image* of $a$ and $a$ is a *preimage* of $b$. The *range*, or *image*, of $f$ is the set of all images of elements of $A$. Also, if $f$ is a function from $A$ to $B$, we say that $f$ *maps* $A$ to $B$.

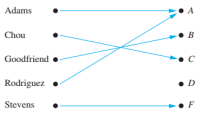



-f: A → B

-a is mapped to b. A is the domain, B is the codomain

-The range of a function is the set of codomain elements that actually got used / mapped to.




-like a method (function) that takes in an int argument (domain) and returns an int (codomain).

int mystery1 (int x) {...}  mystery1: $\mathbb{Z} \to \mathbb{Z}$

boolean mystery2 (int x, double y) {...}

mystery2: $\mathbb{Z} \times \mathbb{R} \to \{T, F\}$

where the reals is approximated by doubles

-Function addition with real numbers

Let $f_1$ and $f_2$ be functions from $A$ to $\mathbf{R}$. Then $f_1 + f_2$ and $f_1 f_2$ are also functions from $A$ to $\mathbf{R}$ defined for all $x \in A$ by

$$(f_1 + f_2)(x) = f_1(x) + f_2(x),$$
$$(f_1 f_2)(x) = f_1(x) f_2(x).$$

Let $f_1$ and $f_2$ be functions from $\mathbf{R}$ to $\mathbf{R}$ such that $f_1(x) = x^2$ and $f_2(x) = x - x^2$. What are the functions $f_1 + f_2$ and $f_1 f_2$?

*Solution:* From the definition of the sum and product of functions, it follows that

$$(f_1 + f_2)(x) = f_1(x) + f_2(x) = x^2 + (x - x^2) = x$$

and

$$(f_1 f_2)(x) = x^2(x - x^2) = x^3 - x^4.$$ ◀

-Image of a subset of A in f:A->B

Let $f$ be a function from $A$ to $B$ and let $S$ be a subset of $A$. The *image* of $S$ under the function $f$ is the subset of $B$ that consists of the images of the elements of $S$. We denote the image of $S$ by $f(S)$, so

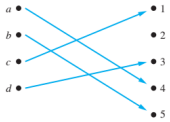
$$f(S) = \{t \mid \exists s \in S\, (t = f(s))\}.$$

We also use the shorthand $\{f(s) \mid s \in S\}$ to denote this set.

Let $A = \{a, b, c, d, e\}$ and $B = \{1, 2, 3, 4\}$ with $f(a) = 2, f(b) = 1, f(c) = 4, f(d) = 1$, and $f(e) = 1$. The image of the subset $S = \{b, c, d\}$ is the set $f(S) = \{1, 4\}$. ◀

**One-to-One Functions**

-Functions that never assign the same value to 2 domain elements. (aka no repeats)




A function $f$ is said to be *one-to-one*, or an *injection*, if and only if $f(a) = f(b)$ implies that $a = b$ for all $a$ and $b$ in the domain of $f$. A function is said to be *injective* if it is one-to-one.

$\forall a \forall b (f(a) = f(b) \to a = b)$ OR $\forall a \forall b (a \neq b \to f(a) \neq f(b))$

-Increasing & Decreasing

A function $f$ whose domain and codomain are subsets of the set of real numbers is called *increasing* if $f(x) \leq f(y)$, and *strictly increasing* if $f(x) < f(y)$, whenever $x < y$ and $x$ and $y$ are in the domain of $f$. Similarly, $f$ is called *decreasing* if $f(x) \geq f(y)$, and *strictly decreasing* if $f(x) > f(y)$, whenever $x < y$ and $x$ and $y$ are in the domain of $f$. (The word *strictly* in this definition indicates a strict inequality.)

for all $x, y \in A$, if $x < y$, then $f(x) \leq f(y)$. increasing (nondecreasing)

for all $x, y \in A$, if $x < y$, then $f(x) < f(y)$ strictly increasing

for all $x, y \in A$, if $x < y$, then $f(x) \geq f(y)$ decreasing (nonincreasing)

for all $x, y \in A$, if $x < y$, then $f(x) > f(y)$ strictly decreasing

-increasing $\forall x \forall y (x < y \to f(x) \leq f(y))$

-strictly increasing $\forall x \forall y (x < y \to f(x) < f(y))$

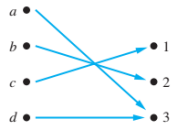
-decreasing $\forall x \forall y (x < y \to f(x) \geq f(y))$

-strictly decreasing $\forall x \forall y (x < y \to f(x) > f(y))$

**Onto Function**

-A function where elements in A map to every element in B at least once.

A function $f$ from $A$ to $B$ is called *onto*, or a *surjection*, if and only if for every element $b \in B$ there is an element $a \in A$ with $f(a) = b$. A function $f$ is called *surjective* if it is onto.



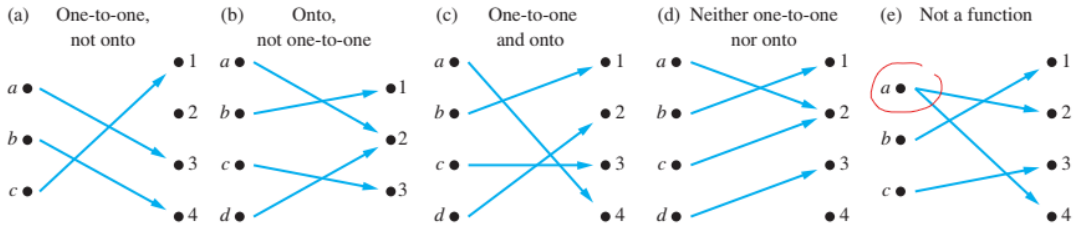

$\forall y \exists x (f(x) = y)$

-Correspondence/Bijection

The function $f$ is a *one-to-one correspondence*, or a *bijection*, if it is both one-to-one and onto. We also say that such a function is *bijective*.

-(aka) Every element in B is mapped to once and only once.

-Onto & 1-to-1 relationship




Suppose that $f : A \to B$.

*To show that $f$ is injective* Show that if $f(x) = f(y)$ for arbitrary $x, y \in A$, then $x = y$.

*To show that $f$ is not injective* Find particular elements $x, y \in A$ such that $x \neq y$ and $f(x) = f(y)$.

*To show that $f$ is surjective* Consider an arbitrary element $y \in B$ and find an element $x \in A$ such that $f(x) = y$.

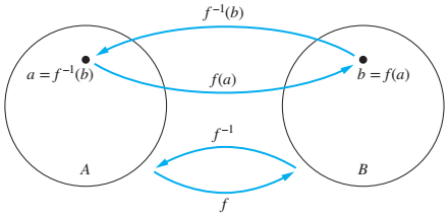
*To show that $f$ is not surjective* Find a particular $y \in B$ such that $f(x) \neq y$ for all $x \in A$.

**Inverse Functions and Compositions of Functions**

-An inverse function maps elements of B to elements of A (so it's opposite the direction)

Let $f$ be a one-to-one correspondence from the set $A$ to the set $B$. The *inverse function* of $f$ is the function that assigns to an element $b$ belonging to $B$ the unique element $a$ in $A$ such that $f(a) = b$. The inverse function of $f$ is denoted by $f^{-1}$. Hence, $f^{-1}(b) = a$ when $f(a) = b$.

-Only 1-to-1 correspondence is invertible. Everything else is invertible.




Let $f$ be the function from $\{a, b, c\}$ to $\{1, 2, 3\}$ such that $f(a) = 2$, $f(b) = 3$, and $f(c) = 1$. Is $f$ invertible, and if it is, what is its inverse?

*Solution:* The function $f$ is invertible because it is a one-to-one correspondence. The inverse function $f^{-1}$ reverses the correspondence given by $f$, so $f^{-1}(1) = c$, $f^{-1}(2) = a$, and $f^{-1}(3) = b$. ◀

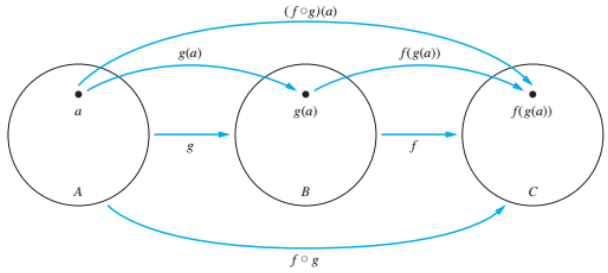
**Composition**

-In other words, f ∘g is the function that assigns to the element a of A the element assigned by f to g(a).

Let $g$ be a function from the set $A$ to the set $B$ and let $f$ be a function from the set $B$ to the set $C$. The *composition* of the functions $f$ and $g$, denoted for all $a \in A$ by $f \circ g$, is the function from $A$ to $C$ defined by

$$(f \circ g)(a) = f(g(a)).$$

(So u do the g() then f())




-You can have a composition of functions that convert to different domains, only if g()'s codomain is the same as f() domain.

**Graphing Functions**

-Just like normal math, assign sets domain axis.

Let $f$ be a function from the set $A$ to the set $B$. The *graph* of the function $f$ is the set of ordered pairs $\{(a, b) \mid a \in A \text{ and } f(a) = b\}$.

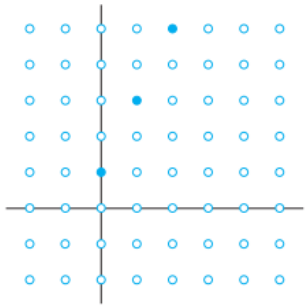


**FIGURE 8** The graph of $f(n) = 2n + 1$ from Z to Z.

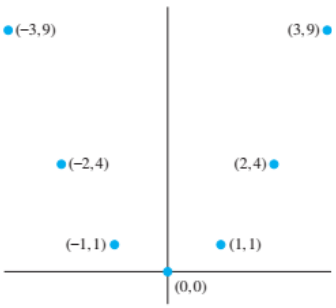



**FIGURE 9** The graph of $f(x) = x^2$ from Z to Z.

-Floor and Ceiling

-kms, it's just a function that rounds down (floor) or up (ceiling)

The *floor function* assigns to the real number $x$ the largest integer that is less than or equal to $x$. The value of the floor function at $x$ is denoted by $\lfloor x \rfloor$. The *ceiling function* assigns to the real number $x$ the smallest integer that is greater than or equal to $x$. The value of the ceiling function at $x$ is denoted by $\lceil x \rceil$.

