START

RECORDING

The Rule of Inclusion / Exclusion

CMSC 250

Inclusion / Exclusion Principle

- We will introduce the inclusion / exclusion principle through its two constituents:
  - Addition rule
  - Subtraction rule
  - *(Ok, to be fully honest, the multiplication rule is still relevant!)*

Picking Projects

- Murad is taking CMSC420 (Data Structures)
- He has to pick **three projects total** for the course.
- The CMSC 420 projects are divided into three categories.

Picking Projects

- Murad is taking CMSC420 (Data Structures)
- He has to pick **three projects total** for the course.
- The CMSC 420 projects are divided into three categories.
    1. Hashing (20 projects available),

Picking Projects

- Murad is taking CMSC420 (Data Structures)
- He has to pick **three projects total** for the course.
- The CMSC 420 projects are divided into three categories.
  1. Hashing (20 projects available),
  2. Multi-Dimensional Indexing(15 projects available)

Picking Projects

- Murad is taking CMSC420 (Data Structures)
- He has to pick **three projects total** for the course.
- The CMSC 420 projects are divided into three categories.
  1. Hashing (20 projects available),
  2. Multi-Dimensional Indexing(15 projects available)
  3. Data Compression (40 data compression projects available).

Picking Projects

- Murad is taking CMSC420 (Data Structures)
- He has to pick **three projects total** for the course.
- The CMSC 420 projects are divided into three categories.
    1. Hashing (20 projects available),
    2. Multi-Dimensional Indexing(15 projects available)
    3. Data Compression (40 data compression projects available).

In how many different ways can Murad pick a project?

Picking Projects

- Murad is taking CMSC420 (Data Structures)
- He has to pick **three projects total** for the course.
- The CMSC 420 projects are divided into three categories.
    1. Hashing (20 projects available),
    2. Multi-Dimensional Indexing(15 projects available)
    3. Data Compression (40 data compression projects available).

In how many different ways can Murad pick a project?

- By the multiplication rule: $20 \times 15 \times 40 = 12000$

Picking Projects

- Suppose now that Murad has to pick one project for CMSC420.
- Categories are the same:
  1. Hashing (20 projects available),
  2. Multi-Dimensional Indexing(15 projects available)
  3. Data Compression (40 data compression projects available).

In how many different ways can Murad pick a project now?

Picking Projects

- Suppose now that Murad has to pick one project for CMSC420.
- Categories are the same:
  1. Hashing (20 projects available),
  2. Multi-Dimensional Indexing(15 projects available)
  3. Data Compression (40 data compression projects available).

In how many different ways can Murad pick a project now?

- There are 20 + 15 + 40 = 75 projects available, so 75 different ways.

Picking Projects

- Suppose now that Murad has to pick one project for CMSC420.
- Categories are the same:
  1. Hashing (20 projects available),
  2. Multi-Dimensional Indexing(15 projects available)
  3. Data Compression (40 data compression projects available).

In how many different ways can Murad pick a project now?

- There are 20 + 15 + 40 = 75 projects available, so 75 different ways.
- Note that if a project was shared between two categories, we'd have an overcount! (74 instead of 75)

Picking Passwords

- Suppose that we want to register for some website, and we have to pick a password.

Picking Passwords

- Suppose that we want to register for some website, and we have to pick a password.
- The website's pretty old-tech, so it tells us that the password should be between 4 and 6 symbols long, with English lowercase or uppercase characters, digits, as well as any one of the "special" characters #, *, _, -, @, &, !

Picking Passwords

- Suppose that we want to register for some website, and we have to pick a password.
- The website's pretty old-tech, so it tells us that the password should be between 4 and 6 symbols long, with English lowercase or uppercase characters, digits, as well as any one of the "special" characters #, *, _, -, @, &, !
    - ***How many different passwords can the website store in its database?***

Picking Passwords

- Suppose that we want to register for some website, and we have to pick a password.
- The website's pretty old-tech, so it tells us that the password should be between 4 and 6 symbols long, with English lowercase or uppercase characters, digits, as well as any one of the "special" characters #, *, _, -, @, &, !
  - ***How many different passwords can the website store in its database?***
  - If we call the sets of different passwords N_4, N_5, N_6, we have:

$$|N_4| + |N_5| + |N_6|$$

- Letters, lowercase and uppercase
- Digits
- #, *, _, -, @, &, !

Calculating...

$|N_4|$ =

$P(69, 4)$

69^4

$\binom{69}{4}$

4^{69}

$|N_5|$ =

$|N_6|$ =

- Letters, lowercase and uppercase
- Digits
- #, *, _, -, @, &, !

Calculating...

$P(69, 4)$

69^4

$\binom{69}{4}$

4^{69}

$$|N_4| = 69^4$$

$$|N_5| =$$

$$|N_6| =$$

Calculating...

- Letters, lowercase and uppercase
- Digits
- #, *, _, -, @, &, !

$P(69,4)$ [✓ 69^4] $\binom{69}{4}$ 4^{69}

$|N_4| = 69^4$

$|N_5| = 69^5$

$|N_6| =$

- Letters, lowercase and uppercase
- Digits
- #, *, _, -, @, &, !

