Intro: Why Study Algorithms?

Daniel Kane

Department of Computer Science and Engineering
University of California, San Diego

Data Structures and Algorithms
Algorithmic Toolbox

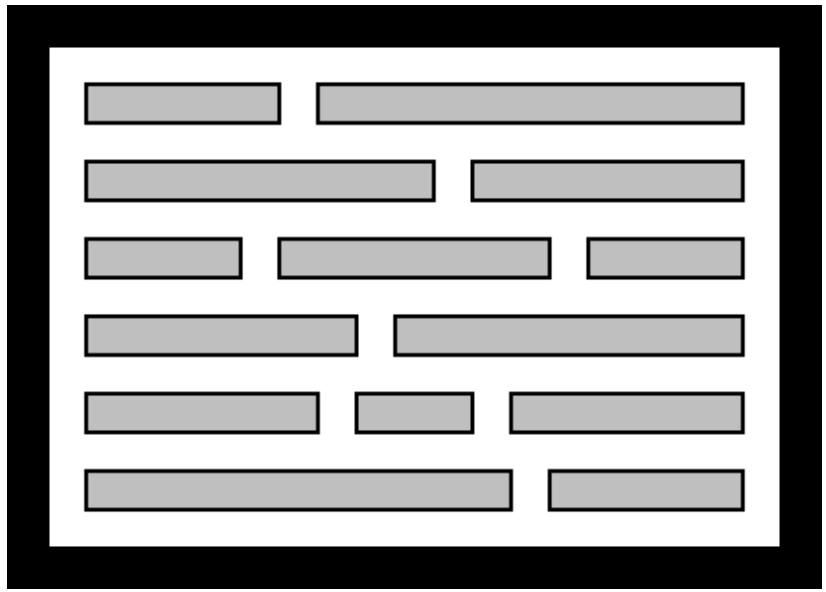
Learning Objectives

- Understand the type of problem that will be covered in this class.
- Recognize some problems for which sophisticated algorithms might not be necessary.
- Describe some artificial intelligence problems which go beyond the scope of this course.

