

Algorithms

Algo-what?! Why?

Algorithms

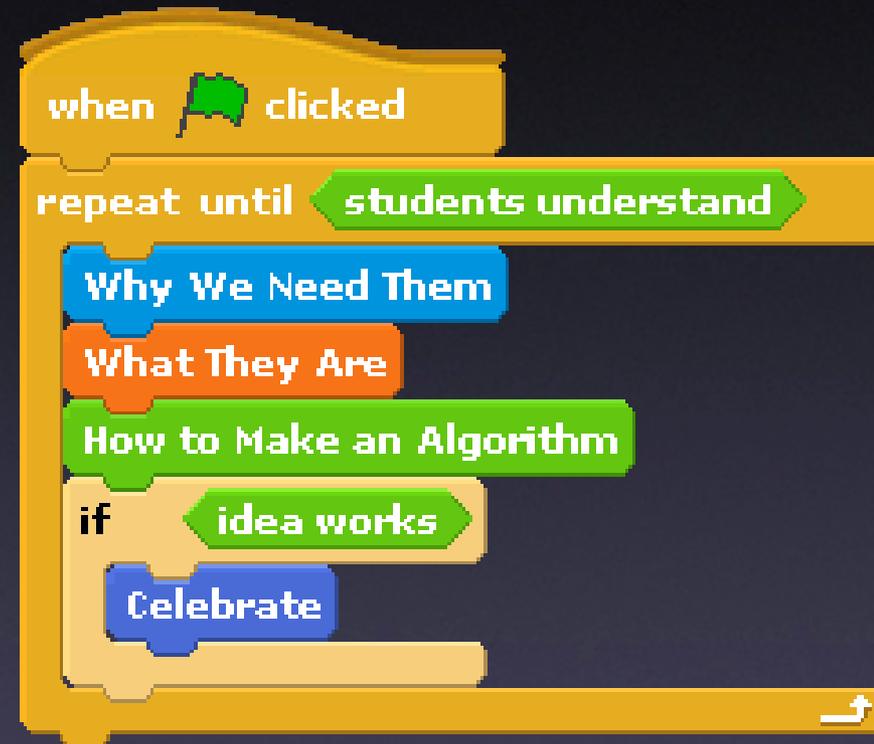
The Plan

Why We Need Them

What They Are

How to Make an Algorithm

Testing Your Idea



Computers are really, really fast.

How fast, you ask?

A reasonably powerful computer in 1961 could perform about 56 multiplication operations in one second.

The most powerful supercomputer in the world today can perform:

