

# DATA STRUCTURES & ALGORITHMS

---

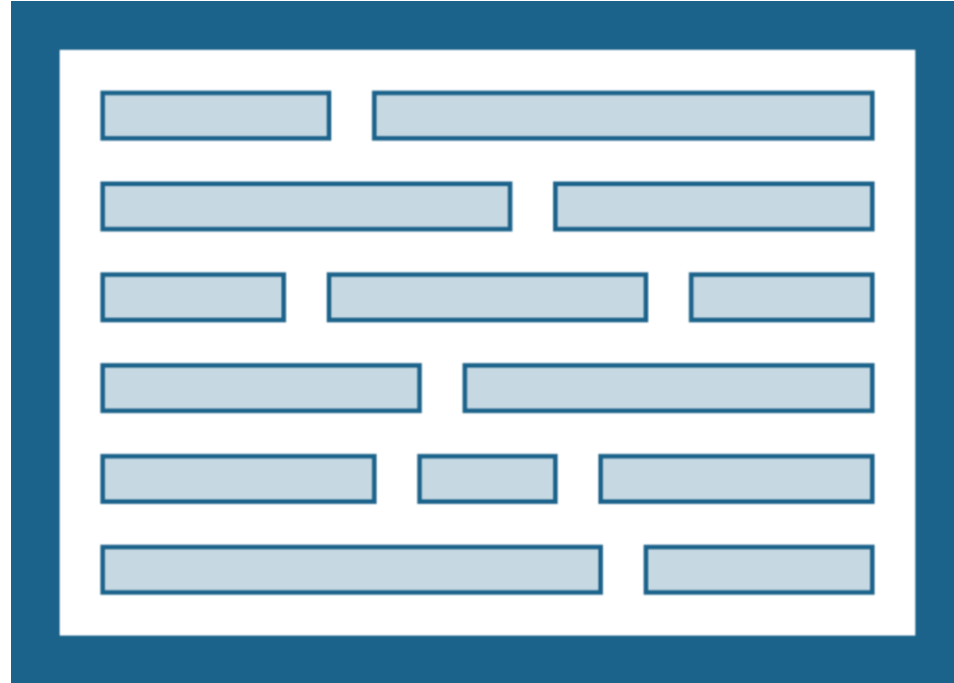
Why Study Algorithms

Instructor: Engr. Laraib Siddiqui

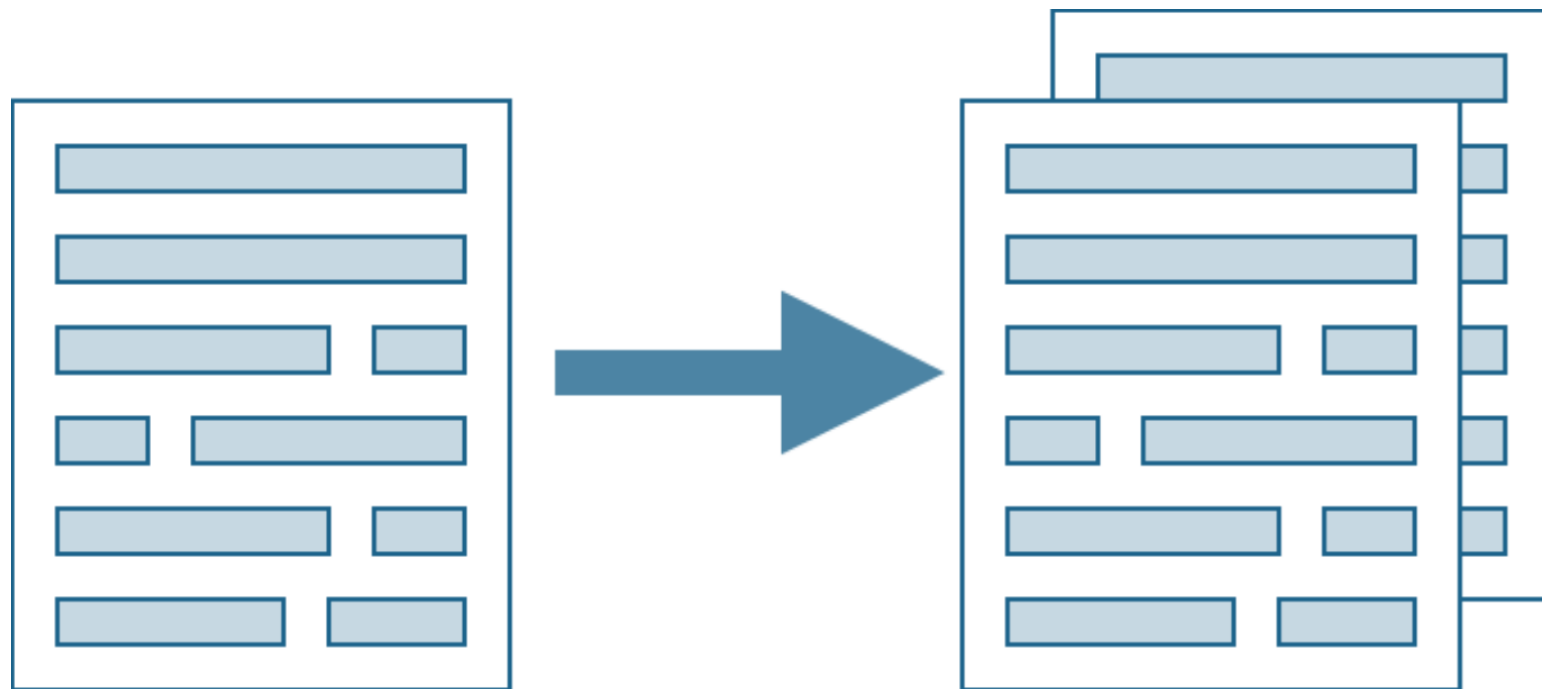
# Straightforward Programming Problems

- Has straightforward implementation.
- Natural solution is already efficient.

