Discrete Structures for Computer Science

William Garrison
bill@cs.pitt.edu
6311 Sennott Square

Lecture #8: Sets

University of Pittsburgh

Based on materials developed by Dr. Adam Lee

Today’s Topics

Introduction to set theory

- What is a set?
- Set notation
- Basic set operations

What is a set?

Definition: A set is an unordered collection of objects

Sets can contain items of mixed types

Examples:

- A = {1, 2, 3, 4}
- B = {Cooper, Dougie, Mr. C}
- C = {motorcycle, 3.14159, Socrates}
- E = {{1, 2, 3}, {6, 7, 8}, {23, 42}}

Sets can contain other sets

Informally: Sets are really just a precise way of grouping a "bunch of stuff"

A set is made up of elements

Definition: The objects making up a set are called elements of that set.

Examples:

- 3 is an element of {1, 2, 3}
- Bob is an element of {Alice, Bob, Charlie, Daniel}

We can express the above examples in a more precise manner as follows:

- $3 \in \{1, 2, 3\}$
- Bob $\in$ {Alice, Bob, Charlie, Daniel}

Question: Is $5 \in \{1, 2, 3, \{4, 5\}\}$?

There are many different ways to describe a set

Explicit enumeration:

- A = {1, 2, 3, 4}

Using ellipses if the general pattern is obvious:

- E = {2, 4, 6, …, 98}

Set builder notation (aka, set comprehensions):

- M = {y | y = 3k for some integer k}

The set M contains…

… all elements y…

… such that…

… y = 3k for some integer k

There are a number of sets that are so important to mathematics that they get their own symbol

$\mathbf{N} = \{0, 1, 2, 3, \ldots\}$

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

$\mathbf{Z}^+ = \{1, 2, \ldots\}$

$\mathbf{Q} = \{p/q \mid p,q \in \mathbf{Z}, q \neq 0\}$

$\mathbf{R}$

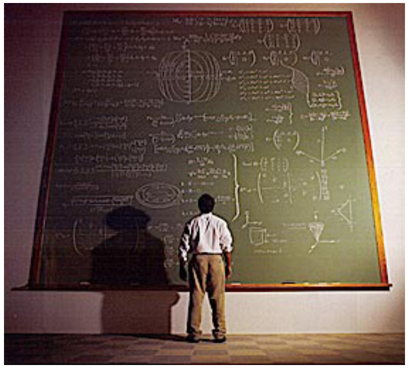
$\emptyset = \{\}$

Note: This notation differs from book to book

- Some authors write these sets as $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Z}^+$, $\mathbb{Q}$, and $\mathbb{R}$
  - I'll do so on the board ("blackboard bold")
- Some authors do not include zero in the natural numbers
  - I like the above because it makes $\mathbf{N} \neq \mathbf{Z}^+$ (more expressive)

Be careful when reading other books or researching on the Web, as things may be slightly different!

You've actually been using sets *implicitly* all along!



Mathematics

F(x,y) ≡ x and y are friends
Domain: "All people"

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

Domains of propositional functions

```
Function min(int x, int y) : int
    if x < y then
        return x
    else
        return y
    endif
end function
```

Programming language data types

Set equality

Definition: Two sets are equal if and only if they contain exactly the same elements.