a) $< 500,000$ ops per second

c) Between 5,000,000 and 100,000,000 ops per second

b) Between 500,000 and 5,000,000 ops per second

d) $> 100,000,000$ ops per second

Really, really fast

233,000,000

operations per SECOND

Really, really fast

2,330,000,000

operations per SECOND

Really, really fast

23,300,000,000

operations per **SECOND**

Really, really fast

233,000,000,000

operations per SECOND

Really, really fast

2,330,000,000,000

operations per SECOND

Really, really fast

23,300,000,000,000

operations per **SECOND**

Really, really fast

233,000,000,000,000

operations per **SECOND**

Really, really fast

2,330,000,000,000,000

operations per SECOND

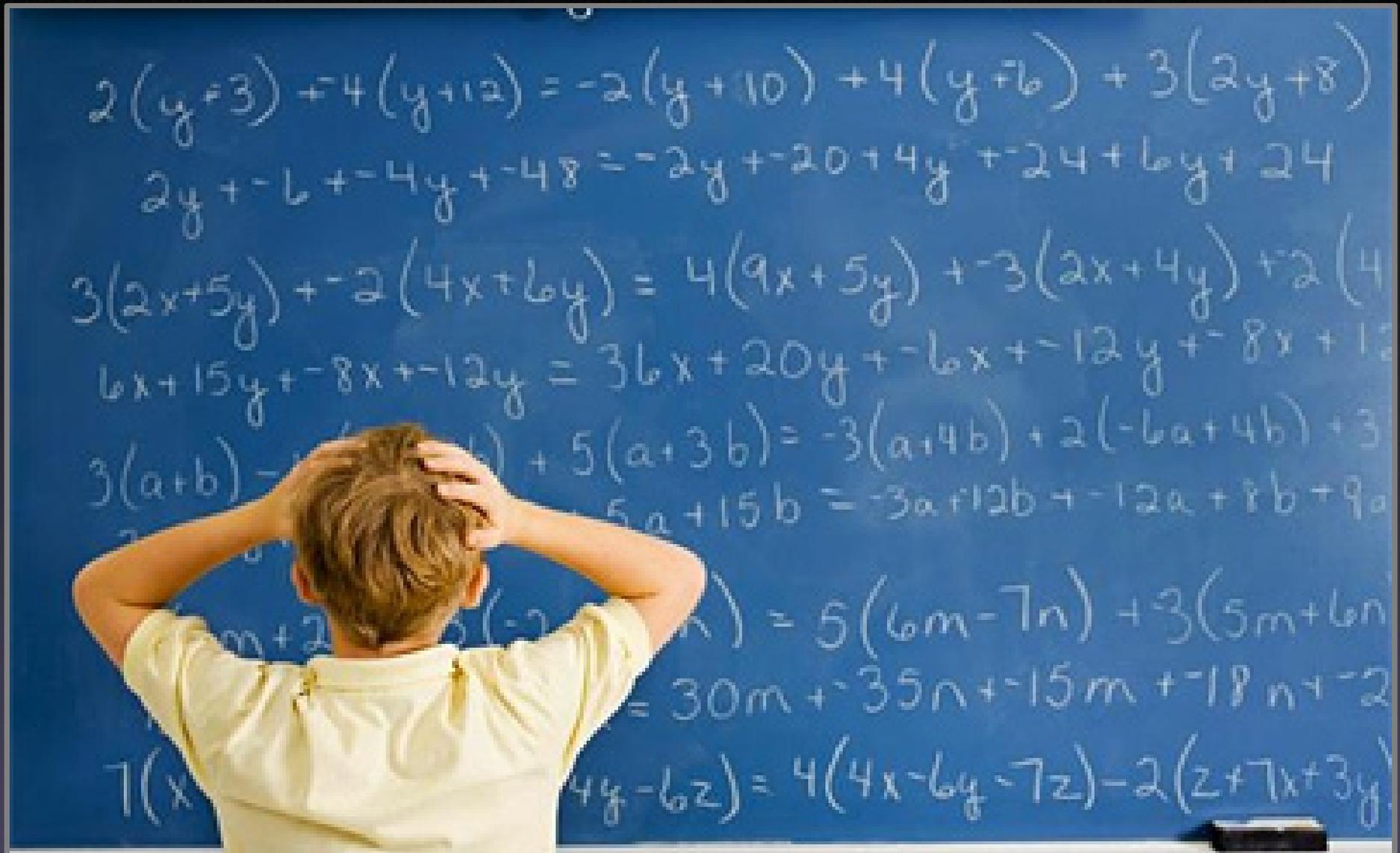
We've got three ways to get our work done faster:

1. Make more computers.
2. Make new computers faster.
3. Make what we're doing faster.

What is an algorithm, anyway?

An algorithm is any well-defined computational procedure that takes some value or set of values as **input** and produces some value or set of values as **output**.

An Algorithm You Already Know and Love*



* = or at least appreciate knowing

The Multiplication Algorithm (for humans)

This algorithm will work on any two numbers, positive or negative. It requires some modification to work with decimal numbers.

$$\begin{array}{r} 182 \\ \times 14 \\ \hline 8 \end{array}$$

The Multiplication Algorithm (for humans)

This algorithm will work on any two numbers, positive or negative. It requires some modification to work with decimal numbers.

