Mat 2345 — Discrete Math

Chapter One
Logic, Sets, & Functions

Nancy Van Cleave

Fall 2007

General Guidelines

- Syllabus
- Schedule (note exam dates)
- Homework & Programs
- Academic Integrity Guidelines
- Course Web Site

Mat2345 Course Overview

An introduction to the mathematical foundations needed by computer scientists.

A survey of discrete sturctures and methods

- logic
- sets
- functions
- algorithms – developing and analyzing
- induction proofs
- recursion relations
- relations
- Boolean algebra, logic gates, circuits
- modeling computation – Finite State and Turing Machines

Course Themes

- **Mathematical Reasoning** – induction
- **Mathematical Analysis** – comparison of algorithms / functions
- **Discrete Structures** – abstract math structures, the relationship between discrete and abstract structures
- **Algorithmic Thinking** – algorithmic paradigms
- **Applications and Modeling** – can we predict behavior?

Student Responsibilities — Week 1

- **Reading**: Textbook, Sections 1.1 & 1.2
- **Homework 1.1**: 2a–d, 4a–e, 6a–e, 8a–d, 20a–c, 35 (show work); Table of **T/F** combinations and simple propositions — due in class on Friday
- **Homework 1.2**: 4, 8a–b, 10a–b, 18, 30, 32 — due in class on Monday
- **Attendance**

Logic

- The rules of logic are used to distinguish between **valid** and **invalid** mathematical arguments.
- Logic rules have many applications in computer science. They are used in:
  - the design of computer circuits
  - the construction of computer programs
  - the verification of the correctness of programs
  - as the basis of some Artificial Intelligence programming languages.
  - and many other ways as well

Propositions

- **PROPOSITION**: a statement that is either true or false, but not both.

  Examples (which are **true?**):
  - The zip code for Charleston, IL is 61920.
  - The Will Roger's Theater is on Monroe Street.
  - $1 + 4 = 5$
  - $1 + 3 = 5$
  - The title of our course is Mathemagics.

  Counterexamples:
  - Where am I?
  - Stop!
  - $x + 2y = 4$

- **Variables** are generally used to represent propositions: $p, q, r, s, \ldots$
- **Tautology**: a proposition which is always **true**.
- **Contradiction**: a proposition which is always **false**.
- **Compound Proposition**: a new proposition formed from existing propositions using **logical operators** (aka *connectives*).
- **Negation**: let p be a propostion. The negation of p is the proposition "It is not the case that p," denoted by $\neg p$.

Truth Tables

Truth Tables display the relationship between the truth values of propositions.

The truth table for **negation**:

p	$\neg p$
T	F
F	T

When proposition p is true, its negation is false.
When it is false, its negation is true.

The negation of "Today is Monday" is "Today is not Monday" or "It is not the case that today is Monday"

Conjunction

Conjunction: the compound proposition **p and q**, or $p \wedge q$ which is **true** when both p and q are **true** and **false** otherwise.

Let p = Today is Monday, and q = It is raining.
What is the value of each of the following conjunctions?

$$p \wedge q$$

$$p \wedge \neg q$$

Disjunction

Disjunction: the compound proposition **p or q**, or $p \vee q$ which is **false** when both p and q are **false** and **true** otherwise.

p	q	$p \wedge q$	$p \vee q$
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	F

Exclusive Or

Exclusive Or: $p \oplus q$, the proposition that is **true** when exactly ***one*** of p and q is **true**, and is **false** otherwise.

- "Fries or baked potato come with your meal"
- "Do the dishes or go to your room"

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Implication

Implication: $p \rightarrow q$, the proposition that is false when p is true and q is false, and true otherwise.

Related implications:

- Converse of $p \rightarrow q$ is $q \longrightarrow p$
- Contrapositive of $p \rightarrow q$ is $\neg q \longrightarrow \neg p$
- Biconditional $p \leftrightarrow q$, p if and only if q, p iff q: the proposition which is true when p and q have the same truth values, and false otherwise.

Implication		
p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Implication: If today is Monday, then MAT2345 meets today.

Converse: If MAT2345 meets today, then today is Monday.

Contrapositive: If MAT2345 does not meet today, then today is not Monday.

Implications, aka Conditionals

Converse, Inverse, and Contrapositive

Direct Statement	$p \rightarrow q$	If p, then q
Converse	$q \rightarrow p$	If q, then p
Inverse	$\sim p \rightarrow \sim q$	If not p, then not q
Contrapositive	$\sim q \rightarrow \sim p$	If not q, then not p

Let p = "they stay" and q = "we leave"

Direct Statement ($p \rightarrow q$):

Converse:

Inverse:

Contrapositive:

Let p = "I surf the web" and q = "I own a PC"

Direct Statement ($p \rightarrow q$):

Converse:

Inverse:

Contrapositive:

Equivalent Conditionals

		Direct	Converse	Inverse	Contrapositive
		$p \rightarrow q$	$q \rightarrow p$	$\sim p \rightarrow \ \sim q$	$\sim q \rightarrow \ \sim p$
p	q	$\sim p \vee q$			
T	T	T	T		
T	F	F	T		
F	T	T	F		
F	F	T	T		

$\square \rightarrow \triangle$ is equivalent to $\sim \square \vee \triangle$

$$\sim \square \vee \triangle \equiv \square \rightarrow \triangle$$

$$\square \vee \triangle \equiv \ \sim \square \rightarrow \triangle$$

Tricky Question

For $p \vee q$, write each of the following:

Direct Statement:

Converse:

Inverse:

Contrapositive:

Alternate Conditional Forms

Common translations of $p \rightarrow q$

If p, then q	p is sufficient for q
If p, q	q is necessary for p
p implies q	All p's are q's
p only if q	q if p

These translations do not in any way depend upon the truth value of $p \rightarrow q$.

"**If you get home late, then you are grounded**" $\equiv$

You are grounded if you get home late.

Getting home late is sufficient for you to get grounded.

Getting grounded is necessary when you get home late.

Getting home late implies that you are grounded.

Logical and Bit Operations

- **bit** $\equiv$ **binary digit**: smallest unit of storage in computer memory, has two possible values — **true** (1) and **false** (0).
- **Boolean Variable**: program unit of storage that can contain one of two values — either **true** or **false**, and can thus be represented by a bit.
- **Bit Operations**: correspond to logical connectives: $\wedge$, $\vee$, $\oplus$, $\neg$
- **Bit String**: a sequence of zero or more bits. The **length** of the string is the number of bits in it.

  Bitwise **OR**, Bitwise **AND**, and Bitwise **XOR** can be applied to bit strings.