$\lfloor 2.5 \rfloor = 2 \quad \lfloor 2.1 \rfloor = 2 \quad \lfloor -1.3 \rfloor = -2$

$\lceil 2.5 \rceil = 3 \quad \lceil 2.1 \rceil = 3 \quad \lceil -1.3 \rceil = -1$

$\lfloor 7 \rfloor = \lceil 7 \rceil = 7$

($n$ is an integer, $x$ is a real number)

(1a) $\lfloor x \rfloor = n$ if and only if $n \le x < n + 1$
(1b) $\lceil x \rceil = n$ if and only if $n - 1 < x \le n$
(1c) $\lfloor x \rfloor = n$ if and only if $x - 1 < n \le x$
(1d) $\lceil x \rceil = n$ if and only if $x \le n < x + 1$

(2) $x - 1 < \lfloor x \rfloor \le x \le \lceil x \rceil < x + 1$

(3a) $\lfloor -x \rfloor = -\lceil x \rceil$
(3b) $\lceil -x \rceil = -\lfloor x \rfloor$

(4a) $\lfloor x + n \rfloor = \lfloor x \rfloor + n$
(4b) $\lceil x + n \rceil = \lceil x \rceil + n$

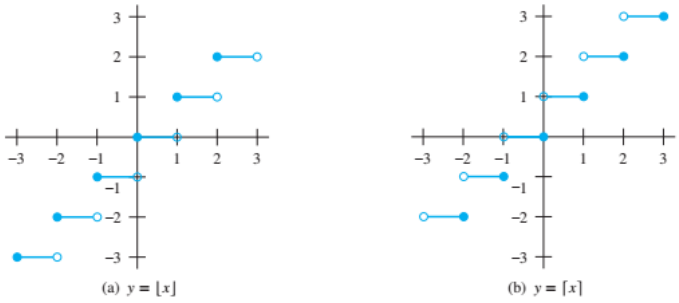



**FIGURE 10** Graphs of the (a) floor and (b) ceiling functions.

-Absolute Value

Absolute value

$|x| = abs(x) = \begin{cases} x & \text{if } x \ge 0 \\ -x & \text{if } x < 0 \end{cases}$ for all $x \in \mathbb{R}$, $\lceil |x| \rceil = |\lceil x \rceil|$ is false.

-Factorial function

$f: \mathbb{N} \to \mathbb{Z}^+$, denoted $f(n) = n!$

$f(n)$ is the product of the first $n$ positive integers.

-Recursive def

$$n! = \begin{cases} 1, & \text{if } n = 1 \\ n \cdot (n-1)!, & \text{if } n > 1 \end{cases}$$

-Stirling's Approx

$$n! \approx \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$$

-Partial Functions

-If it's not a total function that can map all elements in the domain, it's partial. (no special notation, just know it is 4head.)

A *partial function* $f$ from a set $A$ to a set $B$ is an assignment to each element $a$ in a subset of $A$, called the *domain of definition* of $f$, of a unique element $b$ in $B$. The sets $A$ and $B$ are called the *domain* and *codomain* of $f$, respectively. We say that $f$ is *undefined* for elements in $A$ that are not in the domain of definition of $f$. When the domain of definition of $f$ equals $A$, we say that $f$ is a *total function*.

**Sequence**

-A sequence is a discrete structure used to represent an ordered list.

A *sequence* is a function from a subset of the set of integers (usually either the set $\{0, 1, 2, \ldots\}$ or the set $\{1, 2, 3, \ldots\}$) to a set $S$. We use the notation $a_n$ to denote the image of the integer $n$. We call $a_n$ a *term* of the sequence.

- $\{a_n\}$ is how you denote the sequence of those integers.

- $a_n = n$ is $a_1, a_2, a_3, \ldots$ i.e. $1, 2, 3, 4, \ldots$
- $a_n = \frac{1}{n}$ i.e. $1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots$
- $a_n = 2n + 3$ i.e. $5, 7, 9, 11, \ldots$

-Geometric Progression Sequence

-A sequence with a common ratio between terms.

A *geometric progression* is a sequence of the form

$$a, ar, ar^2, \ldots, ar^n, \ldots$$

where the *initial term* $a$ and the *common ratio* $r$ are real numbers. $f(x) = ar^x$

- $a = 1, r = 5$: $1, 5, 25, 125, \ldots$
- $a = 7, r = -1$: $f(n) = 7(-1)^n$ $7, -7, 7, -7, \ldots$
- $a = 2, r = \frac{1}{3}$: $2, \frac{2}{3}, \frac{2}{9}, \frac{2}{27}, \ldots$

-So if you say that it's a geometric sequence, you can just specify what a & r is.

-Arithmetic Progression Sequence

-A sequence with a common difference between terms.

An *arithmetic progression* is a sequence of the form

$$a, a + d, a + 2d, \ldots, a + nd, \ldots$$

where the *initial term* $a$ and the *common difference* $d$ are real numbers. $f(x) = dx + a$

-Same as geometric, just specify what a & d is.

-String

-Another name for finite sets

Sequences of the form $a_1, a_2, \ldots, a_n$ are often used in computer science. These finite sequences are also called **strings**. This string is also denoted by $a_1a_2 \ldots a_n$. (Recall that bit strings, which are finite sequences of bits, were introduced in Section 1.1.) The **length** of a string is the number of terms in this string. The **empty string**, denoted by $\lambda$, is the string that has no terms. The empty string has length zero.

**Recurrence Relations/Equations**

-A sequence that uses previous terms to define its next.

A *recurrence relation* for the sequence $\{a_n\}$ is an equation that expresses $a_n$ in terms of one or more of the previous terms of the sequence, namely, $a_0, a_1, \ldots, a_{n-1}$, for all integers $n$ with $n \geq n_0$, where $n_0$ is a nonnegative integer. A sequence is called a *solution* of a recurrence relation if its terms satisfy the recurrence relation. (A recurrence relation is said to *recursively define* a sequence. We will explain this alternative terminology in Chapter 5.)

-That means there's an initial condition to start from.

example: $a_n = a_{n-1} + 5$ → 3, 8, 13, 18, 23, ...

$a_0 = 9$ ← initial condition → −14, −9, −4, 1, 6, 11, ...

9, 14, 19, 24, ...

example: $a_n = a_{n-1} + a_{n-2}$ Fibonacci sequence

$a_0 = 0,\ a_1 = 1$ ← initial conditions

-Like in the Fibonacci sequence, the number of previous terms needed to define the next must be matched by the amount of initial conditions

-Solving A Recurrence Sequence

-Basically, you have to remove the a_n and make it a function f(n)= the same output as recurrence.

-Iteration method: Repeat the substitution, find the pattern with n++, then choose k= whatever that gets to init condition and boom.

① $a_n = 2a_{n-1}$
$= 2 \cdot 2a_{n-2}$
$= 2 \cdot 2 \cdot 2a_{n-3}$
$\vdots$
$= 2^k a_{n-k}$
$= 2^n a_{n-n}$ (let $k=n$)
$= 2^n a_0$
$= 2^n$

② $a_n = 2a_{n-1} + 1$
$= 2(2a_{n-2}+1)+1$
$= 2 \cdot 2(2a_{n-3}+1)+2+1$
$= 2 \cdot 2 \cdot 2a_{n-3} + 4 + 2 + 1$
$\vdots$
$= 2^k a_{n-k} + (2^k - 1)$
$= 2^n a_{n-n} + (2^n - 1)$ (let $k=n$)
$= 2^n a_0 + (2^n - 1)$
$= 2^n + (2^n - 1)$
$= 2^{n+1} - 1$
1, 3, 7, 15,

(in #2, group the multiplication into power for clarity)

**Summations**

-Add up all them terms! Denoted by greek sugma.

$a_m, a_{m+1}, \ldots, a_n$

from the sequence $\{a_n\}$. We use the notation

$\sum_{j=m}^{n} a_j$, $\sum_{j=m}^{n} a_j$, or $\sum_{m \leq j \leq n} a_j$

(read as the sum from $j = m$ to $j = n$ of $a_j$) to represent

$a_m + a_{m+1} + \cdots + a_n$.

upper limit → $\sum_{j=m}^{n} a_j$ ← lower limit

index of summation

or $\sum_{m \leq j \leq n} a_j$ or $\sum_{j \in \{m, m+1, \ldots, n\}} a_j$

-j can be changed to whatever

$$\sum_{j=m}^{n} a_j = \sum_{i=m}^{n} a_i = \sum_{k=m}^{n} a_k$$

-Ex:

$\sum_{i=1}^{6} i = 1+2+3+4+5+6 = 21$ $\sum_{i=0}^{4} 2^i = 1+2+4+8+16 = 31$ $\sum_{i=3}^{8} (2i+3) = 9+11+13+15+17+19 = 84$

-Some Facts & Theorems (scroll down for the big list)

Suppose $b$ does not depend on $j$.

- $\sum_{j=m}^{n} b\,a_j = b \sum_{j=m}^{n} a_j$

$$\sum_{j=m}^{n} b\,a_j = b\,a_m + b\,a_{m+1} + b\,a_{m+2} + \dots + b\,a_n$$
$$= b\,(a_m + a_{m+1} + a_{m+2} + \dots + a_n)$$
$$= b \sum_{j=m}^{n} a_j$$

$$\sum_{j=m}^{n} (a_j + b_j) = \sum_{j=m}^{n} a_j + \sum_{j=m}^{n} b_j$$

If $a$ and $r$ are real numbers and $r \neq 0$, then

$$\sum_{j=0}^{n} ar^j = \begin{cases} \dfrac{ar^{n+1} - a}{r - 1} & \text{if } r \neq 1 \\ (n+1)a & \text{if } r = 1. \end{cases}$$

-Proof for above theorem

***Proof:*** Let

$$S_n = \sum_{j=0}^{n} ar^j.$$

To compute $S$, first multiply both sides of the equality by $r$ and then manipulate the resulting sum as follows:

$$rS_n = r \sum_{j=0}^{n} ar^j \qquad \text{substituting summation formula for } S$$
$$= \sum_{j=0}^{n} ar^{j+1} \qquad \text{by the distributive property}$$
$$= \sum_{k=1}^{n+1} ar^k \qquad \text{shifting the index of summation, with } k = j + 1$$
$$= \left( \sum_{k=0}^{n} ar^k \right) + (ar^{n+1} - a) \qquad \text{removing } k = n + 1 \text{ term and adding } k = 0 \text{ term}$$
$$= S_n + (ar^{n+1} - a) \qquad \text{substituting } S \text{ for summation formula}$$

From these equalities, we see that

$$rS_n = S_n + (ar^{n+1} - a).$$

Solving for $S_n$ shows that if $r \neq 1$, then

$$S_n = \frac{ar^{n+1} - a}{r - 1}.$$

If $r = 1$, then the $S_n = \sum_{j=0}^{n} ar^j = \sum_{j=0}^{n} a = (n+1)a$. ◁

-Change of index

-Sometimes, shifting the indices to apply some theorem

-Basically, if you reduce the indexes, you have to add the same to the summation.

$$\sum_{j=m}^{n} f(j) = f(m) + f(m+1) + \dots + f(n)$$
$$= \sum_{j=0}^{n-m} f(j+m)$$

-Ex:

$$\sum_{i=1}^{5} i = 1+2+3+4+5 = \sum_{i=0}^{4} (i+1)$$

$$\sum_{i=1}^{6} i = 1+2+3+4+5+6 = 7 \cdot 3 = 21$$

Gauss (smart mathematician!)