# Display given text



# Copy a File



# Search for a Given Word

**Simulation** hypothesis



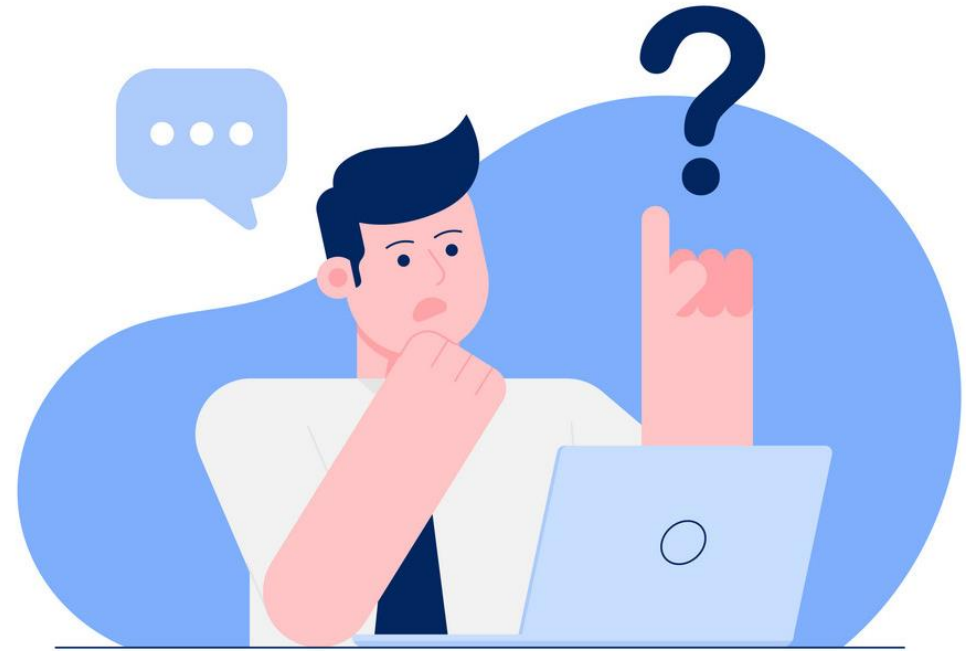
The **simulation** hypothesis or **simulation theory** proposes that all of reality, including the Earth and the universe, is in fact an artificial **simulation**, most likely a **computer**