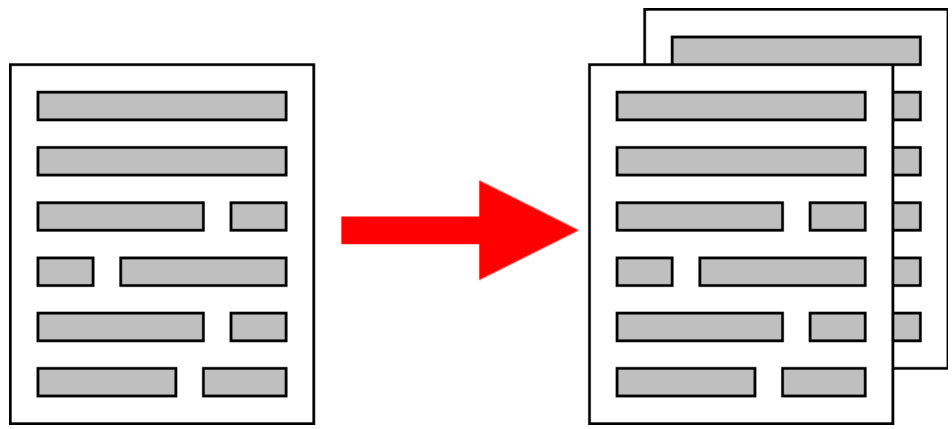
Straightforward Programming Problems

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

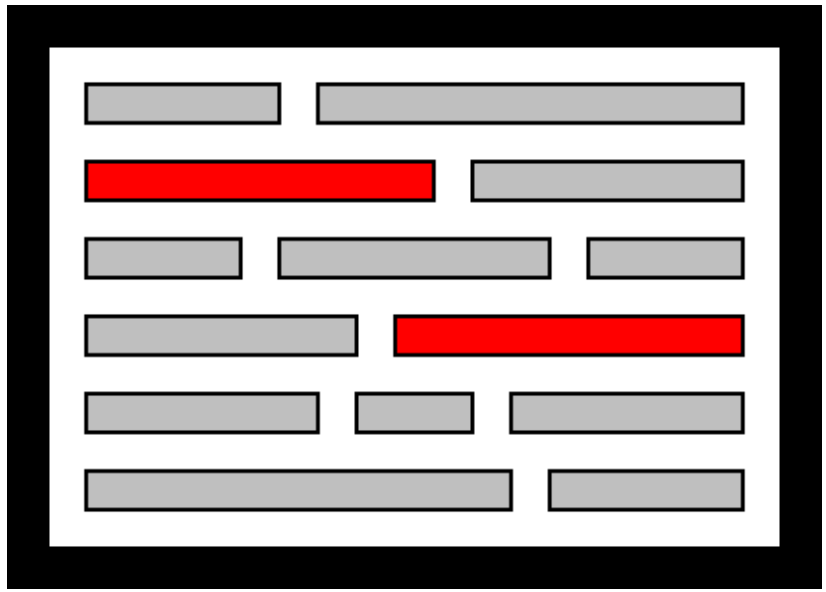
Display given text



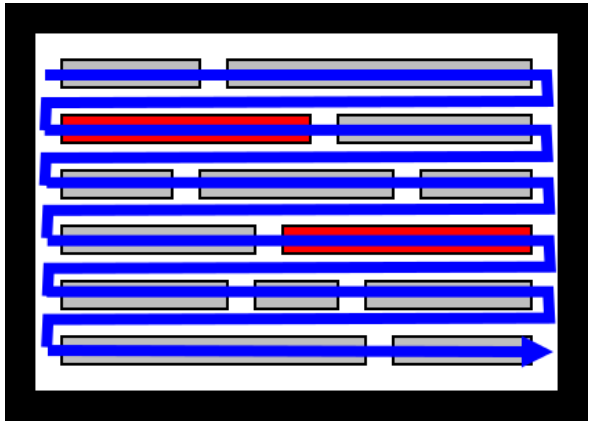
Copy a File



Search for a Given Word



Search for a Given Word



Linear Scan.