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

sum of each pair; # of pairs

What if $n$ is odd?

$$\sum_{j=1}^{5} j^2 = \sum_{k=0}^{4} (k+1)^2$$

-Double Summation

-Like nested loop Kad eww.

-Do the inner summation, then the outer.

-Ex:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(i+j)$$
$$= \sum_{i=1}^{n}\left(\sum_{j=1}^{n} i + \sum_{j=1}^{n} j\right)$$
$$= \sum_{i=1}^{n}\left((n-1+1)i + \frac{(n+1)n}{2}\right)$$
$$= \sum_{i=1}^{n} ni + \sum_{i=1}^{n}\frac{(n+1)n}{2}$$
$$= n\left(\sum_{i=1}^{n} i\right) + \frac{(n+1)n^2}{2}$$
$$= n\,\frac{(n+1)n}{2} + \frac{(n+1)n^2}{2}$$
$$= \frac{(n+1)n^2}{2} + \frac{(n+1)n^2}{2}$$
$$= (n+1)n^2$$

$$\sum_{i=1}^{4}\sum_{j=1}^{3} ij = \sum_{i=1}^{4}(i+2i+3i)$$
$$= \sum_{i=1}^{4} 6i$$
$$= 6+12+18+24 = 60$$

-Weird index of summations

-Index of summation is a set

$$\sum_{s\in\{0,2,4\}} s = 0+2+4 = 6.$$

-Index of summation as inequality

$$\sum_{m\le j\le n} f(j)$$ (so m is lower, n is upper)

-Fat Summation Formulae

$$\sum_{i=437}^{6318} 1 = 1+1+1+\ldots+1 = ?$$

Since each term in the summation is 1, we just need to know how many terms there are.

In general, $\sum_{i=m}^{n} f(i)$, where $f(i)$ does not depend on $i$, is just $(n-m+1)\cdot f(\cdot)$

$$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$$

If $a$ and $r$ are real numbers and $r \neq 0$, then

$$\sum_{j=0}^{n} ar^j = \begin{cases} \dfrac{ar^{n+1}-a}{r-1} & \text{if } r \neq 1 \\ (n+1)a & \text{if } r = 1. \end{cases}$$

-Ultimate Summation Identities (similar to above)

| *Sum* | *Closed Form* |
|---|---|
| $\sum_{k=0}^{n} ar^k\ (r\neq 0)$ | $\frac{ar^{n+1}-a}{r-1}, r\neq 1$ |
| $\sum_{k=1}^{n} k$ | $\frac{n(n+1)}{2}$ |
| $\sum_{k=1}^{n} k^2$ | $\frac{n(n+1)(2n+1)}{6}$ |
| $\sum_{k=1}^{n} k^3$ | $\frac{n^2(n+1)^2}{4}$ |
| $\sum_{k=0}^{\infty} x^k, \lvert x\rvert < 1$ | $\frac{1}{1-x}$ |
| $\sum_{k=1}^{\infty} kx^{k-1}, \lvert x\rvert < 1$ | $\frac{1}{(1-x)^2}$ |

**Splitting a sum**

$$\sum_{i=a}^{b}(x+y) = \sum_{i=a}^{b}x + \sum_{i=a}^{b}y$$

**Factoring out a constant**

$$\sum_{i=a}^{b} cf(i) = c\sum_{i=a}^{b} f(i)$$

**Gauss's identity**

$$\sum_{i=0}^{n-1} i = 0+1+\ldots+n-1 = \frac{n(n-1)}{2}$$

**Finite geometric series**

$$\sum_{i=0}^{n-1} x^i = \frac{x^n-1}{x-1}$$

**Adjusting summation bounds**

$$\sum_{i=a}^{b} f(x) = \sum_{i=0}^{b} f(x) - \sum_{i=0}^{a-1} f(x)$$

**Summation of a constant**

$$\sum_{i=0}^{n-1} c = \underbrace{c+c+\ldots+c}_{n \text{ times}} = cn$$

Note: this rule is a special case of the rule on the left

**Sum of squares**

$$\sum_{i=0}^{n-1} i^2 = \frac{n(n-1)(2n-1)}{6}$$

**Infinite geometric series**

$$\sum_{i=0}^{\infty} x_i = \frac{1}{1-x}$$

Note: applicable only when $-1 < x < 1$

if $|r| < 1$, $\sum_{n=1}^{\infty} a(r)^{n-1} = \frac{a}{1-r}$

**Cardinality of Sets**

-Cardinality of a finite set is the number of elements. Then what about infinite sets?

The sets $A$ and $B$ have the same *cardinality* if and only if there is a one-to-one correspondence from $A$ to $B$. When $A$ and $B$ have the same cardinality, we write $|A| = |B|$.

If there is a one-to-one function from $A$ to $B$, the cardinality of $A$ is less than or the same as the cardinality of $B$ and we write $|A| \le |B|$. Moreover, when $|A| \le |B|$ and $A$ and $B$ have different cardinality, we say that the cardinality of $A$ is less than the cardinality of $B$ and we write $|A| < |B|$.

**SCHRÖDER-BERNSTEIN THEOREM** If $A$ and $B$ are sets with $|A| \le |B|$ and $|B| \le |A|$, then $|A| = |B|$. In other words, if there are one-to-one functions $f$ from $A$ to $B$ and $g$ from $B$ to $A$, then there is a one-to-one correspondence between $A$ and $B$.

(like set equiv with subset)

-Countable, Uncountable, Finite

A set that is either finite or has the same cardinality as the set of positive integers is called *countable*. A set that is not countable is called *uncountable*. When an infinite set $S$ is countable, we denote the cardinality of $S$ by $\aleph_0$ (where $\aleph$ is aleph, the first letter of the Hebrew alphabet). We write $|S| = \aleph_0$ and say that $S$ has cardinality "aleph null."

If $A$ and $B$ are countable sets, then $A \cup B$ is also countable.

finite set  $S$ has $n$ elements, where $n \in \mathbb{Z}$.
If $S$ has $n$ elements, then $|S| = n$.

countable set  $S$ is either finite or has the same cardinality as $\mathbb{Z}$.
example: the even integers

uncountable set  $S$ is not countable.
example: $\mathcal{P}(\mathbb{Z})$ is uncountable.

-The Continuum Hypothesis

$$|\mathcal{P}(\mathbf{Z}^+)| = |\mathbf{R}|$$

**Algorithms**

-Step by step sequence of instruction to do something

An *algorithm* is a finite sequence of precise instructions for performing a computation or for solving a problem.

-Pseudocode Ex

Input: An array $A[1..n]$ of $n$ integers
Output: The maximum element in $A[1..n]$
(computational problem description)

```
maximum(A[1..n])
  max ← A[1]
  for i ← 2 to n do
    if max < A[i] then
      max ← A[i]
  return max
```

pseudocode: not the same as Java or C, but very similar

Input: An array $A[1..n]$ of $n$ integers
Output: The sum of all the elements in $A$

```
sum(A[1..n])
  total ← A[1]
  for i ← 2 to n do
    total ← total + A[i]
  return total
```

Input: An array $A[1..n]$ of $n$ integers and integer key.
Output: Smallest index $i$ such that $A[i] = key$, or $-1$ if there is no such $i$.

```
find(A[1..n], key)
  i ← 1
  while (i ≤ n and A[i] ≠ key) do
    i ← i+1
  if i ≤ n then
    location ← i
  else
    location ← -1
  return location
```

indentation matters!

-Selection Sort and using Summation to find time complexity

Input: An array $A[1..n]$ of real numbers, where $n \geq 2$
Output: $A[1..n]$ with the original elements of $A$ in sorted (nondecreasing) order.
(the sorting problem)

```
selectionSort(A[1..n])
  for i ← 1 to n-1 do
    // find min of A[i..n]
    minIndex ← i
    for j ← i+1 to n do
      if A[j] < A[minIndex]
        minIndex ← j
    // swap A[i] with A[minIndex]
    temp ← A[i]
    A[i] ← A[minIndex]
    A[minIndex] ← temp
```

number of times this line is executed:

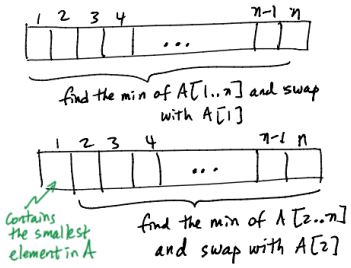
$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} 1$$

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

$$= \sum_{i=1}^{n-1} (n-i)$$

$$= \sum_{i=1}^{n-1} n - \sum_{i=1}^{n-1} i$$

$$=$$




As a function of $n$, how long would it take to run selectionSort on an array of $n$ elements?

-The summation = ½(n^2 - n), dominating the time complexity (so you can disregard the small one off statements), making it quadratic in complexity.

**Big O**

Let $f$ and $g$ be functions from the set of integers or the set of real numbers to the set of real numbers. We say that $f(x)$ is $O(g(x))$ if there are constants $C$ and $k$ such that

$$|f(x)| \leq C|g(x)|$$

whenever $x > k$. [This is read as "$f(x)$ is big-oh of $g(x)$."]

***Remark:*** Intuitively, the definition that $f(x)$ is $O(g(x))$ says that $f(x)$ grows slower than some fixed multiple of $g(x)$ as $x$ grows without bound.

-The x>k means that we only care for when f(x) is always consistently bigger than g(x).
-k & c are called the witnesses or certificates.
-Usually, c>0 because negative numbers are weird.

Let $f(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$, where $a_0, a_1, \ldots, a_{n-1}, a_n$ are real numbers. Then $f(x)$ is $O(x^n)$.

-f(x) is always O(f(x) highest variable term) (ie x^2 + 2x +1 is O(x^2)).