An Exercise

0101	1101	0011	p
1110	1011	0110	q
			bitwise OR
			bitwise AND
			bitwise XOR

1.2 Propositional Equivalences

Contingency: a proposition which is neither a **tautology** nor a **contradiction**.

p	$\neg p$	$p \vee \neg p$ tautology	$p \wedge \neg p$ contradiction
T	F	T	F
F	T	T	F

Logical Equivalence

Logically Equivalent: two compound propositions which always have the same truth value (given the same truth assignments to any Boolean Variables).

p	q	$p \wedge q$	$p \vee q$	$\neg(p \vee q)$	$\neg p$	$\neg q$	$\neg p \wedge \neg q$
T	T	T	T	F	F	F	F
T	F	F	T	F	F	T	F
F	T	F	T	F	T	F	F
F	F	F	F	T	T	T	T

On Worksheet Provided

Using a truth table, show that $p \wedge q$ is logically equivalent to $\neg[p \rightarrow (\neg q)]$

Using a truth table, show that $p \vee q$ is logically equivalent to $(\neg p) \rightarrow q$

Using a truth table, show that $p \vee (q \wedge r)$ is logically equivalent to
$(p \vee q) \wedge (p \vee r)$

Write as a proposition:

```
If I go to Harry's or go to the
country, I will not go shopping.
```

Begin by breaking the compound into separate propositions:

- **H** =
- **C** =
- **S** =

Then write as a compound proposition using **H**, **C**, and **S**:

Name that Term!

What is a proposition which

1. is always `true`? ____________________
2. is always `false`? ____________________
3. is neither `1.` nor `2.`? ____________________

- Two propositions, p and q, are logically equivalent if $p \leftrightarrow q$ is a tautology.
- We write $p \Leftrightarrow q$
- Example: $(p \rightarrow q) \wedge (q \rightarrow p) \Leftrightarrow p \leftrightarrow q$

To show a proposition is not a tautology, use an ***abbreviated*** truth table and

- try to find a ***counter example*** or to ***disprove*** the assertion
- search for a case where the proposition is false

Proving Logical Equivalence

Prove these expressions are logically equivalent:

$$(p \rightarrow q) \wedge (q \rightarrow p) \quad \Leftrightarrow \quad p \leftrightarrow q$$

When would they **not** be equivalent?

Case 1. left side false, right side true...
- Subcase a. $p \rightarrow q$ is false
- Subcase b. $q \rightarrow p$ is false

Case 2. left side true, right side false...
- Subcase a. p = **T**, q = **F**
- Subcase b. p = **F**, q = **T**

There are no more possibilities, so the two propositions must be logically equivalent.

Note, Table 5 in section 1.2 — these are important for simplifying propositions and proving logical equivalences.

The Porsche & The Tiger

A prisoner must make a choice between two doors: behind one is a beautiful red Porsche, and behind the other is a hungry tiger. Each door has a sign posted on it, but only one sign is true.

Door #1. In this room there is a Porsche and in the other room there is a tiger.

Door #2. In one of these rooms there is a Porsche and in one of these rooms there is a tiger.

With this information, the prisoner is able to choose the correct door... Which one is it?

In Review

$\sim p$	**negation** of p	truth value is opposite of p
$p \wedge q$	**conjunction**	true only when both p and q are true
$p \vee q$	**disjunction**	false only when both p and q are false
$p \rightarrow q$	**conditional**	false only when p is true and q is false
$p \leftrightarrow q$	**biconditional**	true only when p and q have the ***same*** truth value.

1.3 Predicates

- **Propositional Function** or **Predicate**: a generalization of a proposition which contains one or more variables.
- Predicates become propositions once every variable is **bound** by:
  - **Assigning it a value** from the **Universe of Discourse, U**, or
  - **Quantifying it**

Example 1

- Let $U = Z = \{\ldots,-2,-1,0,1,2,\ldots\}$, the integers, and let $P(x): x > 0$ be a predicate.
  It has no truth value until the variable x is bound.
- Examples of propositions where x is assigned a value:
  - $P(-3)$
  - $P(0)$
  - $P(3)$

  What is the truth value of each?

Example 2

- $P(y) \vee \neg P(0)$ is ***not*** a proposition.
  The variable y has not been bound.
- Let R be the 3-variable predicate:
  $R(x,y,z): x + y = z$

  What is the truth of:
  - $R(2,-1,5)$
  - $R(3,4,7)$
  - $R(x,3,z)$

Quantifiers

- **Quantifiers** are used to assert that a predicate
  - is true for every value in the Universe of Discourse,
  - is true for some value(s) in the Universe of Discourse, or
  - is true for one and only one value in the Universe of Discourse
- The **Universal quantification of $P(x)$** is the proposition that $P(x)$ is true ***for every x*** in the Universe of Discourse
- Universal quantification is written as: $\forall x \ P(x)$
- For example, let $U = \{1,2,3\}$. Then
  $\forall x \ P(x) \Leftrightarrow P(1) \wedge P(2) \wedge P(3)$.

- The statement ***Every math student studies hard.*** can be expressed as:

  $\forall x \ P(x)$

  if we let $P(x)$ denote the statement ***x studies hard***, and let $U = \{$***all math students***$\}$.

  We can also write this statment as:
  $\forall x \ (S(x) \rightarrow P(x))$

  if we let $S(x)$ denote the statement ***x is a math student***, and $P(x)$ and U are as before.

Existential Quantification

- **Existential quantification** asserts a proposition is true if and only if it is true for at least one value in the universe of discourse.
- The **Existential quantification of $P(x)$** is the proposition that $P(x)$ is true *for some x* in the Universe of Discourse
- Existential quantification is written as: $\exists x\ P(x)$
- For example, let $U = \{1, 2, 3\}$. Then $\exists\, x\ P(x) \Leftrightarrow P(1) \vee P(2) \vee P(3)$.
- **Unique Existential Quantification** asserts a proposition is true for *one and only one* $x \in U$, and is written $\exists\,!\,x\ P(x)$
- **Remember**: a predicate is *not* a proposition until *all* variables have been bound either by quantification or assignment of a value.

Equivalences Involving Negation

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

"*$P(x)$ is not true for all x*" **is logically equivalent to** "*there is some x for which $P(x)$ is not true*"

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

"*There is no x for which $P(x)$ is true*" **is logically equivalent to** "*$P(x)$ is not true for every x*"

- Distributing a negation operator across a quantifier changes a universal to an existential, and vice versa
- If there are multiple quantifiers, they are read from left to right

Examples

Let $U = \mathbb{R}$, the real numbers. Then consider $P(x, y) : xy = 0$

Which of the following are TRUE?

- $\forall\, x\ \forall\, y\ P(x, y)$
- $\forall\, x\ \exists\, y\ P(x, y)$
- $\exists\, x\ \forall\, y\ P(x, y)$
- $\exists\, x\ \exists\, y\ P(x, y)$

Suppose $P(x, y) : \frac{x}{y} = 1 \ldots$ now which are TRUE?

Conversions

Let $U = \{1, 2, 3\}$. Find an expression equivalent to:

$$\forall\, x\ \exists\, y\ P(x, y)$$

where the variables are bound by substitution instead of quantification.

We can expand from the inside out, or the ouside in...