Calculating…

$|N_4|$ = 69^4

$P(69,4)$

69^4

$\binom{69}{4}$

4^{69}

$|N_5|$ = 69^5

$|N_6|$ = 69^6

- Letters, lowercase and uppercase
- Digits
- #, *, _, -, @, &, !

Calculating…

$P(69, 4)$

69^4

$\binom{69}{4}$

4^{69}

$|N_4| = 69^4$

$|N_5| = 69^5$

$|N_6| = 69^6$

That's about 109.5 billion different passwords!

- Letters, lowercase and uppercase
- Digits
- #, *, _, -, @, &, !

Calculating...

$|N_4| = 69^4$

$|N_5| = 69^5$

$|N_6| = 69^6$

$P(69, 4)$ | 69^4 | $\binom{69}{4}$ | 4^{69}

That's about 109.5 billion different passwords!

Finally, notice that N_4, N_5 and N_6 are pairwise disjoint sets (why?)

Picking Different Passwords

- Suppose now that the website tells us that our passwords should not have repeated characters.

Picking Different Passwords

- Suppose now that the website tells us that our passwords should not have repeated characters.
- Call our new sets M_4, M_5, M_6.

Picking Different Passwords

- Suppose now that the website tells us that our passwords should not have repeated characters.
- Call our new sets M_4, M_5, M_6.
- The total #passwords is still yielded as:

$$|M_4| + |M_5| + |M_6|$$

- Letters, lowercase and uppercase
- Digits
- #, *, _, -, @, &, !
- **NO REPEATED CHARS**

Calculating…

$|M_4|$ =

$P(69, 4)$

69^4

$\binom{69}{4}$

4^{69}

$|M_5|$ =

$|M_6|$ =

- Letters, lowercase and uppercase
- Digits
- #, *, _, -, @, &, !
- **NO REPEATED CHARS**

Calculating...

$|M_4| = P(69, 4)$

$P(69, 4)$

69^4

$\binom{69}{4}$

4^{69}

$|M_5| =$

$|M_6| =$

Calculating…

- Letters, lowercase and uppercase
- Digits
- #, *, _, -, @, &, !
- **NO REPEATED CHARS**

$|M_4| = P(69,4)$

$P(69,4)$

69^4

$\binom{69}{4}$

4^{69}

$|M_5| = P(69,5)$

$|M_6| =$

- Letters, lowercase and uppercase
- Digits
- #, *, _, -, @, &, !
- **NO REPEATED CHARS**

Calculating...

$|M_4| = P(69, 4)$

[✓ $P(69, 4)$] 69^4 $\binom{69}{4}$ 4^{69}

$|M_5| = P(69, 5)$

$|M_6| = P(69, 6)$

Calculating…

- Letters, lowercase and uppercase
- Digits
- #, *, _, -, @, &, !
- **NO REPEATED CHARS**

$|M_4| = P(69,4)$

$P(69,4)$ | 69^4 | $\binom{69}{4}$ | 4^{69}

$|M_5| = P(69,5)$

$|M_6| = P(69,6)$

That's about 87.5 billion different passwords!

- Letters, lowercase and uppercase
- Digits
- #, *, _, -, @, &, !
- NO REPEATED CHARS

Calculating…

$|M_4| = P(69,4)$

$P(69,4)$ 69^4 $\binom{69}{4}$ 4^{69}

$|M_5| = P(69,5)$

$|M_6| = P(69,6)$

That's about 87.5 billion different passwords!

Finally, notice that M_4, M_5 and M_6 are still disjoint sets.

The Addition Rule

- The previous example was an instance of the so-called **addition rule**.

The Addition Rule

- The previous example was an instance of the so-called **addition rule**.
- Formally, the rule is stated as follows:

Let $n \in \mathbb{N}^{>0}$. If $A_1, A_2, \ldots, A_n$ are finite, pairwise disjoint sets, then

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

The Addition Rule

- The previous example was an instance of the so-called **addition rule**.
- Formally, the rule is stated as follows:

Let $n \in \mathbb{N}^{>0}$. If $A_1, A_2, \dots, A_n$ are finite, pairwise disjoint sets, then

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

- In our examples,

$$|N_4 \cup N_5 \cup N_6| = \sum_{i=4}^{6} |N_i| \quad (= 69^4 + 69^5 + 69^6)$$

The Addition Rule

- The previous example was an instance of the so-called **addition rule**.
- Formally, the rule is stated as follows:

Let $n \in \mathbb{N}^{>0}$. If $A_1, A_2, \dots, A_n$ are finite, pairwise disjoint sets, then

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

- In our examples,

$$|N_4 \cup N_5 \cup N_6| = \sum_{i=4}^{6} |N_i| \quad (= 69^4 + 69^5 + 69^6)$$

$$|M_4 \cup M_5 \cup M_6| = \sum_{i=4}^{6} |M_i| \quad (= \mathrm{P}(69{,}4) \ + \mathrm{P}(69, 5) + \mathrm{P}(69{,}6))$$

Practice

- Once again, our passwords can use: English lowercase or uppercase characters, digits, as well as any one of the “special” characters #, *, _, -, @, &, !
  - 69 characters total.