-Ex:

-f(n) =n & g(n)=½n
-C = 2, making fx)<=g(x).
-k = 0, because f(x)<= g(x) when x=0
-Therefore n is O(½n)

$f(n) = n$, $g(n) = 2n$

Let $c = 1$, $k = 0$.
$f(n) = n \leq 1 \cdot 2n = \underline{1} \cdot g(n)$, for all $n > \underline{0}$.
$\therefore$ $n$ is $O(2n)$.

$f(n) = 2n$, $g(n) = n$

Let $c = 2$, $k = 0$.
$f(n) = 2n = 2 \cdot n = \underline{2} \cdot g(n)$, for all $n > \underline{0}$.
$\therefore$ $2n$ is $O(n)$.

$f(n) = 10n$, $g(n) = n^2$

Let $c = 10$, $k = 1$.
$f(n) = 10n \leq 10 \cdot n^2 = 10 \cdot g(n)$,
for all $n > 1$.

OR
Let $c = 1$, $k = 10$.
$f(n) = \underline{10n \leq 1 \cdot n \cdot n} = \underline{1} \cdot g(n)$
why?
for all $n > \underline{10}$.

$\therefore$ $10n$ is $O(n^2)$.

$f(n) = n$, $g(n) = n^2$

Let $c = 1$, $k = 1$.
$f(n) = \underline{n \leq 1 \cdot n \cdot n} = \underline{1} \cdot g(n)$,
why?
for all $n > \underline{1}$.

$\therefore$ $n$ is $O(n^2)$.

-Quadratic Example

$f(n) = n^2 + 2n + 1$

$g(n) = n^2$

$n^2 + 2n + 1 \leq n^2 + 2n^2 + n^2$ (for $n > 1$)
$= 4n^2$

So, for $n > 1$,
$f(n) = n^2 + 2n + 1 \leq 4n^2 = \underline{4} \cdot g(n)$, for all $n > \underline{1}$
(4 ← c, 1 ← k)

$\therefore$ $f(n)$ is $O(g(n))$

-4n^2 came from making all other coefficients to x^2, making a function that is greater than the original f(n).

-The Chinn Way
-Find a function that is greater than f(x).
-The function must have each order of term be greater than the original.
-Keep going or reduce the new function to find a c.
-The vertical shift of f(x) will probably work as k.

-Limit Test
-Another way to prove f(x) is O(g(x)) is taking the limit:

$$\lim_{x \to \infty} \frac{f(x)}{g(x)}$$

-if =0, f(n) is o(g(n))
-if =inf, f(n) is o(g(n))
-if ≠0, f(n) is Θ(g(n))

-Not Big Oh
-Then there isn't c or k that satisfy Big Oh.
-Ex:

- Show that $n^2$ is not $O(n)$.

Proof: We need to show that there is no $c$ and $k$ such that $n^2 \leq c \cdot n$ for all $n > k$.

We show this using proof by contradiction.

Suppose there is a $c$ and $k$ such that $n^2 \leq c \cdot n$ for all $n > k$.

Then $n \leq c$ for all $n > k$. (why?)

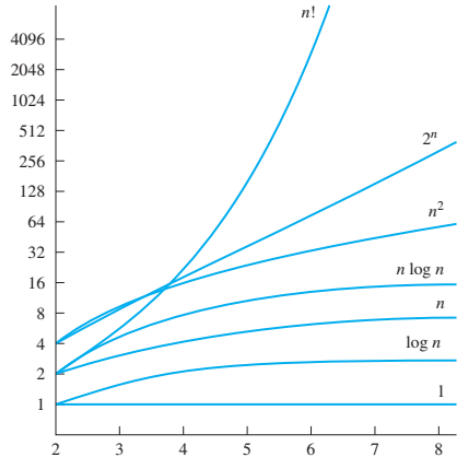
But we can always find an $n > k$ where $n > c$.

For example, $n = \max(c, k) + 1$.

This contradicts the fact that $n \leq c$ for all $n > k$.

So, $n^2$ is not $O(n)$.

-Common functions for big O comparison




-Proving 2^n is not O(n)

-Consider both functions as sequences of the output.

-Show that

-The term at the same n is greater than 2^n (which is not right).

-The ratio at the same n and n-1 is greater in n than 2^n (which is not right).

-Def and Theorems

-Two functions and relationship with O()

Suppose that $f_1(x)$ and $f_2(x)$ are both $O(g(x))$. Then $(f_1 + f_2)(x)$ is $O(g(x))$.

Suppose that $f_1(x)$ is $O(g_1(x))$ and that $f_2(x)$ is $O(g_2(x))$. Then $(f_1 + f_2)(x)$ is $O(g(x))$, where $g(x) = (\max(|g_1(x)|, |g_2(x)|)$ for all $x$.

Suppose that $f_1(x)$ is $O(g_1(x))$ and $f_2(x)$ is $O(g_2(x))$. Then $(f_1 f_2)(x)$ is $O(g_1(x)g_2(x))$.

-Big Omega(lul)

-Big O is upper bound or always greater, Omega is lower bound or always lower.

Let $f$ and $g$ be functions from the set of integers or the set of real numbers to the set of real numbers. We say that $f(x)$ is $\Omega(g(x))$ if there are constants $C$ and $k$ with $C$ positive such that

$$|f(x)| \geq C|g(x)|$$

whenever $x > k$. [This is read as "$f(x)$ is big-Omega of $g(x)$."]

-Omega iff Big O

There is a strong connection between big-$O$ and big-Omega notation. In particular, $f(x)$ is $\Omega(g(x))$ if and only if $g(x)$ is $O(f(x))$. We leave the verification of this fact as a straightforward exercise for the reader.

-Proof:

| | | |
|---|---|---|
| 1 | $f(n) = O(g(n))$ | suppose |
| 2 | $f(n) \leq cg(n),\ n > k,\ c > 0$ | defn O where $c$&$k$ are constants |
| 3 | $\frac{1}{c} f(n) \leq g(n),\ n > k,\ c > 0$ | $\div c$ both sides |
| 4 | $g(n) \geq \frac{1}{c} f(n),\ n > k,\ c > 0$ | flip sides |
| 5 | $g(n) \geq c' f(n),\ n > k',\ c > 0$ | let $c' = \frac{1}{c}$ & $k' = k$ |
| 6 | $g(n) = \Omega(f(n))$ | suppose |
| 7 | $g(n) \geq cf(n),\ n > k,\ c > 0$ | defn $\Omega$ where $c$&$k$ are constants |
| 8 | $f(n) \leq \frac{1}{c} g(n),\ n > k,\ c > 0$ | $\div c$ both sides |
| 9 | $f(n) \leq c' g(n),\ n > k',\ c > 0$ | let $c' = \frac{1}{c}$ & $k' = k$ |
| $\therefore$ | $f(n) = O(g(n)) \leftrightarrow g(n) = \Omega(f(n))$ | 5&9 |

-Big Theta

-If both O and Omega

Let $f$ and $g$ be functions from the set of integers or the set of real numbers to the set of real numbers. We say that $f(x)$ is $\Theta(g(x))$ if $f(x)$ is $O(g(x))$ and $f(x)$ is $\Omega(g(x))$. When $f(x)$ is $\Theta(g(x))$, we say that $f$ is big-Theta of $g(x)$, that $f(x)$ is of *order* $g(x)$, and that $f(x)$ and $g(x)$ are of the *same order*.

**Divisibility**

-Division

If $a$ and $b$ are integers with $a \neq 0$, we say that $a$ *divides* $b$ if there is an integer $c$ such that $b = ac$ (or equivalently, if $\frac{b}{a}$ is an integer). When $a$ divides $b$ we say that $a$ is a *factor* or *divisor* of $b$, and that $b$ is a *multiple* of $a$. The notation $a \mid b$ denotes that $a$ divides $b$. We write $a \nmid b$ when $a$ does not divide $b$.

***Remark:*** We can express $a \mid b$ using quantifiers as $\exists c(ac = b)$, where the universe of discourse is the set of integers.

-Ex: 7/3 is 3∤7, 12/4 is 4|12

-3 ints relationship theorem

Let $a$, $b$, and $c$ be integers, where $a \neq 0$. Then

(*i*) if $a \mid b$ and $a \mid c$, then $a \mid (b + c)$;
(*ii*) if $a \mid b$, then $a \mid bc$ for all integers $c$;
(*iii*) if $a \mid b$ and $b \mid c$, then $a \mid c$.

If $a$, $b$, and $c$ are integers, where $a \neq 0$, such that $a \mid b$ and $a \mid c$, then $a \mid mb + nc$ whenever $m$ and $n$ are integers.

-Proof for (i)

| | | |
|---|---|---|
| 1 | $a \mid b$ and $a \mid c$ | |
| 2 | $b = za$ and $c = na$ | defn, for some ints $a, n$ |
| 3 | $b + c = za + na$ | |
| 4 | $b + c = a(z + n)$ | |
| 5 | $b + c = ta$ | let $t = z + n$ |
| $\therefore$ | $a \mid (b + c)$ | defn |

-The Division Algorithm Theorem thing.

-If a divides by d, there is a unique q and r that a=dq+r. r is the remainder.

-a is called dividend

**THE DIVISION ALGORITHM** Let $a$ be an integer and $d$ a positive integer. Then there are unique integers $q$ and $r$, with $0 \leq r < d$, such that $a = dq + r$.

-d is called divisor

-q is called quotient

-r is called remainder

-div & mod

-div outputs the quotient, mod outputs the remainder.

In the equality given in the division algorithm, $d$ is called the *divisor*, $a$ is called the *dividend*, $q$ is called the *quotient*, and $r$ is called the *remainder*. This notation is used to express the quotient and remainder:

$$q = a \textbf{ div } d, \quad r = a \textbf{ mod } d.$$

***Remark:*** Note that both $a$ **div** $d$ and $a$ **mod** $d$ for a fixed $d$ are functions on the set of integers. Furthermore, when $a$ is an integer and $d$ is a positive integer, we have $a$ **div** $d = \lfloor a/d \rfloor$ and $a$ **mod** $d = a - d$. (See Exercise 24.)

Let $m$ be a positive integer and let $a$ and $b$ be integers. Then

$$(a + b) \textbf{ mod } m = ((a \textbf{ mod } m) + (b \textbf{ mod } m)) \textbf{ mod } m$$

and

$$ab \textbf{ mod } m = ((a \textbf{ mod } m)(b \textbf{ mod } m)) \textbf{ mod } m.$$

-Congruent

-If m divides (a-b), it's congruent. Denoted by “(mod m)”

If $a$ and $b$ are integers and $m$ is a positive integer, then $a$ is *congruent to* $b$ *modulo* $m$ if $m$ divides $a - b$. We use the notation $a \equiv b \pmod{m}$ to indicate that $a$ is congruent to $b$ modulo $m$. We say that $a \equiv b \pmod{m}$ is a **congruence** and that $m$ is its **modulus** (plural **moduli**). If $a$ and $b$ are not congruent modulo $m$, we write $a \not\equiv b \pmod{m}$.