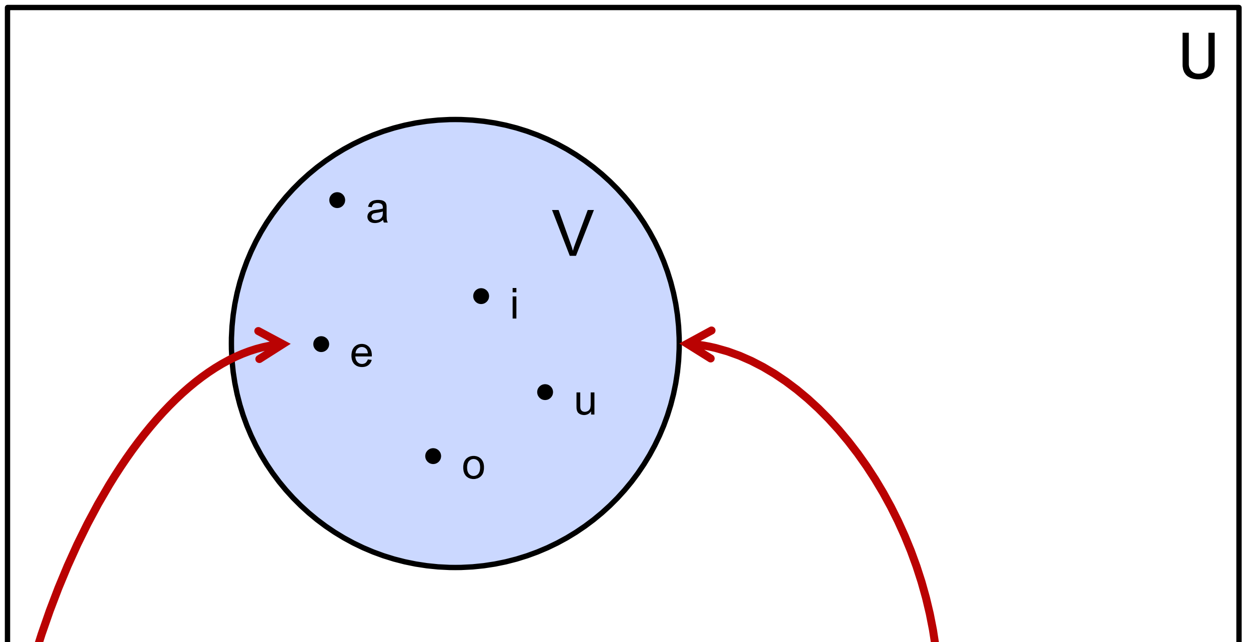
Mathematically: $A = B \text{ iff } \forall x\ (x \in A \Leftrightarrow x \in B)$

Example: Are the following sets equal?

- $\{1, 2, 3, 4\}$ and $\{1, 2, 3, 4\}$
- $\{1, 2, 3, 4\}$ and $\{4, 1, 3, 2\}$
- $\{a, b, c, d, e\}$ and $\{a, a, c, b, e, d\}$
- $\{a, e, i, o\}$ and $\{a, e, i, o, u\}$

We can use Venn diagrams to graphically represent sets

U is the "universe" of all elements



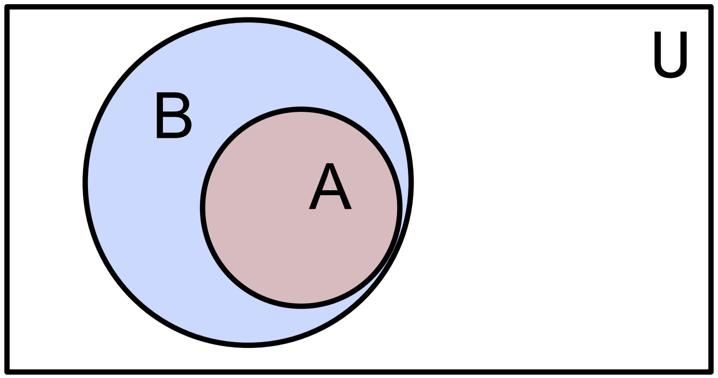
The set V of all vowels is contained within the universe of "all letters"

Sometimes, we add points for the elements of a set

Sets can be contained within one another

Definition: Some set A is a subset of another set B iff every element of A is an element in the set B. We denote this fact as $A \subseteq B$, and call B a superset of A.

Graphically:



Mathematically:

Definition: We say that A is a proper subset of B iff $A \subseteq B$, but $A \neq B$. We denote this by $A \subset B$. More precisely:

Properties of subsets

Property 1: For all sets S, we have that $\emptyset \subseteq S$

Proof: The set $\emptyset$ contains no elements. So, trivially, every element of the set $\emptyset$ is contained in any other set S. ❏

Property 2: For any set S, $S \subseteq S$.

Property 3: If $S_1 = S_2$, then $S_1 \subseteq S_2$ and $S_2 \subseteq S_1$.