# Simple Programming Problems

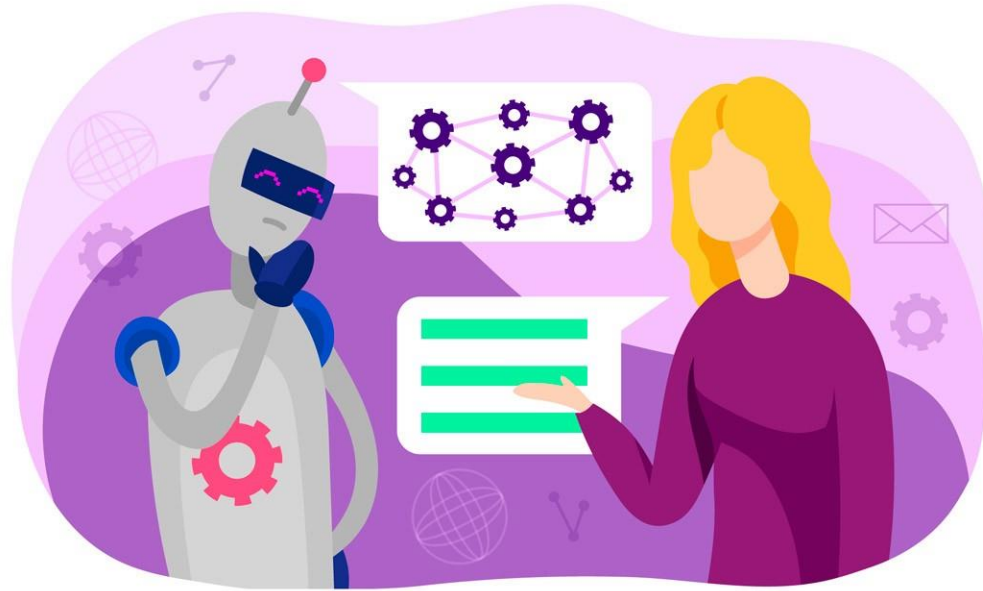
- Has linear scan.
- Cannot do much better.
- The obvious program works.

# Algorithms Problems

- Not so clear what to do.
- Not clear how to do.
- Simple ideas too slow.
- Room for optimization.