Practice

- Once again, our passwords can use: English lowercase or uppercase characters, digits, as well as any one of the "special" characters #, *, _, -, @, &, !
  - 69 characters total.
- Alice likes passwords of length 6 that start with an 'A'.

Practice

- Once again, our passwords can use: English lowercase or uppercase characters, digits, as well as any one of the "special" characters #, *, _, -, @, &, !
    - 69 characters total.
- Alice likes passwords of length 6 that start with an 'A'.
- Bob likes passwords of length 6 that end with a 'B'.

Practice

- Once again, our passwords can use: English lowercase or uppercase characters, digits, as well as any one of the "special" characters #, *, _, -, @, &, !
  - 69 characters total.
- Alice likes passwords of length 6 that start with an 'A'.
- Bob likes passwords of length 6 that end with a 'B'.
- Both are security-conscious, so they never use the same character.

Practice

- Once again, our passwords can use: English lowercase or uppercase characters, digits, as well as any one of the “special” characters #, *, _, -, @, &, !
  - 69 characters total.
- Alice likes passwords of length 6 that start with an ‘A’.
- Bob likes passwords of length 6 that end with a ‘B’.
- Both are security-conscious, so they never use the same character.
- What is the total number of passwords that either Alice or Bob use?

Practice

- Call the sets of passwords that Alice uses P_A.

Practice

- Call the sets of passwords that Alice uses P_A.
  - What is $|P_A|$?

$P(69, 6)$

$P(69, 5)$

69^5

Something Else

Practice

- Call the sets of passwords that Alice uses P_A.
  - What is $|P_A|$?

$P(69, 6)$ $P(69, 5)$ 69^5 [✓ Something Else] $P(68, 5)$

Practice

- Call the sets of passwords that Alice uses P_A.
  - What is $|P_A|$?

$P(69, 6)$

$P(69, 5)$

69^5

[✓ Something Else]

$P(68, 5)$

  - Similarly, $|P_B| = P(68, 5)$

Practice

- Call the sets of passwords that Alice uses P_A.
  - What is $|P_A|$?

$P(69, 6)$ $P(69, 5)$ 69^5 [✓ Something Else] $P(68, 5)$

  - Similarly, $|P_B| = P(68, 5)$
- What am I looking for?

$|P_A \cap P_B|$ $|P_A \cup P_B|$ $|P_A - P_B|$ $|P_B - P_A|$

Practice

- Call the sets of passwords that Alice uses P_A.
  - What is $|P_A|$?

$P(69, 6)$

$P(69, 5)$

69^5

[✓ Something Else]

$P(68, 5)$

- Similarly, $|P_B| = P(68, 5)$
- What am I looking for?

$|P_A \cap P_B|$

[✓ $|P_A \cup P_B|$]

$|P_A - P_B|$

$|P_B - P_A|$

Remember: I'm looking for the #passwords that **either Alice OR Bob use.**

Practice

- **You** told us that we're looking for $|P_A \cup P_B|$
- By the addition rule, $|P_A \cup P_B| = |P_A| + |P_B| = 2 * P(68, 5)$

Practice

- **You** told us that we're looking for $|P_A \cup P_B|$
- By the addition rule, $|P_A \cup P_B| = |P_A| + |P_B| = 2 * P(68, 5)$

You've been punked!

- $A1234B$ was counted twice!

Practice

- **You** told us that we're looking for $|P_A \cup P_B|$
- By the addition rule, $|P_A \cup P_B| = |P_A| + |P_B| = 2 * P(68, 5)$

You've been punked!

- $A1234B$ was counted twice!
- Many passwords were counted twice
  - How many?

Practice

- #passwords **EITHER ALICE OR BOB** both like =

  #passwords Alice likes

  \+ #passwords Bob likes

  \- #passwords they both like

Practice

- #passwords **EITHER ALICE OR BOB** both like =

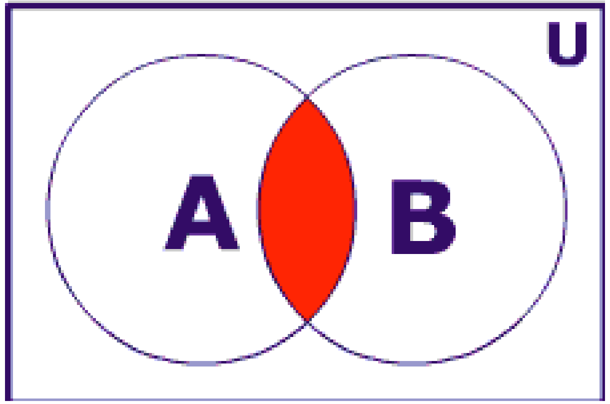
  #passwords Alice likes

  \+ #passwords Bob likes

  \- #passwords they both like

- Or, in terms of Set Theory:

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



Need $|P_A \cap P_B|$

- How many passwords do both Alice and Bob like?

Need $|P_A \cap P_B|$

- How many passwords do both Alice and Bob like?