In-class exercises

Problem 1: Come up with two ways to represent each of the following sets:

- The even integers
- Negative integers between -1 and -10, inclusive
- The positive integers

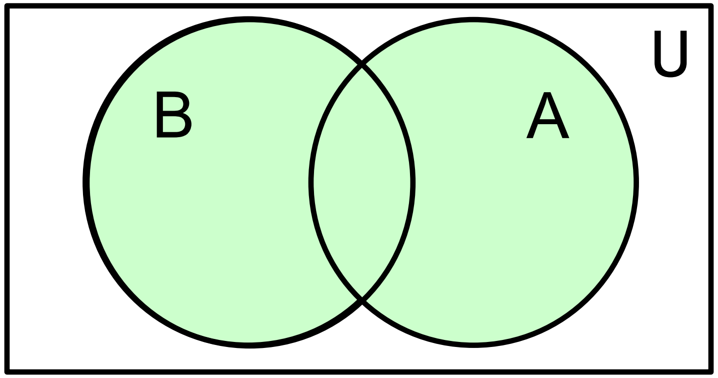
Problem 2: Are the sets {a, b, c} and {c, c, a, b, a, b} equal? Why or why not?

Problem 3: Draw a Venn diagram representing the sets {1, 2, 3} and {3, 4, 5}.

We can create a new set by combining two or more existing sets

Definition: The union of two sets A and B contains every element that is either in A or in B. We denote the union of the sets A and B as A ∪ B.

Graphically:



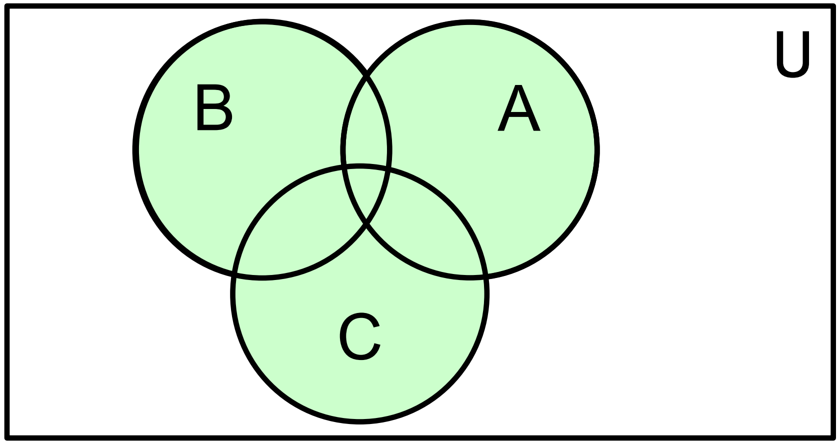
Mathematically:

Example: $\{1, 2, 3\} \cup \{6, 7, 8\} = \{1, 2, 3, 6, 7, 8\}$

We can take the union of any number of sets

Example: A ∪ B ∪ C

Graphically:



In general, we can express the union $S_1 \cup S_2 \cup \ldots \cup S_n$ using the following notation:

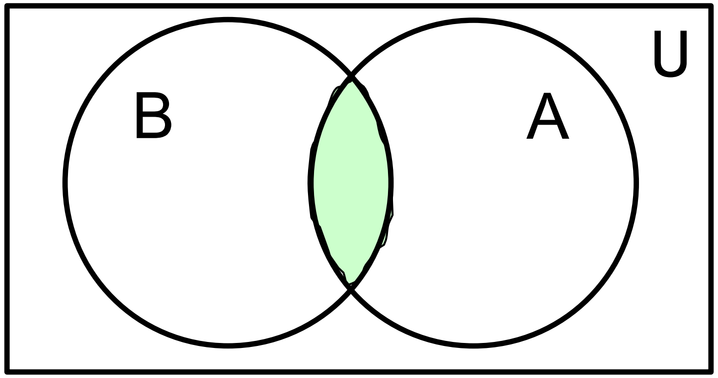
$$\bigcup_{i=1}^{n} S_i$$

This is just like summation notation!

Sometimes we're interested in the elements that are in more than one set

Definition: The intersection of two sets A and B contains every element that is in A and also in B. We denote the intersection of the sets A and B as $A \cap B$.

Graphically:



Mathematically:

Examples:

