Discrete Mathematics
Computational Thinking

1. Formulate the **Question** in English.

2. Determine the **Problem** you need to Solve.

3. List all the pertinent **Parameters**.

4. Build a **Mathematical Model** -- Formulas, Equations, . . .

5. Construct an **Algorithm** to Solve the Problem.

6. Develop **Data Structures** and **Code** to Implement Algorithm.
Computational Thinking

1. Formulate the **Question** in English.

2. Determine the **Problem** you need to Solve.

3. List all the pertinent **Parameters**.

4. Build a **Mathematical Model** -- Formulas, Equations, . . .

5. Construct an **Algorithm** to Solve the Problem.

6. Develop **Data Structures** and **Code** to Implement Algorithm.

**WHAT’S MISSING?**
WHAT’S MISSING?

4. *Build a Mathematical Model*
   - Standard Mathematical Methods?

5. *Construct an Algorithm*
   - Standard Algorithms?
   - Proofs of Correctness?

6. *Develop Data Structures*
   - Standard Data Structures?
Mathematical Tools of the Trade

Mathematical Methods
• Combinatorics and Probability
• Functions, Sets, Relations, . . .

Proof Techniques
• Induction and Logic

Algorithms
• Shortest Path Algorithms
• Searching and Sorting Algorithms

Theorems
• Binomial Theorem

Data Structures
• Graphs and Trees
# Discrete Math vs. Continuous Math

<table>
<thead>
<tr>
<th>Discrete</th>
<th>Continuous</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathbb{N} ) or ( \mathbb{Z} )</td>
<td>( \mathbb{R} ) or ( \mathbb{C} )</td>
</tr>
<tr>
<td>Counting</td>
<td>Limiting</td>
</tr>
<tr>
<td>Sequences  (Discrete Functions)</td>
<td>Continuous Functions</td>
</tr>
<tr>
<td>Sums</td>
<td>Integrals</td>
</tr>
<tr>
<td>Differences</td>
<td>Derivatives</td>
</tr>
</tbody>
</table>
## Discrete Math vs. Continuous Math (continued)

<table>
<thead>
<tr>
<th>Discrete</th>
<th>Continuous</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number Theory*</td>
<td>Calculus</td>
</tr>
<tr>
<td>Digital</td>
<td>Analog</td>
</tr>
</tbody>
</table>

* Other Topics -- logic, proofs, recursion, induction, combinatorics, probability, sets, functions, relations, regular languages, finite automata,...
Why Discrete Mathematics in Computer Science

- Digital Computers
- Discrete Data Structure
  -- Lists, Trees, Graphs, Sets, ...
- Discrete Programs
- Mathematical Tools of the Trade
  -- Not just Programming Languages
  -- Data Structures
  -- Mathematical Methods -- logic, recursion, relations, combinatorics, probability
  -- Algorithms and their Correctness
Motivation for Discrete Mathematics

Many, Many Applications

Fundamental Data Structures

Neat Algorithms

Engaging Theory

Novel Mathematics