Outside in, we get:

$\exists\, y\ P(1, y)\ \wedge\ \exists\, y\ P(2, y)\ \wedge\ \exists\, y\ P(3, y) \Leftrightarrow$

$[\ P(1, 1) \vee P(1, 2) \vee P(1, 3)\ ] \wedge$
$[\ P(2, 1) \vee P(2, 2) \vee P(2, 3)\ ] \wedge$
$[\ P(3, 1) \vee P(3, 2) \vee P(3, 3)\ ]$

English to symbols—I

Let $U = \{$ all EIU students $\}$, and
$F(x)$: x speaks French fluently
$C(x)$: x knows C++

- Someone can speak French and knows C++
  $\exists\, x\ (F(x) \wedge C(x))$
- Someone speaks French, but doesn't know C++
- Everyone can either speak French or knows C++
- No one speaks French or knows C++
- If a student knows C++, they can speak French

English to Symbols—II

Let $U = \{$ fleegles, snurds, thingamabobs $\}$, and
$F(x)$: x is a fleegle
$S(x)$: x is a snurd
$T(x)$: x is a thingamabob

- Everything is a fleegle
  $\forall\, x\ F(x) \Leftrightarrow \neg\exists\, x\ \neg F(x)$
- Nothing is a snurd
- All fleegles are snurds
- Some fleegles are thingamabobs
- No snurd is a thingamabob
- If any fleegle is a snurd then it's also a thingamabob

1.4 Sets

- **Set**: a collection or group of objects, **elements**, or **members**
- **Contain**: a set is said to **contain** its elements
- **Universal Set, U**: there must be an underlying universal set, U, either specifically stated or understood

Notation used to specify sets

- **list the elements between braces**; listing an object more than once does ***not*** change the set—ordering means nothing.
  $$S = \{a,b,c,d\} = \{b,c,a,d,d\}$$
- **specify by predicate**; here, S contains all elements from U which make the predicate P TRUE
  $$S = \{\, x \mid P(x) \,\}$$
- **brace notation with ellipses**; here, the negative integers:
  $$S = \{\ldots,-3,-2,-1\}$$

Common Universal Sets

- $\mathbb{R}$ — Real Numbers
- $\mathbb{N}$ — Natural or Counting Numbers: $\{0,1,2,3,\ldots\}$
- $\mathbb{Z}$ — Integers: $\{\ldots,-3,-2,-1,0,1,2,3,\ldots\}$
- $\mathbb{Z}^+$ — Positive Integers

Notation

- $x \in S$ — x is a member of S, or X is an element of S
- $x \notin S$ — x is not an element of S

Subsets

- **Subset**: Let A and B be sets. Then
  $$A \subseteq B \;\Leftrightarrow\; \forall x \mid x \in A \rightarrow x \in B \mid$$
- **Empty, void,** or **Null Set**: $\varnothing$ is the set with no members
  - the assertion $x \in \varnothing$ is ***always*** FALSE, thus
  - $\forall x \mid x \in \varnothing \rightarrow x \in B \mid$ is always (vacuously) TRUE, therefore
  - $\varnothing$ is a subset of every set
  - **Note**: a set B is always a subset of itself
- **Proper subset**: $A \subset B$ if $A \subseteq B$, but $A \neq B$

Power Set

- **Power Set**: $\mathscr{P}(A)$ is the set of ***all*** possible subsets of the set A
- If $A = \{a, b\}$, then
  $$\mathscr{P}(A) = \{\varnothing, \{a\}, \{b\}, \{a, b\}\}$$
- What is the power set of the set $B = \{0, 1, 2\}$?
- How many elements would $\mathscr{P}(\{a, b, c, d\})$ have?

Cardinality

- **Cardinality**: $|A|$ is the number of distinct elements in A
- If the cardinality is a natural number (in $\mathbb{N}$), then the set is called **finite**; otherwise, it's called **infinite**
- **Example**: Let $A = \{a, b\}$
  - $|\{a, b\}| = 2$
  - $|\mathscr{P}(A)| = |\mathscr{P}(\{a, b\})| = 4$
  - A is finite, and so is $\mathscr{P}(A)$
- **Note**: $|A| = n \rightarrow |\mathscr{P}(A)| = 2^n$
- **Note**: $\mathbb{N}$ is infinite since $|\mathbb{N}|$ is not a natural number; it is called a **transfinite cardinal number**
- **Note**: Sets can be both **members** and **subsets** of other sets

Example

- Let $A = \{\varnothing, \{\varnothing\}\}$
  - A has two elements and hence four subsets:
    $\varnothing$, $\{\varnothing\}$, $\{\{\varnothing\}\}$, $\{\varnothing, \{\varnothing\}\}$
  - Note that $\varnothing$ is both a member of A and a subset of A
- **Russell's Paradox**: Let S be the set of all sets which are not members of themselves.
  Is S a member of itself or not?
- **The Paradox of the Barber of Seville**: The (male) barber of Seville shaves all and only men who do not shave themselves. Who shaves the barber of Seville?

Cartesian Product

- **Cartesian Product of A with B**: $A \times B$ is the set of **ordered** pairs:
  $$\{ \langle a, b \rangle \mid a \in A \wedge b \in B\}$$
- **Notation**: $\Pi_{i=1}^{n} A_i = \{\langle a_1, a_2, \ldots, a_n \rangle \mid a_i \in A_i\}$, an **n-tuple**
- The Cartesian product of any set with $\varnothing$ is $\varnothing$ — why?
- **Example 1**. Let $A = \{a, b\}$ and $B = \{1, 2, 3\}$

  $A \times B = \{\langle a,1\rangle, \langle a,2\rangle, \langle a,3\rangle, \langle b,1\rangle, \langle b,2\rangle, \langle b,3\rangle\}$

  What is $B \times A$?

  What is $A \times B \times A$?

  If $|A| = m$ and $|B| = n$, what is $|A \times B|$?

1.5 Set Operations

- **Boolean Algebra**: an algebraic system, instances of which are **propositional calculus** and **set theory**.
- The operators in set theory are defined in terms of the corresponding operator in propositional calculus.
- As before, there must be a universe, U, and all sets are assumed to be subsets of U.

Equality of Sets

- Two sets A and B are **equal**, denoted $A = B$, IFF
  $$\forall x\,[x \in A \leftrightarrow x \in B]$$
- **Note**: By a previous logical equivalence, we have:
  $$A = B$$
  IFF
  $$\forall x\,[(x \in A \rightarrow x \in B) \wedge (x \in B \rightarrow x \in A)]$$
  — or —
  $$A = B \text{ IFF } A \subseteq B \text{ and } B \subseteq A$$

Set Operations

- **Union** of A and B, denoted $A \cup B$, is the set
  $$\{x \mid x \in A \vee x \in B\}$$
- **Intersection** of A and B, denoted $A \cap B$, is the set
  $$\{x \mid x \in A \wedge x \in B\}$$
  If the intersection is void, A and B are said to be **disjoint**
- **Complement** of A, denoted $\overline{A}$, is the set
  $$\{x \mid \neg(x \in A)\}$$

More Set Operations

- **Difference** of A and B, or the ***complement*** of B ***relative to*** A, denoted $A - B$, is the set
  $$A \cap \overline{B}$$
  Note: The absolute complement of A is $U - A$
- **Symmetric Difference** of A and B, denoted $A \oplus B$, is the set
  $$(A - B) \cup (B - A)$$

Examples

$$U = \{0,1,2,3,4,5,6,7,8,9,10\}$$

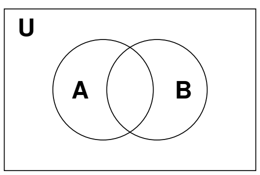
$$A = \{1,2,3,4,5\} \quad \text{and} \quad B = \{4,5,6,7,8\}$$

- $A \cup B = \{1,2,3,4,5,6,7,8\}$
- $A \cap B = \{4,5\}$
- $\overline{A} = \{0,6,7,8,9,10\}$
- $\overline{B} = \{0,1,2,3,9,10\}$
- $A - B = \{1,2,3\}$
- $B - A = \{6,7,8\}$
- $A \oplus B = \{1,2,3,6,7,8\}$

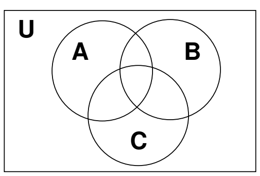
Venn Diagrams

Venn Diagrams are a useful geometric visualization tool for 3 or fewer sets.