-Be careful = vs ≡ in proofs.

Although both notations $a \equiv b \pmod{m}$ and $a$ **mod** $m = b$ include “mod,” they represent fundamentally different concepts. The first represents a relation on the set of integers, whereas the second represents a function. However, the relation $a \equiv b \pmod{m}$ and the **mod** $m$ function are closely related, as described in Theorem 3.

-Ex: 16 and 1 is congruent wrt modulo 5 (15/5). 16 and 2 is not congruent wrt modulo 5 (14/5).

Let $m$ be a positive integer. The integers $a$ and $b$ are congruent modulo $m$ if and only if there is an integer $k$ such that $a = b + km$.

-Proof: Suppose the hypothesis

$$\begin{aligned} a \equiv b (\text{mod } m) &\leftrightarrow m \mid (a - b) \\ &\leftrightarrow m \mid (a - b) \\ &\leftrightarrow a - b = km \, (\text{for some int } k) \\ &\leftrightarrow a = b + km \end{aligned}$$

Let $m$ be a positive integer. If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then

$a + c \equiv b + d \pmod{m}$ and $ac \equiv bd \pmod{m}$.

-Proof: Suppose the hypothesis

| | | |
|---|---|---|
| 1 | $m \mid (a - b)$ | defn of $\equiv \bmod m$ |
| 2 | $m \mid (c - d)$ | defn of $\equiv \bmod m$ |
| 3 | $a - b = km$ | $k \in \mathbb{Z}$ defn of divides |
| 4 | $c - d = lm$ | $l \in \mathbb{Z}$ defn of divides |
| 5 | $a = km + b$ | 3 |
| 6 | $c = lm + d$ | 4 |

| | | | | | |
|---|---|---|---|---|---|
| 7 | $a + c = (km + b) + (lm + d)$ | premise, 3, 4 | 15 | $ac = (km + b)(lm + d)$ | premise, 5, 6 |
| 8 | $a + c = km + lm + b + d$ | 7 | 16 | $ac = klm^2 + kmd + blm + bd$ | 15 |
| 9 | $a + c = m(k + l) + b + d$ | 8 | 17 | $ac = m(klm + kd + bl) + bd$ | 16 |
| 10 | $a + c - (b + d) = m(k + l)$ | 9 | 18 | $ac - bd = m(klm + kd + bl)$ | 17 |
| 11 | $a + c - (b + d) = m(k + l)$ | 10 | 19 | $ac - bd = m(h)$ | $h = klm + kd + bl$ |
| 12 | $(a + c) - (b + d) = m(j)$ | $j = k + l$ | 20 | $m \mid ac - bd$ | defn of divides |
| 13 | $m \mid (a + c) - (b + d)$ | defn of divides | 21 | $ac \equiv bd (\bmod m)$ | defn of $\equiv \bmod m$ |
| 14 | $a + c \equiv (b + d) \bmod m$ | defn of $\equiv \bmod m$ | $\therefore$ | The consequent is true | 14, 21 |

-Arithmetic mod m

-The operator with a subscript (that means there's also -, *, /, etc.)

$a +_m b = (a+b) \bmod m \quad a *_m b = (a*b) \bmod m$

-Properties of modular arithmetic

-Closure: Addition with integers are closed because the output is always in integers.
-Associativity: like (a + b) + c = a + (b + c) but with modular arithmetic
-Communicaticity:
-Identity: Something operated on something gives you what you start with.
-Additive Inverse: Like 5 + -5 is Additive Inverse normally. 4 +_m 1 = 0 where m=5
-Distributive

-Ex:

$$\begin{aligned}
45^{101} \bmod 13 &= 45 \cdot 45^{100} \bmod 13 \\
&= \left(\left(45^{100} \bmod 13\right)(45 \bmod 13)\right) \bmod 13 \\
&= 6 \cdot 45^{100} \bmod 13 \\
&= 6^{101} \bmod 13 \\
&= 6 \cdot 6^{100} \bmod 13 \\
&= 6 \cdot \left(6^2\right)^{50} \bmod 13 \\
&= 6 \cdot 36^{50} \bmod 13 \\
&= 6 \cdot \left(10^2\right)^{25} \bmod 13 \\
&= 6 \cdot 100^{25} \bmod 13 \\
&= 6 \cdot 9^{25} \bmod 13 \\
&= \ldots
\end{aligned}$$

-Application

-Pseudo Random algorithms use mod.
-ISBN scanners using mod to minimize errors.
-Cryptography

**Integer Representation**

-Binary (0,1)

$(10110)_2 = 1 \cdot 2^4 + 1 \cdot 2^2 + 1 \cdot 2^1 = 21$

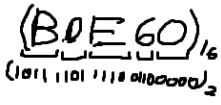
-Octal (0,...,7)

$(24601)_8 = 2 \cdot 8^4 + 4 \cdot 8^3 + 6 \cdot 8^2 + 1 \cdot 8^0$

-Hexadecimal (0,...,9,A,B,C,D,E,F)

$(93CA1)_{16} = 9 \cdot 16^4 + 9 \cdot 16^3 + C \cdot 16^2 + A \cdot 16^1 + 1 \cdot 16^0$

-Shortcut Hex to Bin




**Primes**

-An integer where it's factors are greater than 1. Not prime, then it's composite.

An integer $p$ greater than 1 is called *prime* if the only positive factors of $p$ are 1 and $p$. A positive integer that is greater than 1 and is not prime is called *composite*.

-Prime factorization

**THE FUNDAMENTAL THEOREM OF ARITHMETIC** Every integer greater than 1 can be written uniquely as a prime or as the product of two or more primes, where the prime factors are written in order of nondecreasing size.

-Ex:

$100 = 2 \cdot 2 \cdot 5 \cdot 5 = 2^2 5^2$, $81 = 3 \cdot 3 \cdot 3 \cdot 3 = 3^4$

$641 = 641$, $75 = 3 \cdot 5 \cdot 5 = 3^1 \cdot 5^2$

$999 = 3 \cdot 3 \cdot 3 \cdot 37 = 3^3 \cdot 37$, $83 = 83^1$

$1024 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^{10}$. $48 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 3 = 2^4 \cdot 3^1$

-Other Theorems

If $n$ is a composite integer, then $n$ has a prime divisor less than or equal to $\sqrt{n}$.

-Because sqrt(n) is like a midpoint for factors, if they are not perfect square, one is > sqrt(n) and one is <.

-There are infinitely many primes.

-Proof (Contradiction): Suppose there is a finite number of primes. p1, p2, p3,..., pn for some int n>=0..

-Let Q = p1 * p2 * p3 *...* pn + 1.

-By Thm1, Q is prime or the product of 2 more primes None of the primes pj divides Q, since for some k, Pj/!, then pk divides Q - p1p2...pj = 1

-So, either Q is prime (and is not equal to any of the pj, or it's the product of 2 or more primes, none of which are p1, p2, p3,..., pn.

**Greatest Common Divisor & Least Common Multiple**

-GCD

Let $a$ and $b$ be integers, not both zero. The largest integer $d$ such that $d \mid a$ and $d \mid b$ is called the *greatest common divisor* of $a$ and $b$. The greatest common divisor of $a$ and $b$ is denoted by $\gcd(a, b)$.

-Relatively prime

The integers $a$ and $b$ are *relatively prime* if their greatest common divisor is 1.

-LCM

The *least common multiple* of the positive integers $a$ and $b$ is the smallest positive integer that is divisible by both $a$ and $b$. The least common multiple of $a$ and $b$ is denoted by $\text{lcm}(a, b)$. (so if output is o, a|o & b|o, opposite of gcd)

-GCD and LCM Ex

$a = 4$, $b = 6$ $\quad \gcd(4,6) = 2$ $\quad \text{lcm}(4,6) = 12$

---

$a = 24$, $b = 8$ $\quad \gcd(24,8) = 8$ $\quad \text{lcm}(24,8) = 24$

---

$a = 9$, $b = 49$ $\quad \gcd(9,49) = 1$ $\quad \text{lcm}(9,49) = 441$

-Using prime factorization to find GCD and LCM

$$a = 4 = 2^2$$
$$b = 6 = 2^1 \cdot 3^1$$
$$\gcd(a,b) = 2^{\min(2,1)} \cdot 3^{\min(0,1)} = 2^1 \cdot 3^0$$
$$\text{lcm}(a,b) = 2^{\max(2,1)} \cdot 3^{\max(0,1)} = 2^2 \cdot 3^1$$

-The min and max arguments are from the number of times those factors appear for each a and b.

-This means that a and b can be described by prime factorization.

$$a = p_1^{a_1} \cdot p_1^{a_2} \cdot \ldots \cdot p_n^{a_n}$$
$$b = p_1^{b_1} \cdot p_1^{b_2} \cdot \ldots \cdot p_n^{b_n}$$
$$\gcd(a,b) = p_1^{\min(a_1,b_1)} \cdot p_2^{\min(a_2,b_2)} \cdot \ldots \cdot p_n^{\min(a_n,b_n)}$$
$$\text{lcm}(a,b) = p_1^{\max(a_1,b_1)} \cdot p_2^{\max(a_2,b_2)} \cdot \ldots \cdot p_n^{\max(a_n,b_n)}$$

-Lemma to find GCD and LCM

Let $a = bq + r$, where $a$, $b$, $q$, and $r$ are integers. Then $\gcd(a, b) = \gcd(b, r)$.

-Proof from book

***Proof:*** If we can show that the common divisors of $a$ and $b$ are the same as the common divisors of $b$ and $r$, we will have shown that $\gcd(a, b) = \gcd(b, r)$, because both pairs must have the same *greatest* common divisor.

So suppose that $d$ divides both $a$ and $b$. Then it follows that $d$ also divides $a - bq = r$ (from Theorem 1 of Section 4.1). Hence, any common divisor of $a$ and $b$ is also a common divisor of $b$ and $r$.

Likewise, suppose that $d$ divides both $b$ and $r$. Then $d$ also divides $bq + r = a$. Hence, any common divisor of $b$ and $r$ is also a common divisor of $a$ and $b$.

Consequently, $\gcd(a, b) = \gcd(b, r)$. ◁

-Proof: We show that the set of common divisors of a & b is equal to the set of common divisors of b & r. This would imply both gcd are equal.

-Suppose d | a and d | b.

-That means d | (a-bq=r)

-Since d | a, a = kd & d | b, b=ld (k & l in int.