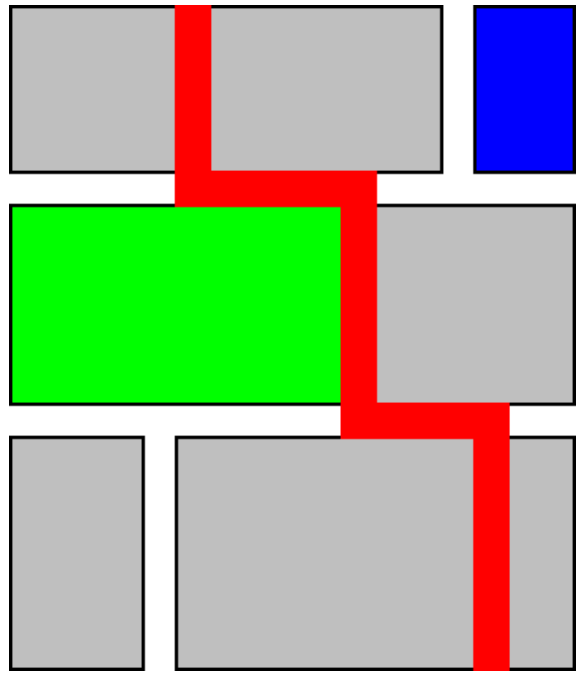
Simple Programming Problems

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

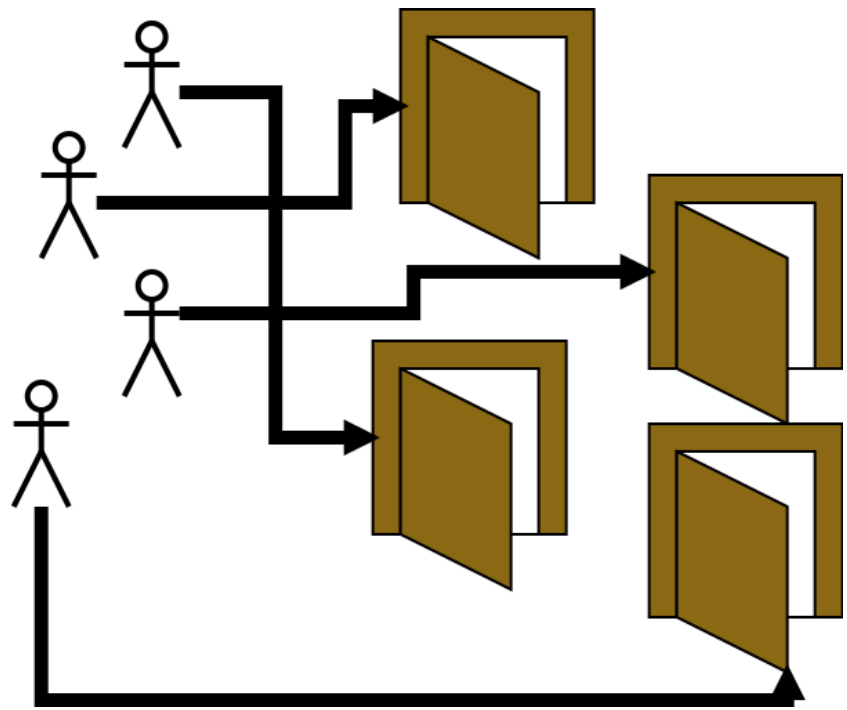
Algorithms Problems

Not so clear what to do.

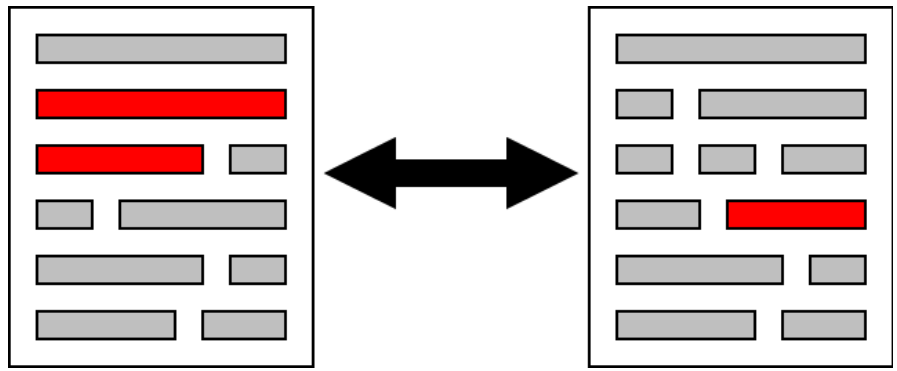
Find the Shortest Path Between Locations



Find the Best Assignment of Students to Dorm Rooms



Measure Similarity of Documents



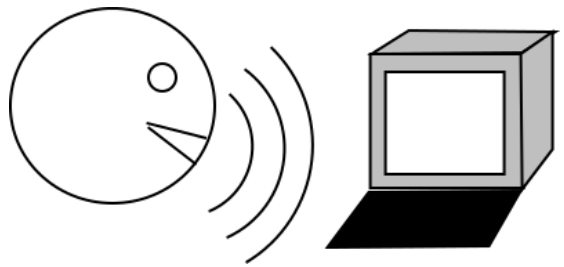
Algorithms Problems

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

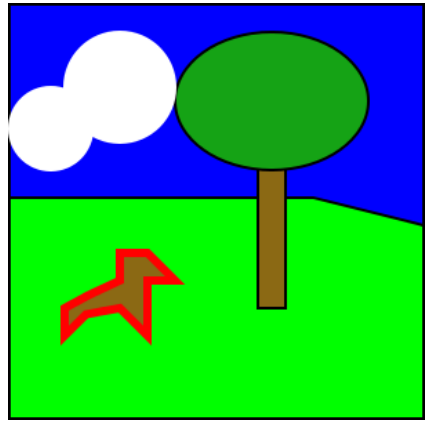
Artificial Intelligence Problems

Hard to even clearly state.