- The Universal set U is a rectangular box
- Each set is represented by a circle and its interior
- All possible combinations of the sets must be represented
- Shade the appropriate region to represent the given set operation



For 2 sets



For 3 sets

Set Identities

Set identities correspond to the logical equivalences.

Example

The complement of the union is the intersection of the complements:

$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

To prove this statement, we need to show:

$$\forall x\, [x \in \overline{A \cup B} \leftrightarrow x \in \overline{A} \cap \overline{B}]$$

To show two sets are equal, we show for all x that x is a member of one set if and only if it is a member of the other.

Universal Instantiation

We now apply an important **rule of inference** (defined later) called

> ***Universal Instantiation***
>
> In a proof, we can eliminate the universal quantifier which binds a variable if we do not assume anything about the variable other than it is an arbitrary member of the Universe. We can then treat the resulting predicate as a proposition.

We say, **"Let x be arbitrary."** Then we can treat the predicates as propositions.

Assertion	Reason
$x \in \overline{A \cup B} \Leftrightarrow x \notin [A \cup B]$	Defn of complement
$x \notin A \cup B \Leftrightarrow \neg[x \in A \cup B]$	Defn of $\notin$
$\Leftrightarrow \neg[x \in A \vee x \in B]$	Defn of union
$\Leftrightarrow \neg x \in A \wedge \neg x \in B$	DeMorgan's Laws
$\Leftrightarrow x \notin A \wedge x \notin B$	Defn of $\notin$
$\Leftrightarrow x \in \overline{A} \wedge x \in \overline{B}$	Defn of complement
$\Leftrightarrow x \in \overline{A} \cap \overline{B}$	Defn of intersection

Hence, $x \in \overline{A \cup B} \leftrightarrow x \in \overline{A} \cap \overline{B}$ is a tautology since

- x was arbitrary
- we have used only logically equivalent assertions and definitions

Universal Generalization

We can apply another rule of inference

Universal Generalization

We can apply a universal quantifier to bind a variable if we have shown the predicate to be true for all values of the variable in the Universe

and claim the assertion is true for all x, i.e.,

$$\forall x[x \in \overline{A \cup B} \leftrightarrow x \in \overline{A} \cap \overline{B}]$$

Q.E.D. — an abbreviation for the Latin phrase ***"Quod Erat Demonstrandum"*** – "which was to be demonstrated" – used to signal the end of a proof.

Alternative Identity

Note: as an alternative which might be easier in some cases, use the identity:

$$A = B \Leftrightarrow [A \subseteq B \text{ and } B \subseteq A]$$

Example

Show $A \cap (B - A) = \emptyset$

The empty set is a subset of every set. Hence,

$$A \cap (B - A) \supseteq \emptyset$$

Therefore, it suffices to show

$$A \cap (B - A) \subseteq \emptyset$$

or

$$\forall x[x \in A \cap (B - A) \rightarrow x \in \emptyset]$$

So, as before, we say **"let x be arbitrary"**

Now we need to show

$$x \in A \cap (B - A) \rightarrow x \in \emptyset$$

is a tautology.

But the consequent is always false.

Therefore, the antecedent had better always be false also.

We procede by applying the definitions:

Assertion	Reason
$x \in A \cap (B - A) \Leftrightarrow x \in A \wedge x \in (B - A)$	Defn of intersection
$\Leftrightarrow x \in A \wedge (x \in B \wedge x \notin A)$	Defn of difference
$\Leftrightarrow (x \in A \wedge x \notin A) \wedge x \in B$	Properties of AND
$\Leftrightarrow 0 \wedge x \in B$	Table 6 in textbook
$\Leftrightarrow 0$	Domination

Hence, because $P \wedge \neg P$ is always false, the implication is a tautology. The result follows by Universal Generalization. Q.E.D.

Indexed Collections

Let $A_1, A_2, \ldots, A_n$ be an indexed collection of sets. Union and intersection are associative (because AND nad OR are), we have:

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

and

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

Examples

Let $A_i = [\, i, \infty), \ 1 \leq i < \infty$

$$\bigcup_{i=1}^{n} A_i = [1, \infty)$$

$$\bigcap_{i=1}^{n} A_i = [n, \infty)$$

1.6 Functions

- **Function**: Let A and B be sets. Then a function (mapping, map) f from A to B, denoted $f : A \rightarrow B$, is a subset of $A \times B$ such that

$$\forall x\, [x \in A \rightarrow \exists y\, [\, y \in B \ \wedge \ \langle x, y \rangle \in f \,]]$$

and

$$[\langle x, y_1 \rangle \in f \ \wedge \ \langle x, y_2 \rangle \in f\,] \rightarrow y_1 = y_2$$

- Note: f associates with each x in A **one and only one** y in B.
- A is called the **domain** of f
- B is called the **codomain** of f

- If $F(x) = y$:
  - y is called the **image** of x under f
  - x is called a **preimage** of y
- Note: there may be more than one preimage of y, but there is only **one** image of x.
- The **range** of f is the set of all images of points in A under f; it is denoted by $f(A)$

Injections, Surjections, and Bijections

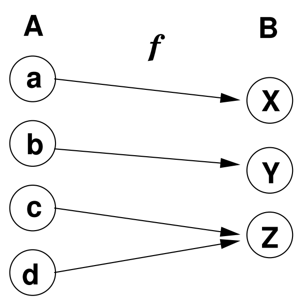
Let f be a function from A to B

- **Injection**: f is **one-to-one** (denoted 1–1) or **Injective** if preimages are unique

  Note: this means that if $a \neq b$ then $f(a) \neq f(b)$
- **Surjection**: f is **onto** or **surjective** if every y in B has a preimage

  Note: this means that for every y in B there must be an x in A such that $f(x) = y$
- **Bijection**: f is **bijective** if it is surjective and injective, in other words, **1–1 and onto**.

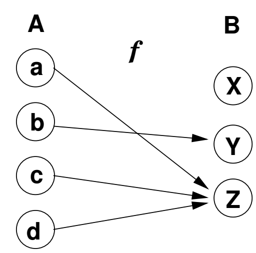
Example I



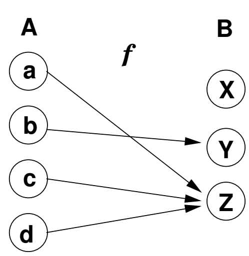
1. Is this an injection?
2. Is this a surjection?
3. Is this a bijection?
4. How do we determine the answers?