-r=a-bq=kd-ldq=d(k-lq), so d | r

-Suppose d | b and d | r.

-Then d | (r+bq=a).

-Since d | b, b = kd & d | r, r= ld (k & l in int)

-a=r+bq=ld+kdq=d(l+kg), so d | a.

-Therefore, the claim is true.

-Algorithm to find GCD of 2 numbers (Euclid's Algorithm) (a > b)

Input: positive integers a and b

Output: gcd(a,b)

```
gcd(a,b)
    x ← a
    y ← b
    while y ≠ 0 do
        r ← x mod y
        x ← y
        y ← r
    return x
```

-This utilizes the lemma, since it is iterating by finding b & r.

-Trace table ex of algorithm

$a = 21,\ b = 13$

| $r$ | $x$ | $y$ |
|---|---|---|
| | 21 | 13 |
| 8 | 13 | 8 |
| 5 | 8 | 5 |
| 3 | 5 | 3 |
| 2 | 3 | 2 |
| 1 | 2 | 1 |
| 0 | 1 | 0 |

returns 1

$a = 96,\ b = 56$

| $r$ | $x$ | $y$ |
|---|---|---|
| | 96 | 56 |
| 40 | 56 | 40 |
| 16 | 40 | 16 |
| 8 | 16 | 8 |
| 0 | 8 | 0 |

returns 8

$a = 18,\ b = 14$

| $r$ | $x$ | $y$ |
|---|---|---|
| | 18 | 14 |
| 4 | 14 | 4 |
| 2 | 4 | 2 |
| 0 | 2 | 0 |

returns 2

**Proof by Induction**

-How do we show an infinite amount of statements are true?

$$(P(1) \wedge \forall k(P(k) \rightarrow P(k+1))) \rightarrow \forall n P(n),$$

-Such as Ax P(x) where P -> q, we've done it already by proving it generically, meaning all cases will work.

-But what about Ax>0 P(x)?

Idea: To show that $\forall n \geq 1\ P(n)$ is true,

we show that ① $P(1)$ is true and

② $\forall k \geq 1\ (P(k) \rightarrow P(k+1))$

Why does this show that $\forall n \geq 1\ P(n)$ is true?

P(1) P(2) P(3) P(4) . . .

✓

P(1)

P(1) ∧ P(1)→P(2) P(2) ∧ P(2)→P(3) . . .

-After showing that P(k) (akaBase Step) is true, showing P(k+1) and onwards is by modus ponens infinitely chaining combo wombo.

-Essense of Induction

-Induction Hypothesis: Suppose P(k) is true for some fixed $k \geq c\ (c \in \mathbb{Z})$. (Is P(k) true at the start?)

-Induction Step: Then show that P(k+1) is true so that $\forall k \geq c(P(k) \rightarrow P(k+1))$. (Is P(k) true for all other k greater than the start?)

-Use the hypothesis here in order to prove it.

-By Modus Ponens & UI with the Induction Hypothesis & Step, we show that it's true like a chain reaction.

-A variation is going backwards $\forall k \geq c(P(k) \rightarrow P(k-1))$.

-Only works on int, because it's a pain to increment by infinitely small steps.

-Classic LULW Ex: Gauss Summation

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

Proof: Proof by induction on n.

Let $P(n)$ be the statement $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$. (VERY IMPORTANT! State what your statement is, else you prob have no goal)

Basis Step: $P(1) = \sum_{i=1}^{1} i = 1 = \frac{1(1+1)}{2} = \frac{2}{2} = 1$, which is true.

Induction Hypothesis: Suppose $P(n)$ is true for some fixed $n \geq 1$.

That means, suppose $\sum_{i=1}^{n} i$ is true for some fixed $n \geq 1$.

Induction Step: We need to show $\sum_{i=1}^{n+1} i = \frac{(n+1)((n+1)+1)}{2}$

$$\begin{aligned}\sum_{i=1}^{n+1} i &= \sum_{i=1}^{n} i + (n+1) \ : (\text{pull out n+1 term of } \textstyle\sum) \\ &= \frac{n(n+1)}{2} + (n+1) \ : (\text{induction hypothesis}) \\ &= \frac{n^2+n}{2} + \frac{2n+2}{2} \\ &= \frac{n^2+3n+2}{2} \\ &= \frac{(n+1)(n+2)}{2} \\ &= \frac{(n+1)((n+1)+1)}{2}\end{aligned}$$

This concludes the proof by induction $\square$

-Another Example, Sum of i^2.

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

Proof: Proof by induction on n.

Let $P(n)$ be the statement $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$.

Basis Step: $P(1) = \sum_{i=1}^{1} i = 1 = \frac{1(1+1)(2(1)+1)}{6} = \frac{6}{6} = 1$, which is true.

Induction Hypothesis: Suppose $\sum_{i=1}^{n} i^2$ is true for some fixed $n \geq 1$.

Induction Step: We need to show $\sum_{i=1}^{n+1} i^2 = \frac{(n+1)((n+1)+1)(2(n+1)+1)}{6}$

$$\begin{aligned}\sum_{i=1}^{n+1} i^2 &= \sum_{i=1}^{n} i^2 + (n+1)^2 : (\text{Pull out n+1 term}) \\ &= \frac{n(n+1)(2n+1)}{6} + (n+1)^2 : \ (\text{Induction Hypothesis}) \\ &= \frac{2n^3+3n^2+n}{6} + \frac{6n^2+12n+6}{6} \\ &= \frac{2n^3+9n^2+13n+6}{6} \\ &\ \ \vdots \ (\text{factor stuff}) \\ &= \frac{(n+1)(n+2)(2n+3)}{6} \\ &= \frac{(n+1)((n+1)+1)(2(n+1)+1)}{6}\end{aligned}$$

This concludes the proof by induction $\square$

-Inequality ex

Claim: $2^n > n$ for all $n \geq 1$.

Proof: By induction on n.

Let $P(n)$ be the statement $2^n > n$.

Basis Step: $P(1)$ is $2^1 = 2 > 1$.

Ind Hypo: Suppose $2^n > n$ for some fixed $n \geq 1$.

Ind Step: We need to show $2^{n+1} > n+1$.

$$\begin{aligned}2^{n+1} &= 2 \cdot 2^n \\ &> 2n \, (\text{by Ind Hypo}) \\ &= n + n \\ &\geq n + 1 \, (n \geq 1)\end{aligned}$$

This concludes the proof by induction $\square$

-Divides ex

Claim: For all $n \geq 1$, 2 divides $n^2 + n$

Proof: By induction on n.

Let $P(n)$ be the statement "2 divides $n^2 + n$".

Basis Step: $P(1) = 1^2 + 1 = 2$ and $2 \mid 2$.

Ind Hypo: Suppose $2 \mid n^2 + n$ for some fixed $n \geq 1$.

Ind Step: We need to show $2 \mid (n+1)^2 + (n+1)$ or $2 \mid n^2 + 3n + 2$.

$$\begin{aligned} n^2+3n+2 &= (n+1)(n+2) \\ &= \left(n^2+n\right)+(2n+2) \\ &= 2k+2(n+1)\,(k\in\mathbb{Z})(\text{Ind Hypo}) \\ &= 2(k+n+1) \end{aligned}$$

Therefore it divides because that junk is even (aka divides by 2).

-Iterative Factorial Pseudocode ex (proving the for loop)

**Input:** Positive integer $n$
**Output:** $n!$

```
factorial (n)
1 result ← 1
2 for i ← 1 to n do
3    result ← i * result
4 return result
```

Claim: At the end of the $i^{th}$ iteration, the variable result is i.

Proof: By induction on i.

Let P(i) be "at the end of the $i^{th}$ iteration, the variable result is i."

Basis step: i=1, result has the value of 1 after line 1. After lines 2 & 3, result contains result * i = 1 * 1 = 1. So, the 1st iteration ends with result = 1!.

Induction Hypothesis: Suppose P(i) is true for some fixed i ≥ 1.

Induction Step: We need to show P(i+1) is true.

-By the ind hyp, at the end of the $i^{th}$ iteration, results = i!. After line 3 in thein the (i+1)th iteration, result = i+1 = (i+1)!.

-So, after the (i+1)! iteration , result = (i+1)!

-Recursive Factorial Pseudocode ex (proving the for loop)

```
factorial (n)
1 if n = 1 then
2    return 1
3 else
4    return n * factorial(n − 1)
```

(input = pos int, output n!)

Claim: factorial(n) returns n! (for all n≥1)

Proof: By induction on n.

Let P(n) be "factorial(n) returns n!"

Basis Step: n=1. factorial(1) executes line1, which meets the condition returning 1=1!. (line2)

Induction Hypothesis: Suppose factorial(n) returns n! for some fixed n≥1.

Induction Step: We need to show factorial(n+1) returns (n+1)!.

Since n≥1, then (n+1)≥2. So factional(n+1) executes line1, the condition is false, and so returns (n+1)*factorial((n+1)-1).

By the induction hypothesis, factorial((n+1)-1) = factorial(n) returns n!.

So, factorial(n+1) returns (n+1)*n! = (n+1)!

-Recursion math ex

Let $a_1=1,\ a_n=n\cdot a_{n-1}$

Claim: $a_n=n!$ for all n≥1.

Proof: By induction on n.

Let P(n) be the statement "$a_n=n!$"

Basis Step: n=1, $a_1$=1=1!

Induction Hypothesis: Suppose $a_n=n!$ for some fixed n≥1.

Induction Step: We need to show P(n+1), which is $a_{n+1}=(n+1)!$

$$\begin{aligned} a_{n+1} &= (n+1)\cdot a_{n+1-1}\ (\text{defn of } a_{n+1}, n+1\geq 2) \\ &= (n+1)\cdot a_n \\ &= (n+1)n! \qquad (\text{ind hypo}) \\ &= (n+1)! \end{aligned}$$

**Strong Induction**

-With strong induction, we assume P(n+1) and on is true, and we need to show P(n). In normal induction, we assume P(n) is true and we show P(n+1), the reversed.

We show $\forall n\geq 1 P(n)$ by:

$\cdot P(1)$

$\cdot \forall n\geq 1((\forall 1\leq j\leq n P(j))\rightarrow P(n+1))$

-Fibonacci Pseudocode ex

**Input:** Positive integer $n$
**Output:** The $n$th Fibonacci number (where $F_1 = 1$, $F_2 = 1$, and $F_n = F_{n-1} + F_{n-2}$)

```
fibonacci (n)
1 if n = 1 or n = 2 then
2    return 1
  else
3    return fibonacci (n − 1) + fibonacci (n − 2)
```

-Proof FAILURE by normal induction

Claim: fibonacci(n) returns the $n^{th}$ fibonacci number ($F_n$) for all n≥1

Proof: By (normal) induction on n.