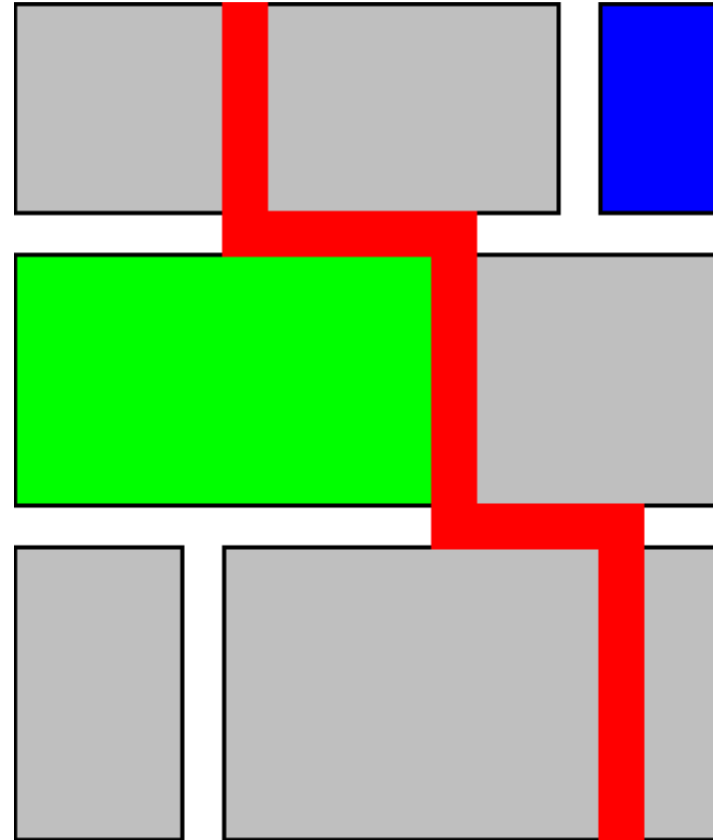


# Understand Natural Language

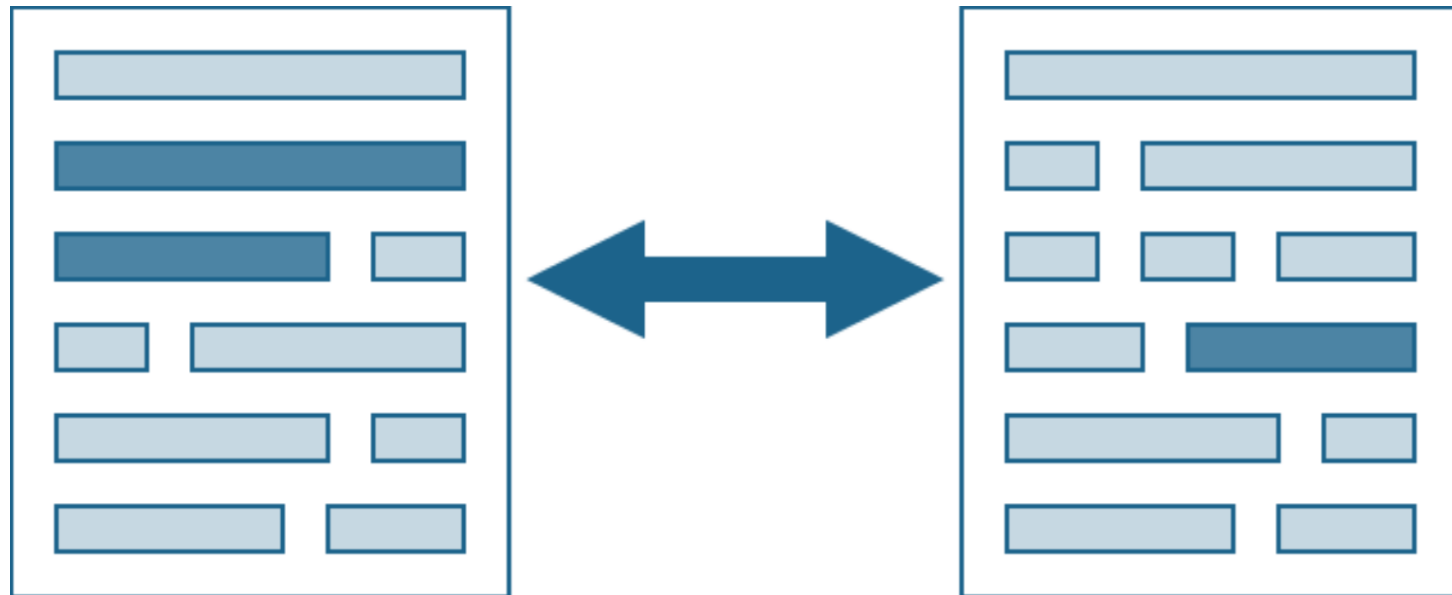




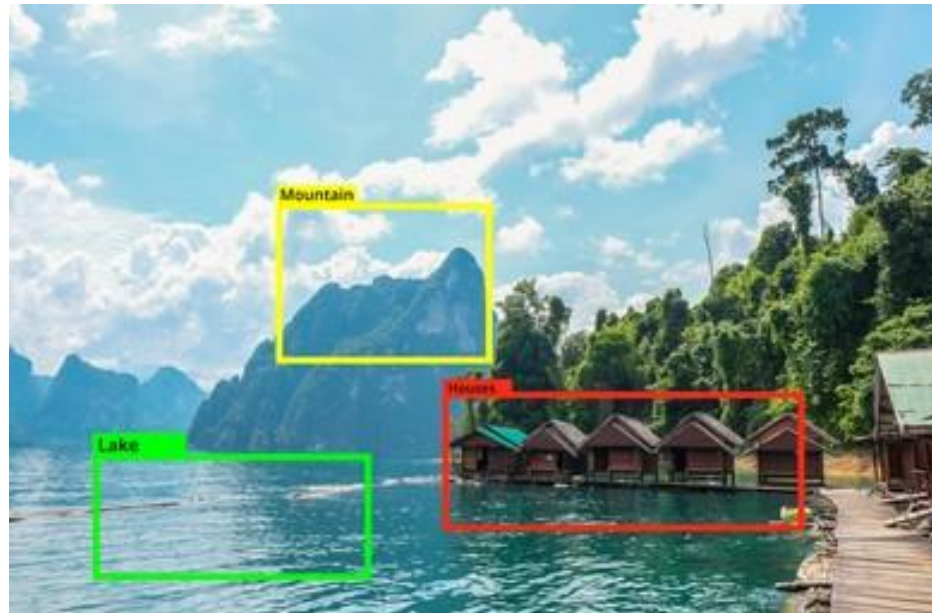
# Find the Shortest Path Between Locations