Example II

If S is a subset of A, then $f(S) = \{f(s) | s \in S\}$

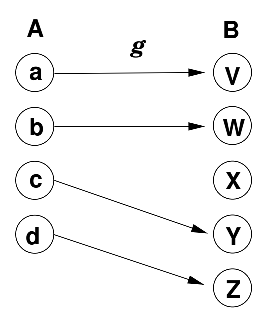


- $f(a) = Z$
- the image of d is Z
- the domain of f is $A = \{a, b, c, d\}$



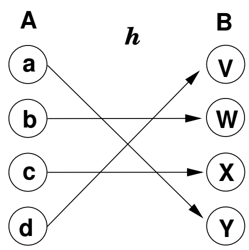
- the codomain is $B = \{X, Y, Z\}$
- the range, $f(A) = \{Y, Z\}$
- the preimage of Y is b
- the preimages of Z are a, c, and d
- $f(\{c, d\}) = \{Z\}$

Example III



1. Is this an injection?
2. Is this a surjection?
3. Is this a bijection?

Example IV



1. Is this an injection?
2. Is this a surjection?
3. Is this a bijection?

Cardinality

- **Note**: whenever there is a bijection from A to B, the two sets **must have the same number of elements** or the same **cardinality**
- That will become our definition, especially for infinite sets.

Let $A = B = \mathbb{R}$, the reals
Determine which are injections, surjections, bijections:

- $f(x) = x$
- $f(x) = x^2$
- $f(x) = x^3$
- $f(x) = |x|$

Let E be the set of even nonnegative integers, $\{0, 2, 4, 6, \ldots\}$

Then there is a bijection f from $\mathbb{N}$ to E, the even nonnegative integers, defined by:

$$f(x) = 2x$$

Hence, the set of even nonnegative integers has the **same** cardinality as the set of natural numbers...

OH, NO! IT CAN'T BE... E is only half as big!!

Inverse Functions

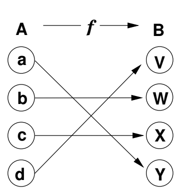
- **Inverse Function**: Let f be a bijection from A to B. Then the **inverse** of f, denoted f^{-1}, is the function from B to A defined as:

$$f^{-1}(y) = x \text{ IFF } f(x) = y$$

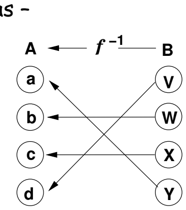
- Note: no inverse exists unless f is a bijection.

Example

Let f be defined by the diagram –



Then f^{-1} is defined as –

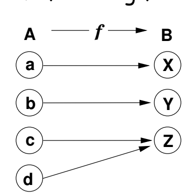


Inverse Applied to a Subset

- **Inverse Function over Subsets**: Let S be a subset of B. Then

$$f^{-1}(S) = \{x \mid f(x) \in S\}$$

- Example: Let f be the following function –



$$f^{-1}(\{Z\}) = \{c, d\}$$

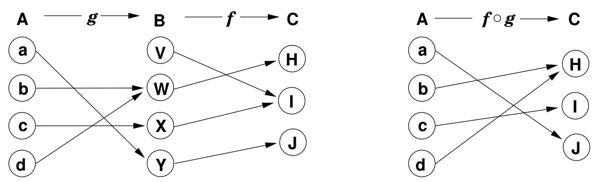
$$f^{-1}(\{X, Y\}) = \{a, b\}$$

Composition

- **Composition**: Let $f : B \to C$, $g : A \to B$. The composition of f with g, denoted $f \circ g$, is the function from A to C defined by

$$f \circ g(x) = f(g(x))$$

- Example:



Another Example

Let $f(x) = x^2$ and $g(x) = 2x + 1$

Then $f \circ g(x) = (2x+1)^2 = 4x^2 + 4x + 1$

And $g \circ f(x) = 2x^2 + 1$

Discussion

- Suppose $f : B \to C$, $g : A \to B$, and $f \circ g$ is injective
- What can we say about f and g?
- Using the definition of **injective**, we know that if $a \neq b$, then $f(g(a)) \neq f(g(b))f$, since the composition is injective
- Since f is a function, it cannot be the case that $g(a) = g(b)$, since f would have two different images for the same point.
- Hence, $g(a) \neq g(b)$
- It follows that g must be an injection
- However, f need not be an injection... how could you show this?

FLOOR and CEILING Functions

- **Floor**: The FLOOR function, denoted

  $f(x) = \lfloor x \rfloor$ or $f(x) = \text{FLOOR}(x)$

  is the largest integer less than or equal to x.

- **Ceiling**: The CEILING function, denoted

  $f(x) = \lceil x \rceil$ or $f(x) = \text{CEILING}(x)$

  is the smallest integer greater than or equal to x.

- Examples: $\lfloor 3.5 \rfloor = 3$, $\lceil 3.5 \rceil = 4$

- Note: The FLOOR function is equivalent to truncation for positive numbers

1.7 Sequences, Summations, and Cardinality of Infinite Sets

- **Sequence**: A sequence is a function from a subset of the natural numbers (usually of the form $\{0, 1, 2, \dots\}$ to a set S

- The sets

$$\{0, 1, 2, 3, \dots, k\}$$

  and

$$\{1, 2, 3, \dots, k\}$$

  are called initial segments of $\mathbb{N}$

- Notation: if f is a function from $\{0, 1, 2, \dots\}$ to S, we usually denote $f(i)$ by a_i and we write:

$$\{a_0, a_1, a_2, a_3, \dots\} = \{a_i\}_{i=0}^{k} = \{a_i\}_0^k$$

  where k is the upper limit (usually ∞)

Sequence Examples

- Using zero-origin indexing, if $f(i) = \frac{1}{(i+1)}$, then the sequence

$$f = \{1, \tfrac{1}{2}, \tfrac{1}{3}, \tfrac{1}{4}, \dots\} = \{a_0, a_1, a_2, a_3, \dots\}$$

- Using one-origin indexing, the sequence f becomes

$$f = \{\tfrac{1}{2}, \tfrac{1}{3}, \tfrac{1}{4}, \dots\} = \{a_1, a_2, a_3, \dots\}$$

Summation Notation

Given a sequence $\{a_i\}_0^k$ we can add together a subset of the sequence by using the summation and function notation

$$a_{g(m)} + a_{g(m+1)} + \cdots + a_{g(n)} = \sum_{j=m}^{n} a_{g(j)}$$

or more generally

$$\sum_{j \in S} a_j$$

Examples

- $r^0 + r^1 + r^2 + r^3 + \ldots r^n = \sum_{j=0}^{n} r^j$
- $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \ldots = \sum_{i=1}^{\infty} \frac{1}{i}$
- $a_{2m} + a_{2(m+1)} + \cdots + a_{2(n)} = \sum_{j=m}^{n} a_{2j}$
- If $S = \{2, 5, 7, 10\}$, then $\sum_{j \in S} a_j = a_2 + a_5 + a_7 + a_{10}$

What are these sums?