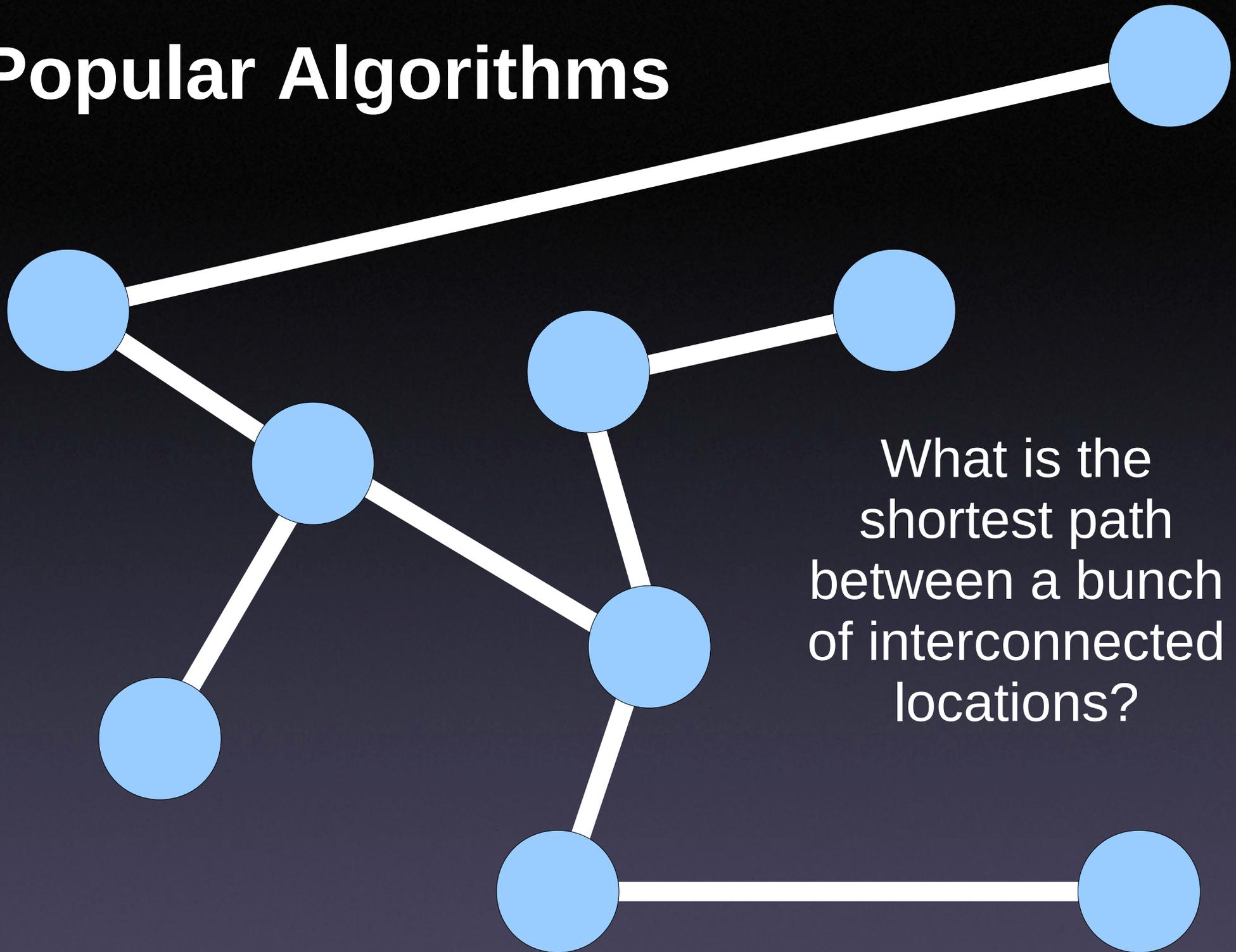
$$\begin{array}{r} \boxed{3} \\ 182 \\ \times 14 \\ \hline 28 \end{array}$$

The Multiplication Algorithm (for humans)