Let P(n) be the statement "fibonacci(n) returns $F_n$"

Basis Steps: n=1 and 2, fibonacci(n) follows line1 where the conditional is true and returns 1 at line2.

Ind Hyp: Suppose fib(n) returns $F_n$, for some fixed n≥2.

Ind Step: We need to show fib(n+1) returns $F_{n+1}$,

Since n≥2, then n+1≥3.

fib(n+1) follows line1, the conditional is false, and it returns fib((n+1)-1) + fib((n+1)-2).

By the induction hypothesis fib((n) + fib(n+1)-2) = $F_n$ + fib(n-1).
(But now you are stuck because what is fib(n-1)?)

-Proof W by Strong Induction

Claim: fibonacci(n) returns the $n^{th}$ fibonacci number ($F_n$) for all n≥1
Proof: By strong induction on n.
Let P(n) be the statement "fibonacci(n) returns $F_n$"
Basis Steps: n=1 and 2, fibonacci(n) follows line1 where the conditional is true and returns 1 at line2.
Ind Hyp: Suppose for all 1≤j≤n fib(j) returns $F_n$, for some fixed n≥2.
Ind Step: We need to show fib(n+1) returns $F_{n+1}$,
Since n≥2, then n+1≥3.
fib(n+1) follows line1, the conditional is false, and it returns fib((n+1)-1) + fib((n+1)-2) = fib((n) + fib(n-1)
(n-1≤n≤n & 1≤n-1≤n, because n≥2).
By the induction hypothesis fib((n) + fib(n-1) = $F_n + F_{n-1}$.
And by defn of $F_n, n+1 > 2$, fib(n+1) returns $F_n + F_{n-1} = F_{n+1}$.

-Prime Factorization ex

Claim: If $n$ is an integer and $n > 1$, then $n$ can be written as a product of primes.
Proof: Proof by strong induction.
Let the statement $P(n)$ be the claim :)
Basis Step: $n = 2, 2$ can be written as the product of 1 prime, itself.
Ind Hyp: Suppose for all $2 \leq j \leq n$, and for some fixed $n \geq 2$, $P(j)$ is true.
Ind Setp: We need to show that $P(n+1)$ is true.
$n+1$ is either prime, or composite.
Case1: $n+1$ is prime. The product of primes of $n+1$ is itself.
Case2: $n+1$ is composite. $n+1 = a \cdot b$, where $2 \leq a \leq b \leq n$.
By Ind Hyp, $a$ can be written as the product of prime and $b$ can written as a product of primes.
$n+1$ can be written as the product of primes of a and the product of primes of b, which is a product of primes.

**Matrices**

-John wick redpilled linear algebra!

A *matrix* is a rectangular array of numbers. A matrix with $m$ rows and $n$ columns is called an $m \times n$ matrix. The plural of matrix is *matrices*. A matrix with the same number of rows as columns is called *square*. Two matrices are *equal* if they have the same number of rows and the same number of columns and the corresponding entries in every position are equal.

Let $m$ and $n$ be positive integers and let

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}.$$

The $i$th *row* of $\mathbf{A}$ is the $1 \times n$ matrix $[a_{i1}, a_{i2}, \ldots, a_{in}]$. The $j$th *column* of $\mathbf{A}$ is the $m \times 1$ matrix

$$\begin{bmatrix} a_{1j} \\ a_{2j} \\ \cdot \\ \cdot \\ \cdot \\ a_{mj} \end{bmatrix}.$$

The $(i, j)$th *element* or *entry* of $\mathbf{A}$ is the element $a_{ij}$, that is, the number in the $i$th row and $j$th column of $\mathbf{A}$. A convenient shorthand notation for expressing the matrix $\mathbf{A}$ is to write $\mathbf{A} = [a_{ij}]$, which indicates that $\mathbf{A}$ is the matrix with its $(i, j)$th element equal to $a_{ij}$.

-Addition

-only on same dimension matrices. Commutative and Associative.

Let $\mathbf{A} = [a_{ij}]$ and $\mathbf{B} = [b_{ij}]$ be $m \times n$ matrices. The *sum* of $\mathbf{A}$ and $\mathbf{B}$, denoted by $\mathbf{A} + \mathbf{B}$, is the $m \times n$ matrix that has $a_{ij} + b_{ij}$ as its $(i, j)$th element. In other words, $\mathbf{A} + \mathbf{B} = [a_{ij} + b_{ij}]$.

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 2 & -3 \\ 3 & 4 & 0 \end{bmatrix} + \begin{bmatrix} 3 & 4 & -1 \\ 1 & -3 & 0 \\ -1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 4 & 4 & -2 \\ 3 & -1 & -3 \\ 2 & 5 & 2 \end{bmatrix}.$$

-Multiplication

-Row * Column cancer. Only works if A's rows = B's columns. Not communicative.

Let $\mathbf{A}$ be an $m \times k$ matrix and $\mathbf{B}$ be a $k \times n$ matrix. The *product* of $\mathbf{A}$ and $\mathbf{B}$, denoted by $\mathbf{AB}$, is the $m \times n$ matrix with its $(i, j)$th entry equal to the sum of the products of the corresponding elements from the $i$th row of $\mathbf{A}$ and the $j$th column of $\mathbf{B}$. In other words, if $\mathbf{AB} = [c_{ij}]$, then

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{ik}b_{kj}.$$

$$A_{m \times n} B_{n \times p} = C_{m \times p}$$

$$c_{ij} = \sum_{k=1}^{n} a_{ik} \cdot b_{kj}$$

$$\begin{bmatrix} 5 & -4 & 6 \\ -2 & 1 & -1 \\ 4 & 2 & -6 \end{bmatrix}\begin{bmatrix} 1 & 1 & -1 \\ 0 & -3 & 6 \\ 7 & 3 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} (5\cdot 1)+(-4\cdot 0)+(6\cdot 7) & (5\cdot 1)+(-4\cdot -3)+(6\cdot 3) & (5\cdot -1)+(-4\cdot 6)+(6\cdot 1) \\ (-2\cdot 1)+(1\cdot 0)+(-1\cdot 7) & (-2\cdot 1)+(1\cdot -3)+(-1\cdot 3) & (-2\cdot -1)+(1\cdot 6)+(-1\cdot 1) \\ (4\cdot 1)+(2\cdot 0)+(-6\cdot 7) & (4\cdot 1)+(2\cdot -3)+(-6\cdot 3) & (4\cdot -1)+(2\cdot 6)+(-6\cdot 1) \end{bmatrix}$$
$$= \begin{bmatrix} 47 & 35 & -23 \\ -9 & -8 & 7 \\ -38 & -20 & 2 \end{bmatrix}$$

-If you do $A_{m\times n}B_{n\times p} = C_{m\times p}$

- $m \cdot n \cdot p$ terms to add together.
-Communicative cases: p=m=n.
-

-Equal (obviously bruh)

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \neq \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix}$$

-Identity

The *identity matrix of order n* is the $n \times n$ matrix $\mathbf{I}_n = [\delta_{ij}]$, (the *Kronecker delta*) where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$. Hence,

$$\mathbf{I}_n = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ 0 & 0 & \dots & 1 \end{bmatrix}.$$

$$AI = A = IA$$

-Power

$$A^0 = I$$
$$A^n = \underbrace{A \dots A}_{n \text{ times}} = AA^{n-1}$$

-Zero-one matrices

-Matrices with only 1s and 0s.

Let $\mathbf{A} = [a_{ij}]$ and $\mathbf{B} = [b_{ij}]$ be $m \times n$ zero–one matrices. Then the *join* of $\mathbf{A}$ and $\mathbf{B}$ is the zero–one matrix with $(i, j)$th entry $a_{ij} \vee b_{ij}$. The join of $\mathbf{A}$ and $\mathbf{B}$ is denoted by $\mathbf{A} \vee \mathbf{B}$. The *meet* of $\mathbf{A}$ and $\mathbf{B}$ is the zero–one matrix with $(i, j)$th entry $a_{ij} \wedge b_{ij}$. The meet of $\mathbf{A}$ and $\mathbf{B}$ is denoted by $\mathbf{A} \wedge \mathbf{B}$.

-Join (OR) and Meet (AND) (same dimension only)

Find the join and meet of the zero–one matrices

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}.$$

*Solution:* We find that the join of $\mathbf{A}$ and $\mathbf{B}$ is

$$\mathbf{A} \vee \mathbf{B} = \begin{bmatrix} 1 \vee 0 & 0 \vee 1 & 1 \vee 0 \\ 0 \vee 1 & 1 \vee 1 & 0 \vee 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

The meet of $\mathbf{A}$ and $\mathbf{B}$ is

$$\mathbf{A} \wedge \mathbf{B} = \begin{bmatrix} 1 \wedge 0 & 0 \wedge 1 & 1 \wedge 0 \\ 0 \wedge 1 & 1 \wedge 1 & 0 \wedge 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \wedge \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} (1 \wedge 1) \vee (0 \wedge 0) \vee (1 \wedge 1) \\ (0 \wedge 1) \vee (0 \wedge 0) \vee (1 \wedge 1) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

-Boolean Power

Let $\mathbf{A}$ be a square zero–one matrix and let $r$ be a positive integer. The $r$th *Boolean power* of $\mathbf{A}$ is the Boolean product of $r$ factors of $\mathbf{A}$. The $r$th Boolean product of $\mathbf{A}$ is denoted by $\mathbf{A}^{[r]}$. Hence,

$$\mathbf{A}^{[r]} = \underbrace{\mathbf{A} \odot \mathbf{A} \odot \mathbf{A} \odot \cdots \odot \mathbf{A}}_{r \text{ times}}.$$

(This is well defined because the Boolean product of matrices is associative.) We also define $\mathbf{A}^{[0]}$ to be $\mathbf{I}_n$.

$$A^{[0]} = I_n$$
$$A^{[n]} = \underbrace{A \dots A}_{n \text{ times}} = AA^{[n-1]}$$

**Relations**

-Binary Relation 😳gf?

Let $A$ and $B$ be sets. A *binary relation from A to B* is a subset of $A \times B$.

-Basically a subset of A x B that takes an element from A and one from B and puts them in an ordered pair.

-Ex:

$A = \{x \mid x \text{ is a person}\}$

$B = \mathbb{Z}$

$\text{A possible relation is } \{(\text{Mary}, 25), (\text{Juan}, 17), (\text{Ivan}, 33)\}$

-This relation can be Mary is 25 yo, Juan is 17 yo, and Ivan is 33.