- $\sum_{i=0}^{1} i^2 =$
- $\sum_{i=0}^{3} i^2 =$
- $\sum_{j=-1}^{1} 2^j =$
- $\sum_{k=3}^{5} (-1)^k =$

Product Notation

Similarly for multiplying together a subset of a sequence

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

Geometric Progression

Geometric Progression: a sequence of the form:

$$a, ar, ar^2, ar^3, ar^4, \ldots$$

There's a proof in the textbook that

$$\sum_{i=0}^{n} r^i = \frac{r^{n+1}-1}{r-1} \text{ if } r \neq 1$$

You should be able to determine the sum:

- if $r = 0$
- if the index starts at k instead of 0
- if the index ends at something other than n (e.g., $n-1$, $n+1$, etc.)

Cardinality and Countability

Cardinality: The cardinality of a set A is equal to the cardinality of a set B, denoted $|A| = |B|$, if there exists a bijection from A to B.

Countable: A set is countable if it has the same cardinality as a subset of the natural numbers, $\mathbb{N}$

If $|A| = |\mathbb{N}|$, the set A is ***countably infinite***.

The (transfinite) cardinal number of the set $\mathbb{N}$ is

aleph null $= \aleph_0$

If a set is not countable, we say it is ***uncountable***

Uncountable Set Examples

The real numbers in the closed interval $[0, 1]$

$\mathscr{P}(\mathbb{N})$, the power set of $\mathbb{N}$

Note: with infinite sets, proper subsets can have the same cardinality. This cannot happen with finite sets

Countability carries with it the implication that there is a *listing* of the elements of the set

Definition: $|A| \le |B|$ if there is an injection from A to B.

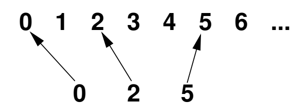
Theorem. If $|A| \le |B|$ and $|B| \le |A|$, then $|A| = |B|$.
This implies

- if there is an injection from A to B and
- if there is an injection from B to A

then

- there must be a bijection from A to B

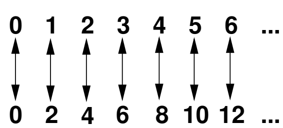
- This is **difficult** to prove, but is an example of demonstrating existence without construction.
- It is often easier to build the injections and then conclude the bijection exists.
- Example I.
  Theorem: If A is a subset of B, then $|A| \le |B|$.
  Proof: the function $f(x) = x$ is an injection from A to B
- Example II. $|\{0, 2, 5\}| \le \aleph_0$
  The injection $f\{0, 2, 5\} \to \mathbb{N}$ defined by $f(x) = x$ is:



Some Countably Infinite Sets

- The set of even integers $\mathbb{E}$ is countably infinite...
  Note that $\mathbb{E}$ **is a proper subset of** $\mathbb{N}$

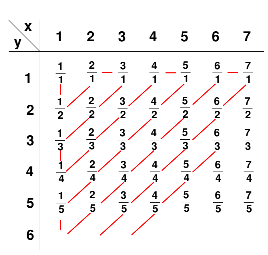
  Proof: Let $f(x) = 2x$. Then f is a bijection from $\mathbb{N}$ to $\mathbb{E}$



- $\mathbb{Z}^+$, the set of positive integers is countably infinite
- The set of positive rational numbers, $\mathbb{Q}^+$, is countably infinite

Proof: $\mathbb{Q}^+$ is countably infinite

- $\mathbb{Z}^+$ is a subset of $\mathbb{Q}^+$, so $|\mathbb{Z}^+| = \aleph_0 \le |\mathbb{Q}^+|$
- Next, we must show that $|\mathbb{Q}^+| \le \aleph_0$.
- To do this, we show that the positive rational numbers with repetitions, $\mathbb{Q}_\mathbb{R}$, is countably infinite.
- Then, since $\mathbb{Q}^+$ is a subset of $\mathbb{Q}_\mathbb{R}$, it follows that $|\mathbb{Q}^+| \le \aleph_0$, and hence $|\mathbb{Q}^+| = \aleph_0$



- The position on the path (listing) indicates the image of the bijection function f from $\mathbb{N}$ to $\mathbb{Q}_\mathbb{R}$:
  $f(0) = \frac{1}{1}$, $f(1) = \frac{1}{2}$, $f(2) = \frac{2}{1}$, $f(3) = \frac{3}{1}$, etc.
- Every rational number appears on the list at least once, some many times (repetitions).
- Hence, $|\mathbb{N}| = |\mathbb{Q}_\mathbb{R}| = \aleph_0$

More Examples of Countably Infinite

- The set of all rational numbers, $\mathbb{Q}$, positive and negative, is countably infinite.
- The set of (finite length) strings S over a finite alphabet A is countably infinite.
  To show this, we assume that:
  - A is non-void
  - There is an "alphabetical" ordering of the symbols in A

  Proof: List the strings in lexicographic order —
  - all the strings of zero length
  - then all the strings of length 1 in alphabetical order,
  - then all the strings of length 2 in alphabetical order,
  - etc.

  This implies a bijection from $\mathbb{N}$ to the list of strings and hence it is a countably infinite set

String Example

Let $A = \{a, b, c\}$

Then the lexicographic ordering of A is:

$$\{\Lambda, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, aab, aac, aba, \ldots\}$$

$$= \{f(0), f(1), f(2), f(3), f(4), \ldots\}$$

The Set of All C++ Programs is **countable**

Proof: Let S be the set of legitimate characters which can appear in a C++ program.

- A C++ compiler will determine if an input program is a syntactically correct C++ program (the program doesn't have to do anything useful).
- Use the lexicographic ordering of S and feed the strings into the compiler.
- If the compiler says YES, this is a syntactically correct C++ program, we add the program to the list.
- Else, we move on to the next string

In this way we construct a list or an implied bijection from $\mathbb{N}$ to the set of C++ programs.

Hence, the set of C++ programs is countable.

Cantor Diagonalization

An important technique used to construct an object which is **not** a member of a countable set of objets with (possibly) infinite descriptions

Theorem: The set of real numbers between 0 and 1 is uncountable.

Proof: We assume that it is countable and derive a **contradiction**.

If the set is countable, we can list all the real numbers (i.e., there is a bijection from a subset of $\mathbb{N}$ to the set).

We show that no matter what list you produce we can construct a real number between 0 and 1 which is not in the list.

Hence, the number we constructed cannot exist in the list and therefore the set is not countable.

It's actually much bigger than countable — it's said to have the **cardinality of the continuum**, $\mathfrak{c}$

Represent each real number in the list using ***its decimal expansion***

E.g., $\frac{1}{3} = 0.3333333\ldots\ldots$

$\frac{1}{2} = 0.5000000\ldots\ldots$

$= 0.4999999\ldots\ldots$

If there is more than one expansion for a number, it doesn't matter as long as our construction takes this into account.