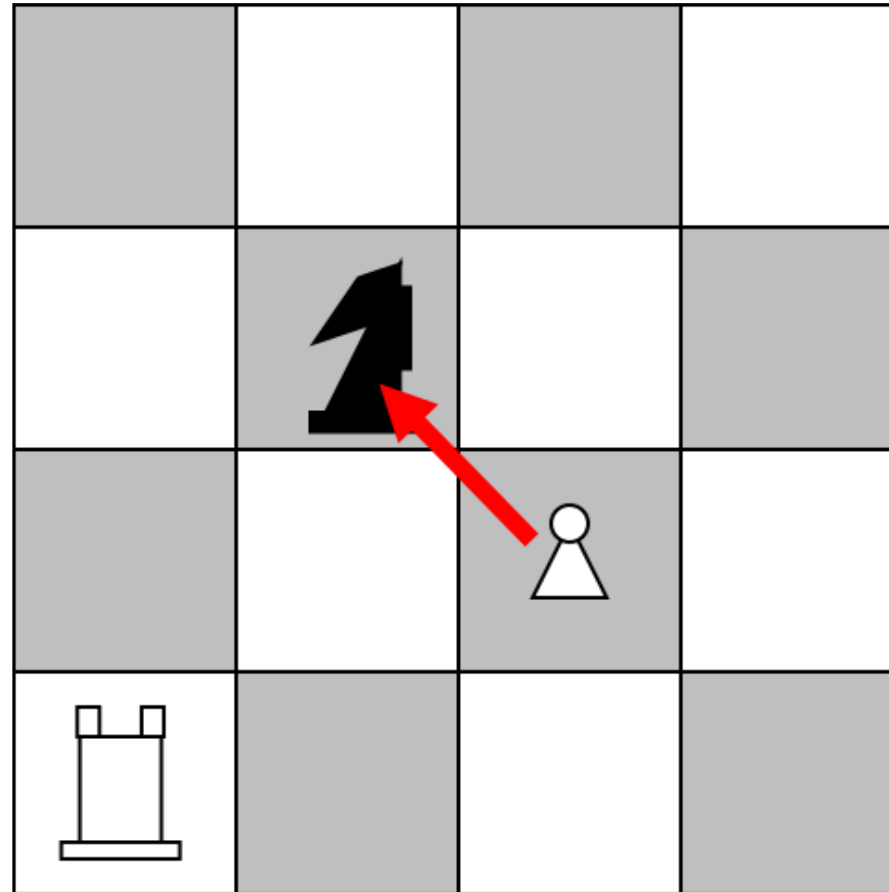
# Measure Similarity of Documents



# Identify Objects



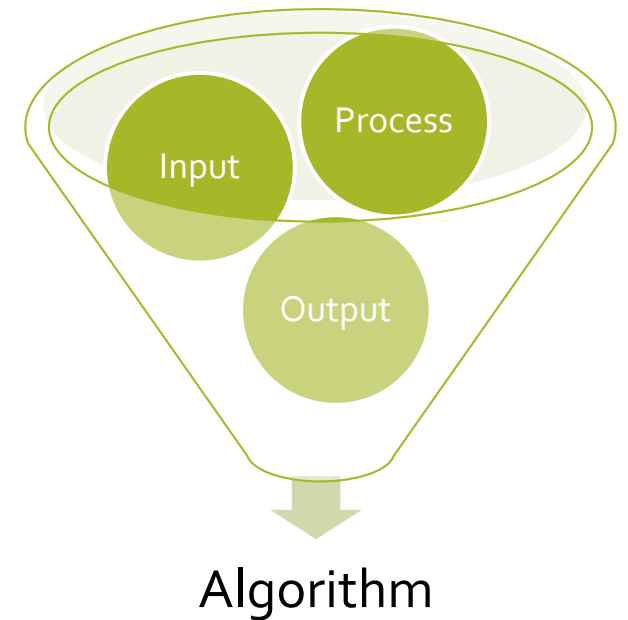
# Play Games Well



# Algorithm

- Set of instructions to accomplish a particular task.
- Method/process to solve a problem.

*Program is an instance of an algorithm, written in some specific programming language.*



# Why we need algorithms?

- Internet
  - Web search, packet routing, distributed file sharing, ...
- Computers.
  - Circuit layout, file system, compilers, ...
- Computer graphics.
  - Movies, video games, virtual reality, ...
- Security.
  - Cell phones, e-commerce, voting machines, ...
- Multimedia.
  - MP3, JPG, DivX, HDTV, face recognition, ...
- Social networks.
  - Recommendations, news feeds, advertisements, ...

# Basic Algorithm



- Input = a, b, c
- Output = max
- Process
  - Let max = a
  - If  $b > \text{max}$  then  
max = b
  - If  $c > \text{max}$  then  
max = c
- Display max

**NOTE:** Order is very important!

# Steps for Algorithm development

- Devising
  - ✓ Method to solve a problem
- Validating
  - ✓ Proof the correctness of algorithm
- Expressing
  - ✓ Implement in desired programming language



# Efficiency of an Algorithm

Measured by the **amount or resources** it uses (time/space)

Algorithm should:

- Independent of the programming language
- Should be applicable to all input sizes
- Understandable to programmer