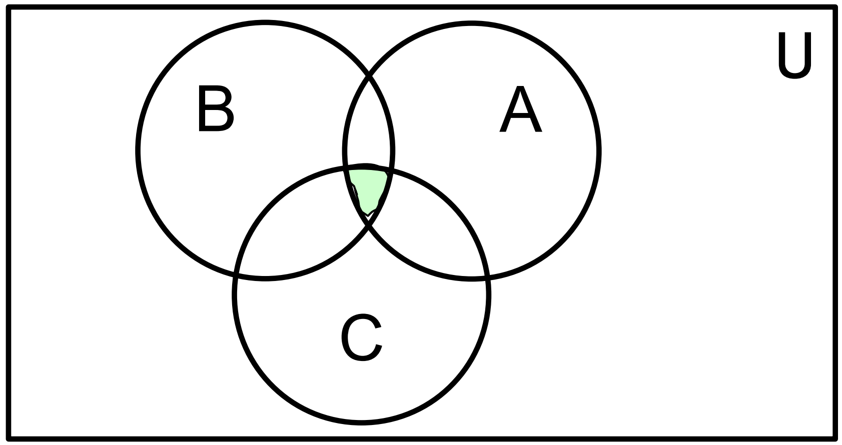
- $\{1, 2, 3, 7, 8\} \cap \{6, 7, 8\} = \{7, 8\}$
- $\{1, 2, 3\} \cap \{6, 7, 8\} = \emptyset$

We say that two sets A and B are <u>disjoint</u> if $A \cap B = \emptyset$

We can take the intersection of any number of sets

Example: $A \cap B \cap C$

Graphically:



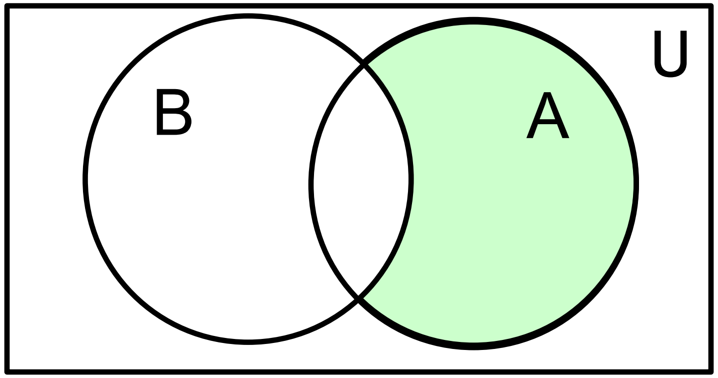
As with the union operation, we can express the intersection $S_1 \cap S_2 \cap \ldots \cap S_n$ as:

$$\bigcap_{i=1}^{n} S_i$$

Set differences

Definition: The difference of two sets A and B, denoted by A – B, contains every element that is in A, but not in B.

Graphically:



Mathematically:

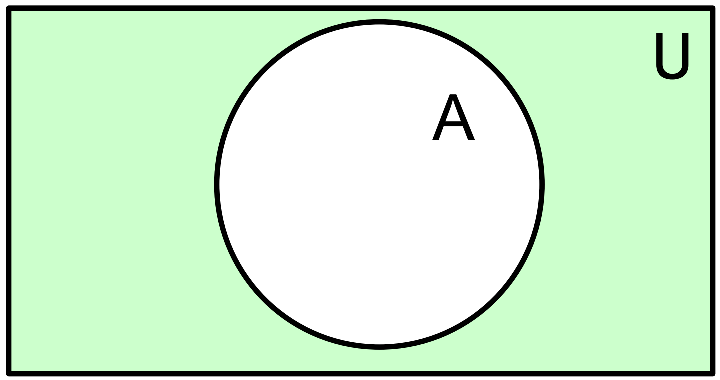
Example: {1, 2, 3, 4, 5} – {4, 5, 6, 7, 8} = {1, 2, 3}

Be careful: Some authors use the notation A \ B to denote the set difference A – B.

If we have specified a universe U, we can determine the complement of a set

Definition: The complement of a set A, denoted by $\overline{A}$, contains every element that is in U, but not in A.

Graphically:



Mathematically:

Examples: Assume that U = {1, 2, …, 10}

- $\overline{\{1, 2, 3, 4, 5\}}$ =
- $\overline{\{2, 4, 6, 8, 10\}}$ =

Cardinality is the measure of a set's size

Definition: Let S be a set. If there are exactly n elements in S, where n is a nonnegative integer, then S is a finite set whose cardinality is n. The cardinality of S is denoted by |S|.

Example: If S = {a, e, i, o, u}, then |S| =

Useful facts: If A and B are finite sets, then

- $|A \cup B| = |A| + |B| - |A \cap B|$
- $|A - B| = |A| - |A \cap B|$

Aside: We'll talk about the cardinality of infinite sets later in the course.

Power set

Definition: Given a set S, its power set is the set containing all subsets of S. We denote the power set of S as P(S).