$P(69, 4)$

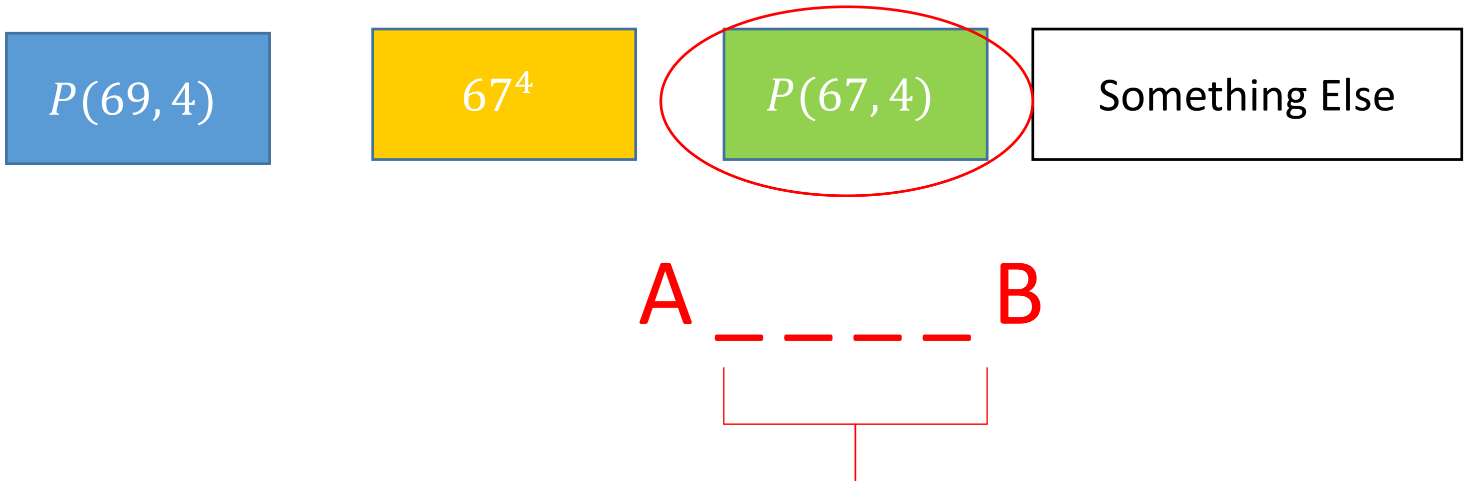
$P(68, 4)$

$P(67, 4)$

Something Else

Need $|P_A \cap P_B|$

- How many passwords do both Alice and Bob like?



$P(69,4)$ 67^4 $P(67,4)$ Something Else

A _ _ _ _ B

- 4 positions
- Cannot choose 'A' and 'B' because they've been used already!
  - So 67 characters available
- Order matters.

$|P_A \cup P_B|$

- From the rule we supplied earlier:

$$|P_A \cup P_B| = |P_A| + |P_B| - |P_A \cap P_B| = 2 * P(68,5) - P(67,4) =$$

$|P_A \cup P_B|$

- From the rule we supplied earlier:

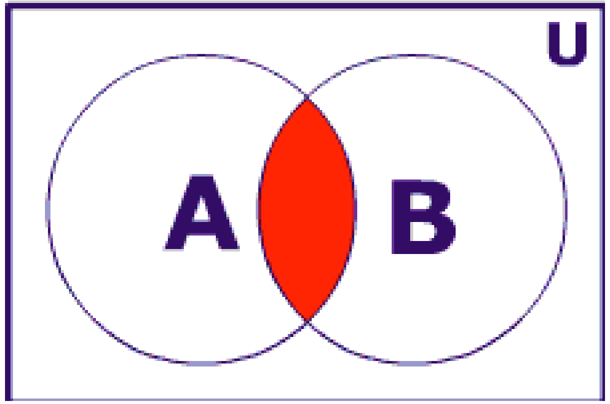
$$|P_A \cup P_B| = |P_A| + |P_B| - |P_A \cap P_B| = 2 * P(68,5) - P(67,4) =$$

NOPE, WE’RE BUSY PEOPLE

General Rule

- For any finite sets A, B:

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

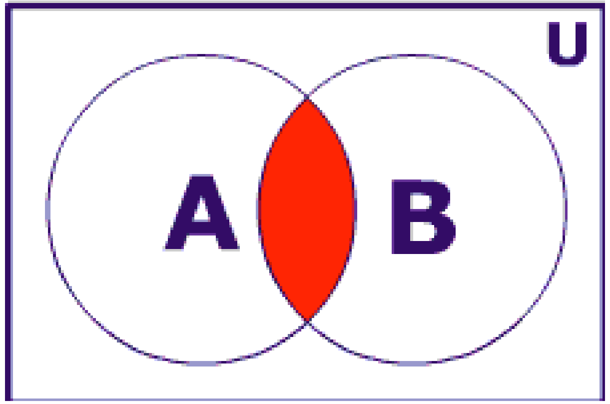


General Rule

- For any finite sets A, B:

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

- This is the **inclusion-exclusion principle.**



Applications

A Number-Theoretic Problem

- How many numbers between 1 and 1000 are divisible by either 2 or 3?