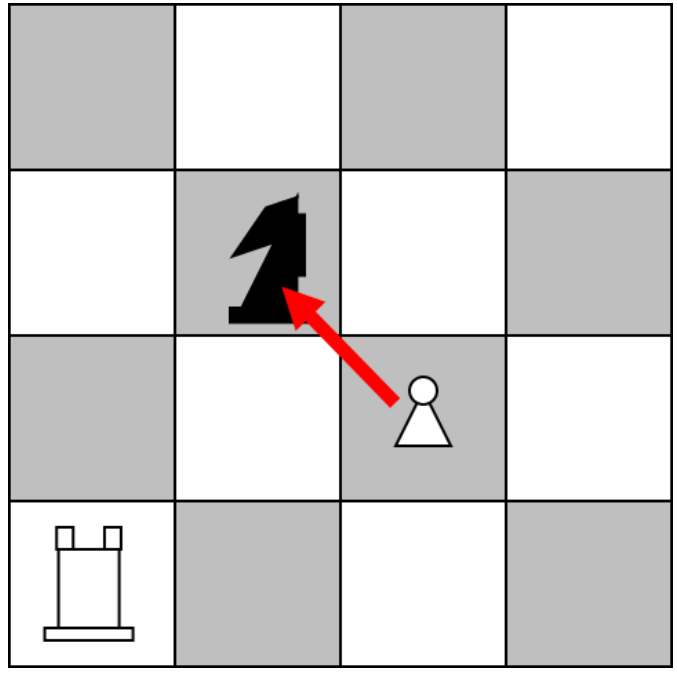
Understand Natural Language



Identify Objects In Photographs



Play Games Well



What We'll Cover

Focus on algorithms problems.

- Clearly formulated.
- Hard to do efficiently.

Intro: Coming Up

Daniel Kane

Department of Computer Science and Engineering
University of California, San Diego

Data Structures and Algorithms
Algorithmic Toolbox

Coming Up

Cover two algorithm problems:

- Fibonacci Numbers
- Greatest Common Divisors

Coming Up

Cover two algorithm problems:

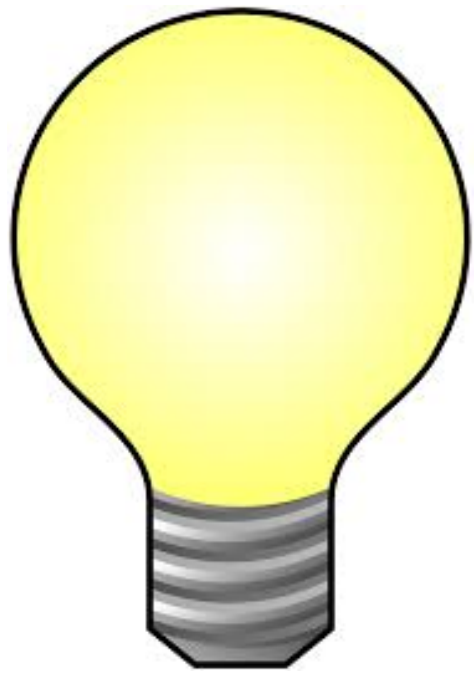
- Fibonacci Numbers
- Greatest Common Divisors

Examples of why algorithms are important.

Straightforward Algorithm

Take Too Long

Slightly More Complicated Algorithm



That is Very Fast