Examples:

- P({1}) = {∅, {1}}
- P({1, 2, 3}) = {∅, {1}, {2}, {3}, {1,2}, {1,3}, {2, 3}, {1, 2, 3}}

Note:

- The set ∅ is in the power set of any set S
- The set S is in its own power set
- $|P(S)| = 2^{|S|}$
- Some authors use the notation 2^S to represent the power set of S

How do we represent ordered collections?

Definition: The ordered n-tuple $(a_1, a_2, \ldots, a_n)$ is the ordered collection that has a_1 as its first element, a_2 as its second element, …, and a_n as its n^{th} element.

Note: $(a_1, a_2, \ldots, a_n) = (b_1, b_2, \ldots, b_n)$ iff $a_i = b_i$ for $i = 1, \ldots, n$.

Special case: Ordered pairs of the form $(x \in \mathbf{Z}, y \in \mathbf{Z})$ are the basis of the Cartesian plane!

- $(a, b) = (c, d)$ iff $a = c$ and $b = d$
- $(a, b) = (b, a)$ iff $a = b$

How can we construct and describe ordered n-tuples?

We use the Cartesian product operator to construct ordered n-tuples

Definition: If A and B are sets, the Cartesian product of A and B, which is denoted $A \times B$, is the set of all ordered pairs (a, b) such that $a \in A$ and $b \in B$.

Mathematically:

Examples: Let $A = \{1, 2\}$ and $B = \{y, z\}$

- What is $A \times B$?
- $B \times A$?
- Are $A \times B$ and $B \times A$ equivalent?

Cartesian products can be made from more than two sets

Example: Let

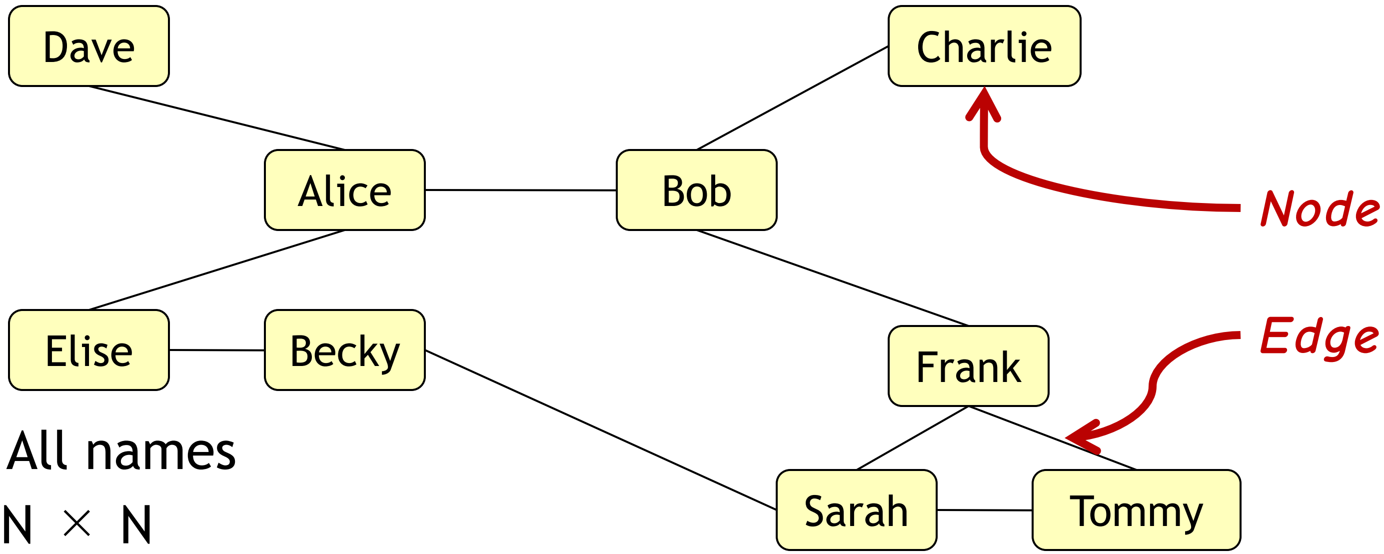
- $S = \{x \mid x \text{ is enrolled in CS 441}\}$
- $G = \{x \mid x \in \mathbf{R} \land 0 \leq x \leq 100\}$
- $Y = \{\text{freshman, sophomore, junior, senior}\}$

The set $S \times Y \times G$ consists of all possible (CS441 student, year, grade) combinations.