A Number-Theoretic Problem

- How many numbers between 1 and 1000 are divisible by either 2 or 3?
- $A_2 = \{x \in \mathbb{N} | (1 \leq x \leq 1000) \wedge (x \equiv 0\ (mod\ 2))\}$
- $A_3 = \{x \in \mathbb{N} | (1 \leq x \leq 1000) \wedge (x \equiv 0\ (mod\ 3))\}$

A Number-Theoretic Problem

- How many numbers between 1 and 1000 are divisible by either 2 or 3?
- $A_2 = \{x \in \mathbb{N} | (1 \leq x \leq 1000) \wedge (x \equiv 0 \ (mod\ 2))\}$
- $A_3 = \{x \in \mathbb{N} | (1 \leq x \leq 1000) \wedge (x \equiv 0 \ (mod\ 3))\}$
- Generally, $A_i = \{x \in \mathbb{N} | (1 \leq x \leq 1000) \wedge (x \equiv 0 \ (mod\ i))\}$
- $|A_2| = \lfloor {}^{1000}/_{2} \rfloor = 500$
- $|A_3| = \lfloor {}^{1000}/_{3} \rfloor = 333$
- $|A_i| = \left\lfloor {}^{1000}/_{i} \right\rfloor$

A Number–Theoretic Problem

- $|A_2 \cup A_3| = |A_2| + |A_3| - |A_2 \cap A_3| = 833\ - |A_2 \cap A_3|$

A Number–Theoretic Problem

- $|A_2 \cup A_3| = |A_2| + |A_3| - |A_2 \cap A_3| = 833\ - |A_2 \cap A_3|$
  - **What is the set $\boldsymbol{A_2 \cap A_3}$?**

A Number–Theoretic Problem

- $|A_2 \cup A_3| = |A_2| + |A_3| - |A_2 \cap A_3| = 833 \ - |A_2 \cap A_3|$
  - **What is the set $A_2 \cap A_3$?**
  - It's just A_6.
- $|A_6| = \lfloor {}^{1000}/_{6} \rfloor = 166$
- So $|A_2 \cup A_3| = 833 \ - 166 = 667$

Counting Bit-Strings (for you to do **NOW**)

- Let's consider the problem of counting how many bit strings of length 8 exist such that they either start with 1 or end with 00.

Counting Bit-Strings (for you to do **NOW**)

- Let’s consider the problem of counting how many bit strings of length 8 exist such that they either start with 1 or end with 00.
  - How many strings start with a 1?

Counting Bit-Strings (for you to do **NOW**)

- Let’s consider the problem of counting how many bit strings of length 8 exist such that they either start with 1 or end with 00.
  - How many strings start with a 1? 2^7

Counting Bit-Strings (for you to do **NOW**)

- Let’s consider the problem of counting how many bit strings of length 8 exist such that they either start with 1 or end with 00.
  - How many strings start with a 1? 2^7
  - How many strings end with 00? 2^6

Counting Bit-Strings

- Let’s consider the problem of counting how many bit strings of length 8 exist such that they either start with 1 or end with 00.
  - How many strings start with a 1? 2^7
  - How many strings end with 00? 2^6
  - Unfortunately, we **overcount** strings such as 11010100…

Counting Bit-Strings

- Let’s consider the problem of counting how many bit strings of length 8 exist such that they either start with 1 or end with 00.
  - How many strings start with a 1? 2^7
  - How many strings end with 00? 2^6
  - Unfortunately, we **overcount** strings such as 11010100…
    - **But we can count exactly how many those strings are!**

Counting Bit-Strings

- Let’s consider the problem of counting how many bit strings of length 8 exist such that they either start with 1 or end with 00.
  - How many strings start with a 1? 2^7
  - How many strings end with 00? 2^6
  - Unfortunately, we **overcount** strings such as 11010100…
    - **But we can count exactly how many those strings are!**
    - **They are 2^5**

Counting Bit-Strings

- Let’s consider the problem of counting how many bit strings of length 8 exist such that they either start with 1 or end with 00.
  - How many strings start with a 1? 2^7
  - How many strings end with 00? 2^6
  - Unfortunately, we **overcount** strings such as 11010100…
    - **But we can count exactly how many those strings are!**
    - **They are 2^5**
    - **Therefore, final answer = $2^7 + 2^6 - 2^5 = 128 + 64 - 32 = 160$ ☺**

Practice (For You)

- Bloomberg Corp. receives 350 applications.
  - 220 are from CS majors
  - 147 are Business majors
  - 51 majored in both.
- How many majored in neither?

Practice (For You)

- Bloomberg Corp. receives 350 applications.
  - 220 are from CS majors
  - 147 are Business majors
  - 51 majored in both.
- How many majored in neither?
- Let’s define some sets…

Practice (For You)

- Bloomberg Corp. receives 350 applications.
  - 220 are from CS majors
  - 147 are Business majors
  - 51 majored in both.
- How many majored in neither?
- Let's define some sets…
  - CS = set of Computer Science majors, |CS| = 220
  - B = set of Business majors, |B| = 147.
  - Then, $CS \cup B$ is the set of Comp Sci **or** Business majors.