The resulting list:

$$\begin{aligned} r_1 &= 0.d_{11}d_{12}d_{13}d_{14}d_{15}d_{16}\ldots\ldots \\ r_2 &= 0.d_{21}d_{22}d_{23}d_{24}d_{25}d_{26}\ldots\ldots \\ r_3 &= 0.d_{31}d_{32}d_{33}d_{34}d_{35}d_{36}\ldots\ldots \\ &\vdots \end{aligned}$$

Now, **construct the number** $x = 0.x_1x_2x_3x_4x_5x_6x_7\ldots\ldots$

$$x_i = 3 \text{ if } d_{ii} \neq 3$$
$$x_i = 4 \text{ if } d_{ii} = 3$$

Note: choosing 0 and 9 is not a good idea because of the non-uniqueness of decimal expansions.

Then x is not equal to any number in the list.

Hence, no such list can exist, and thus the interval $(0, 1)$ is uncountable.

Computability

Computable: A number x between 0 and 1 is **computable** if there is a C++ program which, when given the input i, will produce the i^{th} digit in the decimal expansion of x.

Example: The number $\frac{1}{3}$ is computable.

The C++ program which always outputs the digit 3, regardless of the input, computes the number

Some Things are Not Computable

Theorem. There exists a number x between 0 and 1 which is **not computable**.

There **does not exist** a C++ program (or a program in any other computer language) which will compute it!

Why? Because there are more numbers between 0 and 1 than there are C++ programs to compute them.

(In fact, there are $\mathfrak{c}$ such numbers!)

Yet another example of the non-existence of programs to compute things!

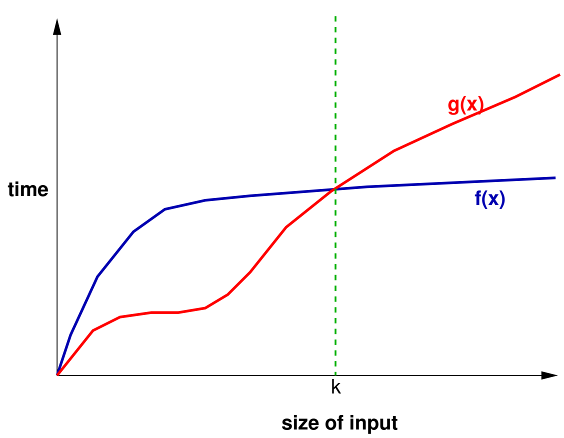
1.8 Growth of Functions

- **Time Complexity** is a measure of the computational "steps" of an algorithm relative to the size of input
- Algorithms are analyzed to see how the number of computational steps grows in relation to the size of input, n
- Once we have a function to compute the **time complexity** of algorithms which solve the same problem, we can compare them to determine which is more efficient
- For example, if the time it takes one sorting algorithm to sort n values is
$$T_1(n) = \frac{3}{2}n^2 + 3n$$
and another takes time
$$T_2(n) = 5n \log n + 29$$
which algorithm should we implement?

Comparing Function Growth

- Quantify the concept: **g grows at least as fast as f**
- What really matters when comparing the complexity of algorithms?
  - We mostly care about the behavior for **large** problems (i.e., what happens for "sufficiently large" input sizes)
  - Even bad algorithms can be used to solve "sufficiently small" problems
  - We can ignore some implementation details such as loop counter incrementation — we can straight-line any loop, etc.
- Remember, the functions we're discussing represent the **time complexities** of algorithms.

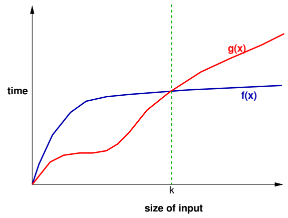
$f \in O(g)$



Big-Oh Notation

- **g asymptotically dominates f**:
Let f and g be functions from $\mathbb{N}$ to $\mathbb{R}$.
Then $f \in O(g)$ — or f is Big-Oh of g, or f is order g — IFF
$$\exists k \; \exists C \; \forall n[n > k \rightarrow |f(n)| \leq C|g(n)|, \;\; k, C > 0]$$
- For sufficiently large n, the function f is bounded from above by a positive, constant multiple of the function g.
- Alternatively:
$$\text{If } \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \text{ then } f \in o(g)$$
(f is Little-Oh of g, or strictly bounded by)

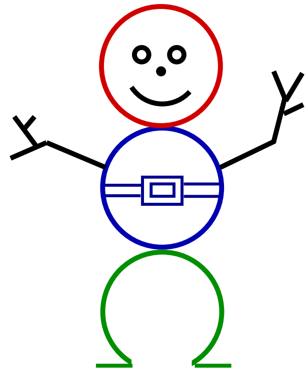
Proving Asymptotic Domination



To prove $f \in O(g)$, given f and g:

- Determine / choose some positive k
- Determine / choose a positive C (may depend upon choice of k)
- Once k and C are chosen, the implication must be proven true

Three Important Complexity Classes



Bounds from above
Big – Oh

Growth rates equivalent
Theta

Bounds from below
Big – Omega

Omega Man

Complexity Classes

The sets $O(g)$, $o(g)$, $\Omega(g)$, $\omega(g)$, and $\Theta(g)$ are called **complexity classes**.

$O(g)$ is a **set** which contains all the functions which g ***dominates***.

f is $O(g)$ means $f \in O(g)$

We say $f \in \Omega(g)$ if there are positive constants k and C such that $F(n) \geq Cg(n)$ whenever $n > k$

If $f \in O(g)$ and $f \in \Omega(g)$, then $f \in \Theta(g)$

We use "little-oh" and "little-omega" when we have strict inequality

Example

Let $f(n) = 4n + 5$ and $g(n) = n^2$.
We wish to show that $f \in O(g)$

We need to find k and C, and show the implication

$$\forall n > k, \quad f(n) \leq Cg(n)$$

is true for the values we choose.

To find k, we can set the functions equal, and solve for n:

$$\begin{aligned} 4n + 5 &= n^2 \\ 0 &= n^2 - 4n - 5 \\ 0 &= (n - 5)(n + 1) \end{aligned}$$

So, $n = 5$ or $n = -1$ … but n is the size of input and therefore cannot be negative. Thus, k must be at least 5.

If we choose $K = 6$, then C can be any positive number greater than or equal to 1.

All that is left is the proof that $\forall n > k, \; f(n) \leq Cg(n)$, which we shall revisit when we discuss induction proofs.