Note: My grades database is a subset of $S \times Y \times G$ that defines a relation between students in the class, their year at Pitt, and their grade!

We will study the properties of relations towards the end of this course.

Sets and Cartesian products can be used to represent trees and graphs



Let:

- N = All names
- F = N × N

A social network can be represented as a graph (V, E) in which the set V denotes the people in the network and the set E denotes the set of "friendship" links: (V, E) ∈ P(N) × P(F)

In the above network:

- V = {Alice, Bob, …, Tommy} ⊆ N
- E = {(Alice, Bob), (Alice, Dave), …, (Sarah, Tommy)} ⊆ N × N

Set notation allows us to make quantified statements more precise

We can use set notation to make the domain of a quantified statement explicit.

Example: $\forall x \in \mathbf{R}\ (x^2 \geq 0)$

- The square of any real number is a least zero

Example: $\forall n \in \mathbf{Z}\ \exists j,k \in \mathbf{Z}\ [(3n+2 = 2j+1) \rightarrow (n = 2k+1)]$

- If n is an integer and 3n + 2 is odd, then n is odd.

Note: This notation is far less ambiguous than simply stating the domains of propositional functions. In the remainder of the course, we will use this notation whenever possible.

Truth sets describe when a predicate is true

Definition: Given a predicate P and its corresponding domain D the truth set of P enumerates all elements in D that make the predicate P true.

Examples: What are the truth sets of the following predicates, given that their domain is the set **Z**?

- $P(x) \equiv |x| = 1$
- $Q(x) \equiv x^2 > 0$
- $R(x) \equiv x^5 = 1049$

Note:

- $\forall x\ P(x)$ is true iff the truth set of P is the entire domain D
- $\exists x\ P(x)$ is true iff the truth set of P is non-empty

How do computers represent and manipulate finite sets?

Observation: Representing sets as unordered collections of elements (e.g., arrays of Java Object data types) can be inefficient.

As a result, sets are usually represented using either hash maps or bitmaps.

You'll learn about these in 1501, so today we'll focus on bitmap representations.

This is probably best explained through an example…

Playing with the set $S=\{x \mid x \in \mathbf{N}, x<10\}$

To represent a set as a bitmap, we must first agree on an ordering for the set. In the case of S, let's use the natural ordering of the numbers.

Now, any subset of S can be represented using $|S|=10$ bits. For example:

- $\{1, 3, 5, 7, 9\}$ = 0101 0101 01
- $\{1, 1, 1, 4, 5\}$ = 0100 1100 00

What subsets of S do the following bitmaps represent?

- 0101 1010 11
- 1111 0000 10

Set operations can be carried out very efficiently as bitwise operations

Example: {1, 3, 7} ∪ {2, 3, 8}

$$
\begin{array}{r}
0101\ 0001\ 00 \\
\vee\ 0011\ 0000\ 10 \\
\hline
\end{array}
$$

Example: {1, 3, 7} ∩ {2, 3, 8}

$$
\begin{array}{r}
0101\ 0001\ 00 \\
\wedge\ 0011\ 0000\ 10 \\
\hline
\end{array}
$$

Note: These operations are much faster than searching through unordered lists!

Set operations can be carried out very efficiently as bitwise operations

Example: $\overline{\{1, 3, 7\}}$

$$\begin{array}{l} \neg 0101\ 0001\ 00 \\ \hline 1010\ 1110\ 11 \end{array} = \{0, 2, 4, 5, 6, 8, 9\}$$

Since the set difference A – B can be written as $A \cap \overline{(A \cap B)}$, we can calculate it as $A \wedge \neg(A \wedge B)$.

Although set difference is more complicated than the basic operations, it is still much faster to calculate set differences using a bitmap approach as opposed to an unordered search.

In-class exercises

Problem 4: Let A = {1, 2, 3, 4}, B = {3, 5, 7, 9}, and C = {7, 8, 9, 10}. Calculate the following:

- $A \cap B$
- $A \cup B \cup C$
- $B \cap C$
- $A \cap B \cap C$

Problem 5: Come up with a bitmap representation of the sets A = {a, c, d, f} and B = {a, b, c}. Use this to calculate the following:

- $A \cup B$
- $A \cap B$

Final thoughts

- Sets are one of the most basic data structures used in computer science

- Today, we looked at:
  - How to define sets
  - Basic set operations
  - How computers represent sets

- Next time:
  - Set identities (Section 2.2)
  - Functions (Section 2.3)