Practice (For You)

- Bloomberg Corp. receives 350 applications.
  - 220 are from CS majors
  - 147 are Business majors
  - 51 majored in both.
- How many majored in neither?
- Let's define some sets…
  - CS = set of Computer Science majors, |CS| = 220
  - B = set of Business majors, |B| = 147.
  - Then, $CS \cup B$ is the set of Comp Sci **or** Business majors.
- We have that $|CS \cup B| = |CS| + |B| - |CS \cap B| = 220 + 147\ - 51 = 316$

Practice (For You)

- Bloomberg Corp. receives 350 applications.
  - 220 are from CS majors
  - 147 are Business majors
  - 51 majored in both.
- How many majored in neither?
- Let's define some sets…
  - CS = set of Computer Science majors, |CS| = 220
  - B = set of Business majors, |B| = 147.
  - Then, $CS \cup B$ is the set of Comp Sci **or** Business majors.
- We have that $|CS \cup B| = |CS| + |B| - |CS \cap B| = 220 + 147\ - 51 = 316$
- So a total of 350 – 316 = 34 applicants were neither CS nor Business majors

A More Complex Problem

- Some Discrete Mathematics students were polled about their past Computer Science & Mathematics course experience.

A More Complex Problem

- Some Discrete Mathematics students were polled about their past Computer Science & Mathematics course experience.
  - 30 had taken precalculus
  - 18 had taken calculus
  - 26 had taken Java
  - 9 had taken **both** precalculus and calculus
  - 16 had taken **both** precalculus and Java
  - 8 had taken **both** calculus and Java
  - 5 had taken all three courses

A More Complex Problem

- Some Discrete Mathematics students were polled about their past Computer Science & Mathematics course experience.
  - 30 had taken precalculus
  - 18 had taken calculus
  - 26 had taken Java
  - 9 had taken **both** precalculus and calculus
  - 16 had taken **both** precalculus and Java
  - 8 had taken **both** calculus and Java
  - 5 had taken all three courses
- How many students were polled?

A More Complex Problem

- P = precalc, J = Java, C = calc
- Is $|P \cup J \cup C| = |P| + |J| + |C|$?

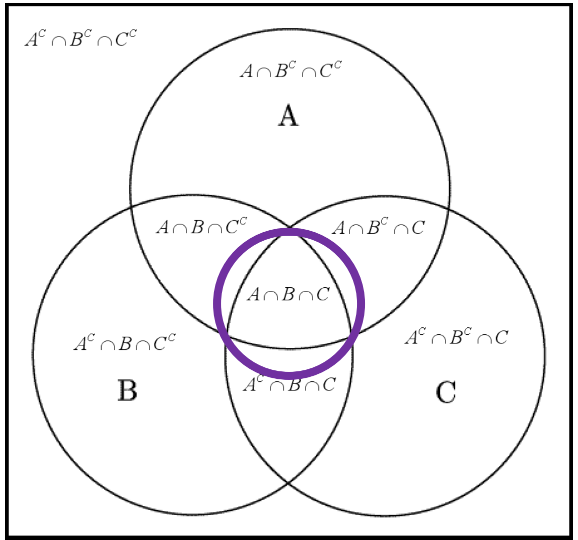
A More Complex Problem

- P = precalc, J = Java, C = calc
- Is $|P \cup J \cup C| = |P| + |J| + |C|$? NO. Overcounting strikes again.
  - We count students in $(P \cap J), (P \cap C), (J \cap C)$ **twice.**
- Is $|P \cup J \cup C| = |P| + |J| + |C| - (|P \cap J| + |P \cap C| + |J \cap C|)$?

A More Complex Problem

- P = precalc, J = Java, C = calc
- Is $|P \cup J \cup C| = |P| + |J| + |C|$? NO. Overcounting strikes again.
  - We count students in $(P \cap J), (P \cap C), (J \cap C)$ **twice.**
- Is $|P \cup J \cup C| = |P| + |J| + |C| - (|P \cap J| + |P \cap C| + |J \cap C|)$?

NO. We are losing the students in $(\boldsymbol{P} \cap \boldsymbol{C} \cap \boldsymbol{J})$!



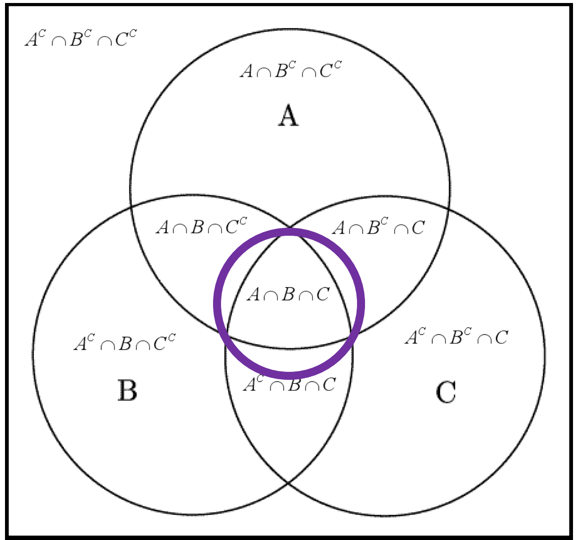
A More Complex Problem

- P = precalc, J = Java, C = calc
- Is $|P \cup J \cup C| = |P| + |J| + |C|$? NO. Overcounting strikes again.
  - We count students in $(P \cap J), (P \cap C), (J \cap C)$ **twice.**
- Is $|P \cup J \cup C| = |P| + |J| + |C| - (|P \cap J| + |P \cap C| + |J \cap C|)$?