-Function as a special kind of relation

-A function is a special kind of relation. What additional properties does a function have?

-$\{(\text{Mary}, 25), (\text{Juan}, 17), (\text{Ivan}, 33), (\text{Juan}, 23)\}$ is not a function. It is relating Juan twice.

-If it's a function f: A -> B

-Every element from A must be mapped to some B.

-A Relation on a Set / Itself

-A relation on set a is A subset of A x A

A *relation on a set* $A$ is a relation from $A$ to $A$.

-Ex: whatever x and y that meets the set builder conditions can be a relation in R

$R = \{(x, y) \mid x, y \in \mathbb{Z} \text{ and } x = y^2\}$

$(0, 0) \in R$

$(1, 1) \in R$

$(4, 2) \in R$

$(9, -3) \in R$

-Ex: Classic LULW of knowing people

$R = \{(x, y) \mid x \text{ and } y \text{ are people and } x \text{ knows } y\}$

$A = \{\text{Alice, Barbara, Chan, Debra, Enrico, Francois}\}$

-Binary Matrix to represent relations. (Letter is 1st of Name)

| | $A$ | $B$ | $C$ | $D$ | $E$ | $F$ |
|---|---|---|---|---|---|---|
| $A$ | 0 | 1 | 0 | 0 | 0 | 1 |
| $B$ | 0 | 0 | 1 | 1 | 0 | 0 |
| $C$ | 1 | 0 | 0 | 0 | 0 | 0 |
| $D$ | 0 | 1 | 0 | 0 | 0 | 0 |
| $E$ | 0 | 0 | 0 | 0 | 0 | 1 |
| $F$ | 0 | 0 | 0 | 0 | 0 | 0 |

(Since rox by column, an element is (Alice, Barbara), Alice knows Barbara)

-You can also do a scuffed spaghetti graph, drawing from one element to another element.

-Reflexive

-Each element is paired with itself.

A relation $R$ on a set $A$ is called *reflexive* if $(a, a) \in R$ for every element $a \in A$.

-The binary matrix representation would have a full diagonal of 1.

| | $A$ | $B$ | $C$ | $D$ | $E$ | $F$ |
|---|---|---|---|---|---|---|
| $A$ | 1 | 1 | 0 | 0 | 0 | 1 |
| $B$ | 0 | 1 | 1 | 1 | 0 | 0 |
| $C$ | 1 | 0 | 1 | 0 | 0 | 0 |
| $D$ | 0 | 1 | 0 | 1 | 0 | 0 |
| $E$ | 0 | 0 | 0 | 0 | 1 | 1 |
| $F$ | 0 | 0 | 0 | 0 | 0 | 1 |

-scuffed spaghetti graphs would have each element having an arrow pointing to itself.

-Symmetric

-For every ordered pair (a,b), there is (b,a).

-Antisymmetric is that if (a,b), then there is no (b,a).

-Antisymmetry doesn't care about reflexivity, so it can be both.

-A reflexive relation where the only pairs are the ones to make it reflexive is both symmetric and antisymmetric.

A relation $R$ on a set $A$ is called *symmetric* if $(b, a) \in R$ whenever $(a, b) \in R$, for all $a, b \in A$.
A relation $R$ on a set $A$ such that for all $a, b \in A$, if $(a, b) \in R$ and $(b, a) \in R$, then $a = b$ is called *antisymmetric*.

-Symmetric across the main diagonal

| | $A$ | $B$ | $C$ | $D$ | $E$ | $F$ |
|---|---|---|---|---|---|---|
| $A$ | 0 | 1 | 1 | 0 | 1 | 1 |
| $B$ | 1 | 0 | 1 | 1 | 0 | 0 |
| $C$ | 1 | 0 | 0 | 0 | 0 | 0 |
| $D$ | 0 | 1 | 0 | 0 | 0 | 0 |
| $E$ | 1 | 1 | 0 | 0 | 1 | 1 |
| $F$ | 1 | 0 | 0 | 0 | 1 | 0 |

-scuffed spaghetti graphs would have each element pair have arrows point to and from.

-Transitive

-If (a,b) and (b,c), there is a (a,c) for every pair of elements like that.

A relation $R$ on a set $A$ is called *transitive* if whenever $(a, b) \in R$ and $(b, c) \in R$, then $(a, c) \in R$, for all $a, b, c \in A$.

-Ex:

$A = \mathbb{Z}$

$R = \{(x, y) \mid x \text{ divides } y\}$

$(2, 4), (4, 32), (2, 32) \in R$

-Equivalence

-Symmetric, Reflexive, & Transitive

A relation on a set $A$ is called an *equivalence relation* if it is reflexive, symmetric, and transitive.

| | $A$ | $B$ | $C$ | $D$ | $E$ | $F$ |
|---|---|---|---|---|---|---|
| $A$ | 1 | 0 | 0 | 0 | 0 | 0 |
| $B$ | 0 | 1 | 0 | 1 | 1 | 0 |
| $C$ | 0 | 0 | 1 | 0 | 0 | 1 |
| $D$ | 0 | 1 | 0 | 1 | 0 | 0 |
| $E$ | 0 | 1 | 0 | 0 | 1 | 0 |
| $F$ | 0 | 0 | 1 | 0 | 0 | 1 |

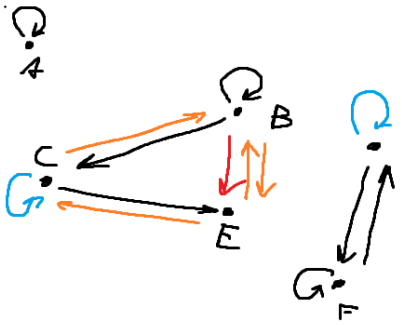
-Closure

-The minimum steps/edges that make the relation symmetric, reflexive, transitive.

-"symmetric closure", "transitive closure", "reflexive closure".

If $R$ is a relation on a set $A$, then the **closure** of $R$ with respect to **P**, if it exists, is the relation $S$ on $A$ with property **P** that contains $R$ and is a subset of every subset of $A \times A$ containing $R$ with property **P**.

-Ex: Blue are edges for a reflexive closure, red for transitive closure, orange for symmetric closure.




(oops, there should be blue loop on E)

-Meet Power and Relations

- $R^{[2]}$ can mean that if (a,b) & (b,c) in *R*, there is (a,c) after 3 connections (going from a->b->c).

- $R^{[3]}$ means in 3 connections.

-An application of this is roads connecting to other roads. $R^{[n]}$ means that after n road changes, you can get from roads (a,b) if that pair is $R^{[n]}$.

-n-ary Relation

-So instead of pair, it's n-tuple.

Let $A_1, A_2, \ldots, A_n$ be sets. An *n-ary relation* on these sets is a subset of $A_1 \times A_2 \times \cdots \times A_n$. The sets $A_1, A_2, \ldots, A_n$ are called the *domains* of the relation, and $n$ is called its *degree*.

-Ex: Three sets can be the set of First Names (F), Middle Names (M), and Last Names (L).

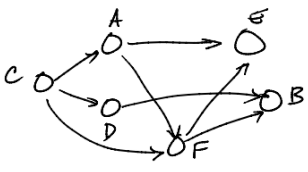
-A element in the Relation of FxMxL can be (Joe, Bruh, Mama) denoting a named Joe Bruh Mama.

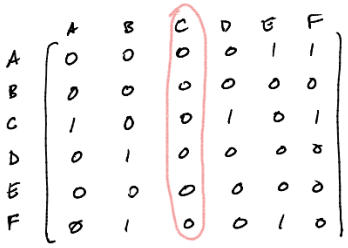
-Databases use this to format n-tuples to store data.

-Topological Sort

-Sorting by if a prerequisite is needed, where the first element has no prereqs.

Problem: Given a list of tasks, where each task has a list of "prerequisites," find an ordering of the tasks that respects the constraints.

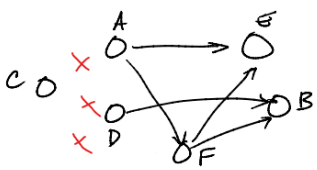


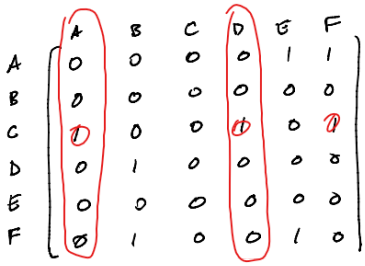




-From the graph and diagram

-C doesn't have an arrow pointing to it, and its column is only 0.

-The next step is to cut off all connections from c and see which has no prereqs.



**IDK**

-Min and Max Recursive defn

$$\max(\max(a_1, a_2, \ldots, a_{n-1}), a_n)$$
$$\min(\min(a_1, a_2, \ldots, a_{n-1}), a_n)$$

**Cheat sheesh?**

$$0^b = 0$$
$$a^1 = 0$$
$$a^b \cdot a^c = a^{b+c}$$
$$a^b / a^c = a^{b-c} = 1/a^{c-b}$$
$$a^{-b} = 1/a^b$$
$$a^{b/c} = \sqrt[c]{a^b}$$
$$a^b \cdot a^b = a^{2b}$$
$$a^{b/b} = a^1$$
$$\left(a^b\right)^c = a^{bc}$$
$$a^c \cdot b^c = (ac)^b$$
$$a^c / b^c = (a/b)^c$$
$$x = \frac{-b \pm \sqrt{b^2 + 4ac}}{2a}$$

$$\log_e x = \ln x$$
$$\log_a (bc) = \log_a b + \log_a c$$
$$\log_a (b/c) = \log_a b - \log_a c$$
$$\log_a b^c = c \log_a b$$
$$\log_a b / \log_a c = \log_c b$$
$$a^{\log_a b} = b$$
$$\log_a a = 1$$
$$\log_a 1 = 0$$
$$(\ln x)' = 1/x$$
$$(\log_b x)' = 1/x \ln b$$
$$(a^x)' = a^x \ln a$$

$$\log(1) = 0$$
$$\log_a\left(x^b\right) = b \cdot \log_a(x)$$
$$\log_a\left(\frac{1}{x}\right) = -\log_a(x)$$
$$\log_a(b) = \frac{\ln(b)}{\ln(a)}$$
$$\log_x\left(\left(\frac{1}{x}\right)^n\right) = -n$$

$$\log_a(a) = 1$$
$$\log_{a^b}(x) = \frac{1}{b}\log_a(x)$$
$$\log_{\frac{1}{a}}(x) = -\log_a(x)$$
$$\log_x\left(x^n\right) = n$$
$$a^{\log_a(b)} = b$$