Big-Oh Properties

- f is $O(g)$ IFF $O(f) \subseteq O(g)$
- If $f \in O(g)$ and $g \in O(f)$, then $O(f) = O(g)$
- The set $O(g)$ is ***closed under addition***:
  If $f \in O(g)$ and $h \in O(g)$, then $f + h \in O(g)$

- $O(g)$ is ***closed under multiplication by a scalar*** $a \in \mathbb{R}$:

  If $f \in O(g)$ then $af \in O(g)$

  I.e., $O(g)$ is a ***vector space***

- Also, as you would expect,

  If $f \in O(g)$ and $g \in O(h)$, then $f \in O(h)$

  In particular,

  $$O(f) \subseteq O(g) \subseteq O(h)$$

Theorem

If $f_1 \in O(g_1)$ and $f_2 \in O(g_2)$, then:

1. $f_1 f_2 \in O(g_1 g_2)$
2. $f_1 + f_2 \in O(\max\{g_1, g_2\})$

Functional Values for Small n

$\log_2 n$	$\sqrt{n}$	n	n^2	2^n	$n!$	n^n
0	1.0000	1	1	2	1	1
1.0000	1.4142	2	4	4	2	4
1.5850	1.7321	3	9	8	6	27
2.0000	2.0000	4	16	16	24	256
2.3219	2.2361	5	25	32	120	3125
2.5850	2.4495	6	36	64	720	46,656
2.8074	2.6458	7	49	128	5040	823,543
3.0000	2.8284	8	64	256	40,320	1.67×10^7
3.1699	3.0000	9	81	512	362,880	3.87×10^8
3.3219	3.1623	10	100	1024	3.6×10^6	10^{10}

Approximate Functional Values for Powers of n

$\log_2 n$	$\sqrt{n}$	n	n^2	2^n	$n!$	n^n
3.32	3.16	10^1	10^2	1024	$3.63(10^6)$	10^{10}
6.64	10	10^2	10^4	$1.27(10^{30})$	$9.3(10^{157})$	10^{200}
9.97	31.62	10^3	10^6	$1.07(10^{301})$	$4(10^{2567})$	10^{3000}
13.29	100	10^4	10^8	$2(10^{3010})$	$2.9(10^{35,659})$	$10^{40,000}$
16.61	316.2	10^5	10^{10}	$10^{30,103}$	$2.9(10^{456,573})$	$10^{500,000}$
19.93	1000	10^6	10^{12}	$10^{301,030}$	$8.3(10^{5,565,708}$	$10^{60,000,000}$
39.86	10^6	10^{12}	10^{24}	BIG	LARGE	HUGE

Important Complexity Classes

Theorem. The hierarchy of several familiar sequences in the sense that each sequence is Big-Oh of any sequence to its right:

$$1,\ \log_2 n,\ \ldots,\ \sqrt[4]{n},\ \sqrt[3]{n},\ \sqrt{n},\ n,\ n\log_2 n,\ n\sqrt{n},\ n^2,\ n^3,\ n^4,\ \ldots,\ 2^n,\ n!,\ n^n$$

Similarly, stated in set notation:

$$O(c) \subseteq O(\log n) \subseteq O(n) \subseteq O(n \log n) \subseteq O(n^2) \subseteq O(n^j) \subseteq O(c^n) \subseteq O(n!)$$

where $j > 2$ and $c > 1$

Time Equivalencies

SECOND	1	10^0
MILLISECONDS	1,000	10^3
MICROSECONDS	1,000,000	10^6
NANOSECONDS	1,000,000,000	10^9

Largest Problem Sizes

Let $f(n)$ be the time complexity of an algorithm in MICROSECONDS.

The largest problem of size n that can be solved in:

1 SECOND	IS	$f(n)/10^6$
1 MINUTE	IS	$f(n)/(60*10^6)$
1 HOUR	IS	$f(n)/(60*60*10^6)$
1 DAY	IS	$f(n)/(24*60*60*10^6)$
1 MONTH	IS	$f(n)/(\mathbf{30}*24*60*60*10^6)$
1 YEAR	IS	$f(n)/(12*\mathbf{30}*24*60*60*10^6)$
1 CENTURY	IS	$f(n)/(100*12*\mathbf{30}*24*60*60*10^6)$

Largest Problems "Do-able" in 1 Second

1. Let $f(n) = n$. Then the largest problem for which we can compute an answer in one second is:
   $$n/10^6 = 1$$
   $$n = 10^6$$
2. Let $f(n) = n^2$. Then the largest problem for which we can compute an answer in one second is:
   $$n^2/10^6 = 1$$
   $$n = \sqrt{10^6} = 10^3$$
3. Let $f(n) = {}^n2$. Then the largest problem for which we can compute an answer in one second is:
   $$2^n/10^6 = 1$$
   $$2^n = 10^6 \approx 2^{19}$$
   $$n \approx 19$$

Let $f(n) = n!$. Then the largest problem for which we can compute an answer in one second is:

$$n!/10^6 = 1$$

$$n! = 10^6$$

Here, it helps to use a calculator..., and we find

$$9! = 362,880 \text{ — too small}$$

$$10! = 3,628,880 \text{ — too large}$$

$$n \approx 9$$

(Recall that n is input size)

Review — Exponents

$x^a x^b = x^{a+b}$	$2^5 2^7 = 2^{12}$
$\frac{x^a}{x^b} = x^{a-b}$	$\frac{3^8}{3^2} = 3^6$
$(x^a)^b = x^{ab}$	$(5^2)^3 = 5^6$

Notes

$x^n + x^n = 2x^n$ ...... **not** x^{2n}	$3^2 + 3^2 = 2(3^2)$
$5^3 + 5^3 =$	$2^7 + 2^7 =$
$2^n + 2^n =$	$2^{100} + 2^{100} =$
$3^4 + 5^4 =$	$2^n + 3^n =$

Review — Logarithms

Logarithm: $x^a = b$ IFF $\log_x b = a$

$2^3 = 8$ IFF $\log_2 8 = 3$

$5^4 = 625$ IFF log ______________________

__________ = __________ IFF $\log_3 81 = 4$

Theorem. $\log ab = \log a + \log b$

$$\begin{aligned} \log 32 &= \log(2^5) = 5 \\ &= \log(8*4) \\ &= \log 8 + \log 4 \\ &= \log 2^3 + \log 2^2 \\ &= 3 + 2 = 5 \ \checkmark \end{aligned}$$

Other Formulae You Should Know

- $\log \frac{a}{b} = \log a - \log b$
  - $\log \frac{32}{4} = \log \frac{2^5}{2^2} = \log 2^{5-2} = \log 2^3 = 3$ — and —
  - $\log \frac{32}{4} = \log 32 - \log 4 = \log 2^5 - \log 2^2 = 5 - 2 = 3$
  - Determine $\log \frac{1024}{64}$ both ways shown above
- $\log a^b = b \log a$
  - $\log 16^3 = \log(2^4)^3 = \log 2^{12} = 12$ — and —
  - $\log 16^3 = 3 \log 16 = 3 \log 2^4 = 3*4 = 12$
  - Determine $\log 128^5$ both ways shown above

Other General Knowledge

- $\log x < x \quad \forall x > 0 \qquad \log n < n \quad \forall n > 0$

- $\log 1 = 0,\ \ \log 2 = 1,\ \ \log 1024 = 10,$
  $\log 1{,}048{,}576 = 20$

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

  Thus, if $n = 3$:

  $\sum_{i=0}^{3} 2^i = 2^0 + 2^1 + 2^2 + 2^3 = 1 + 2 + 4 + 8 = 15$

  — and —

  $2^{n+1} - 1 = 2^{3+1} - 1 = 2^4 - 1 = 16 - 1 = 15$

- In general:

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

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

  Thus, $\sum_{i=1}^{5} i = 1 + 2 + 3 + 4 + 5 = 15$ — and —

  $\sum_{i=1}^{5} i = \frac{5(5+1)}{2} = \frac{5*6}{2} = \frac{30}{2} = 15$