NO. We are losing the students in $(\boldsymbol{P \cap C \cap J})$!

So we need to add them back:

$$|P \cup J \cup C| =$$
$$|P| + |J| + |C| - (|P \cap J| + |P \cap C| + |J \cap C|) + (\boldsymbol{P \cap C \cap J})$$



A More Complex Problem

Problem givens	**Translation into sets**
30 had taken precalculus	$\|P\| = 30$
18 had taken calculus	$\|C\| = 18$
26 had taken Java	$\|J\| = 26$
9 had taken **both** precalculus and calculus	$\|P \cap C\| = 9$
16 had taken **both** precalculus and Java	$\|P \cap J\| = 16$
8 had taken **both** calculus and Java	$\|J \cap C\| = 8$
5 had taken all three courses	$\|P \cap C \cap J\| = 5$

A More Complex Problem

Problem givens	**Translation into sets**
30 had taken precalculus	$\|P\| = 30$
18 had taken calculus	$\|C\| = 18$
26 had taken Java	$\|J\| = 26$
9 had taken **both** precalculus and calculus	$\|P \cap C\| = 9$
16 had taken **both** precalculus and Java	$\|P \cap J\| = 16$
8 had taken **both** calculus and Java	$\|J \cap C\| = 8$
5 had taken all three courses	$\|P \cap C \cap J\| = 5$

- We can then answer:

$$\begin{aligned} |\boldsymbol{P} \cup \boldsymbol{J} \cup \boldsymbol{C}| &= |P| + |J| + |C| - (|P \cap J| + |P \cap C| + |J \cap C|) + (\boldsymbol{P} \cap \boldsymbol{C} \cap \boldsymbol{J}) \\ &= 30 + 26 + 18 - (16 + 9 + 8) + 5 = \mathbf{46} \end{aligned}$$

A General Theorem

- For three finite sets A, B, C, we have:

$$|A \cup B \cup C| =$$

$$|A| + |B| + |C| - (|A \cap B| + |B \cap C| + |C \cap A|) + |A \cap B \cap C|$$

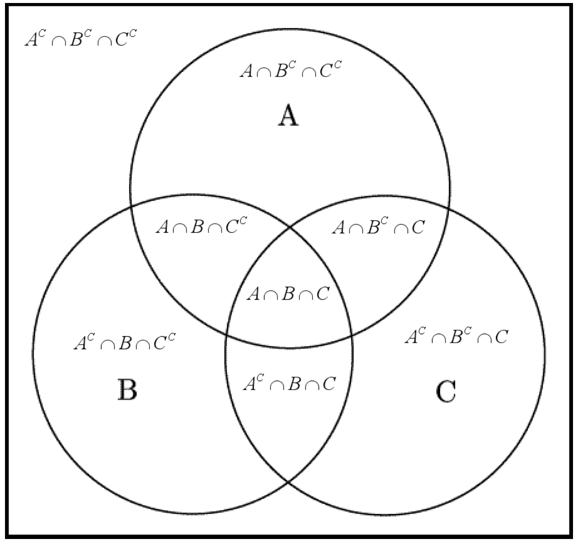
A General Theorem

- For three finite sets A, B, C, we have:

$$|A \cup B \cup C| =$$

$$|A| + |B| + |C| - (|A \cap B| + |B \cap C| + |C \cap A|) + |A \cap B \cap C|$$

- This is the inclusion-exclusion principle for 3 sets.



Divisibility Problem Again (For You, Now)

- How many numbers between 1 and 1000 are divisible by 2,3 or 5?

Divisibility Problem Again (For You, Now)

- How many numbers between 1 and 1000 are divisible by 2,3 or 5?
- Recall: $A_i = \{x \in \mathbb{N} \mid (1 \leq x \leq 1000) \wedge (x \equiv 0\ (mod\ i)\}$

Divisibility Problem Again **(For You, Now)**

- How many numbers between 1 and 1000 are divisible by 2,3 or 5?
- Recall: $A_i = \{x \in \mathbb{N} \mid (1 \leq x \leq 1000) \land (x \equiv 0 \ (mod\ i)\}$
- Therefore:

$$|A_2 \cup A_3 \cup A_5| =$$

$$= |A_2| + |A_3| + |A_5| - (|A_2 \cap A_3| + |A_2 \cap A_5| + |A_3 \cap A_5|) + |A_2 \cap A_3 \cap A_5|$$

$$= |A_2| + |A_3| + |A_5| - (|A_6| + |A_{10}| + |A_{15}|) + |A_{30}|$$

$$= \left\lfloor {}^{1000}/_{2} \right\rfloor + \left\lfloor {}^{1000}/_{3} \right\rfloor + \left\lfloor {}^{1000}/_{5} \right\rfloor - \left(\left\lfloor {}^{1000}/_{6} \right\rfloor + \left\lfloor {}^{1000}/_{10} \right\rfloor + \left\lfloor {}^{1000}/_{15} \right\rfloor \right) + \left\lfloor {}^{1000}/_{30} \right\rfloor$$

Divisibility Problem Again **(For You, Now)**

- How many numbers between 1 and 1000 are divisible by 2,3 or 5?
- Recall: $A_i = \{x \in \mathbb{N} \mid (1 \leq x \leq 1000) \wedge (x \equiv 0\ (mod\ i)\}$
- Therefore:

$$|A_2 \cup A_3 \cup A_5| =$$

$$= |A_2| + |A_3| + |A_5| - (|A_2 \cap A_3| + |A_2 \cap A_5| + |A_3 \cap A_5|) + |A_2 \cap A_3 \cap A_5|$$

$$= |A_2| + |A_3| + |A_5| - (|A_6| + |A_{10}| + |A_{15}|) + |A_{30}|$$

$$= \left\lfloor {}^{1000}/_{2} \right\rfloor + \left\lfloor {}^{1000}/_{3} \right\rfloor + \left\lfloor {}^{1000}/_{5} \right\rfloor - \left(\left\lfloor {}^{1000}/_{6} \right\rfloor + \left\lfloor {}^{1000}/_{10} \right\rfloor + \left\lfloor {}^{1000}/_{15} \right\rfloor \right) + \left\lfloor {}^{1000}/_{30} \right\rfloor$$

$$= 500 + 333 + 200 - 166\ - 100\ - 66 + 33 = 734$$

Question For You

- Previously, we found out that the #integers between 1 and 1000 div by 2 or 3 is 667.

Question For You

- Previously, we found out that the #integers between 1 and 1000 div by 2 or 3 is 667.
- We now found out that the #integers between 1 and 1000 div by 2, 3 or 5 is 735 > 667.

Question For You

- Previously, we found out that the #integers between 1 and 1000 div by 2 or 3 is 667.
- We now found out that the #integers between 1 and 1000 div by 2, 3 or 5 is 734 > 667.
- If we do the same thing for the #integers between 1 and 1000 div by 2, 3, 5 or 7, we will end up with a number between 735 and 1000.

Question For You

- Previously, we found out that the #integers between 1 and 1000 div by 2 or 3 is 667.
- We now found out that the #integers between 1 and 1000 div by 2, 3 or 5 is 734 > 667.
- If we do the same thing for the #integers between 1 and 1000 div by 2, 3, 5 or 7, we will end up with a number between 735 and 1000.
- Is there a prime p between 1 and 1000 for which the #integers between 1 and 1000 div by 2, 3, 5, 7, …, p is 1000?

Question For You

- Previously, we found out that the #integers between 1 and 1000 div by 2 or 3 is 667.
- We now found out that the #integers between 1 and 1000 div by 2, 3 or 5 is 734 > 667.
- If we do the same thing for the #integers between 1 and 1000 div by 2, 3, 5 or 7, we will end up with a number between 735 and 1000.
- Is there a prime p between 1 and 1000 for which the #integers between 1 and 1000 div by 2, 3, 5, 7, …, p is 1000?
  - Sure, the last prime before 1000! (non-constructive proof)

Question For You

- Previously, we found out that the #integers between 1 and 1000 div by 2 or 3 is 667.
- We now found out that the #integers between 1 and 1000 div by 2, 3 or 5 is 734 > 667.
- If we do the same thing for the #integers between 1 and 1000 div by 2, 3, 5 or 7, we will end up with a number between 735 and 1000.
- Is there a prime p between 1 and 1000 for which the #integers between 1 and 1000 div by 2, 3, 5, 7, …, p is 1000?
    - Sure, the last prime before 1000! (non-constructive proof)
    - If you wanted to do a constructive proof, what would you need to do?

Here’s One For You (Now)

- Inclusion-Exclusion rule for 4 (four) sets A_1, A_2, A_3, A_4

Here's One For You (Now)

- Inclusion-Exclusion rule for 4 (four) sets A_1, A_2, A_3, A_4

$|A_1 \cup A_2 \cup A_3 \cup A_4| =$

$$|A_1| + |A_2| + |A_3| + |A_4|$$

$$-(|A_1 \cap A_2| + |A_1 \cap A_3| + |A_1 \cap A_4| + |A_2 \cap A_3| + |A_2 \cap A_4| + |A_3 \cap A_4|)$$

$$+(|A_1 \cap A_2 \cap A_3| + |A_1 \cap A_2 \cap A_4| + |A_2 \cap A_3 \cap A_4| + |A_1 \cap A_3 \cap A_4|)$$

$$-|\mathrm{A}_1 \cap A_2 \cap A_3 \cap A_4|$$

STOP

RECORDING