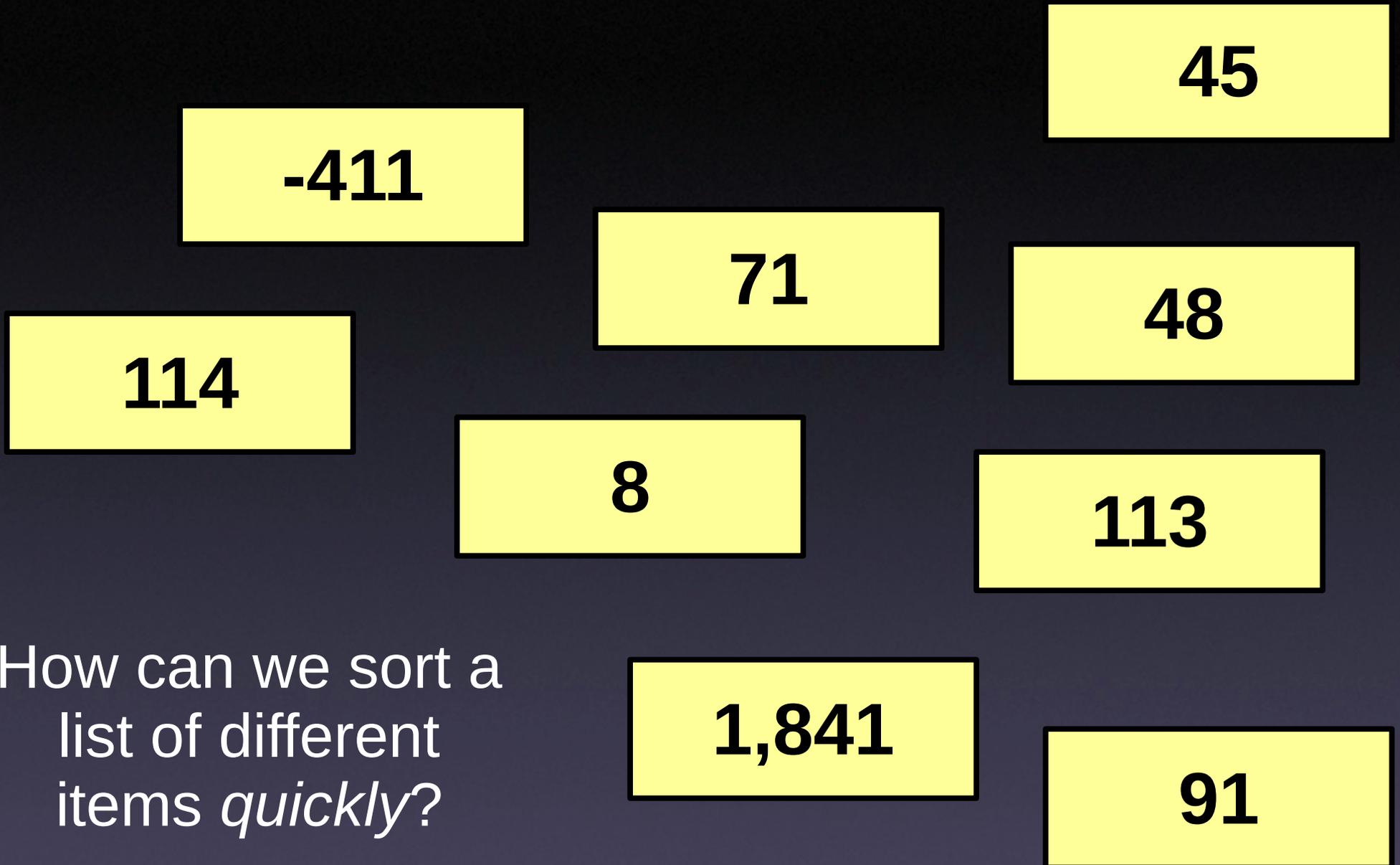
This algorithm will work on any two numbers, positive or negative. It requires some modification to work with decimal numbers.

$$\begin{array}{r} \boxed{3} \\ 182 \\ \times 14 \\ \hline 728 \\ \dots \end{array}$$

Popular Algorithms



Popular Algorithms



How can we sort a list of different items *quickly*?

Popular Algorithms

113

-411

71

8

91

114

How can we
search through a
list of items
quickly?

45

48

1,841

Abstraction in Algorithms

get the shortest path from  to  in 

A block that finds the shortest path between two places in a graph.

sort the list 

A block that (somehow) sorts the input list.

does the number  appear in  ?

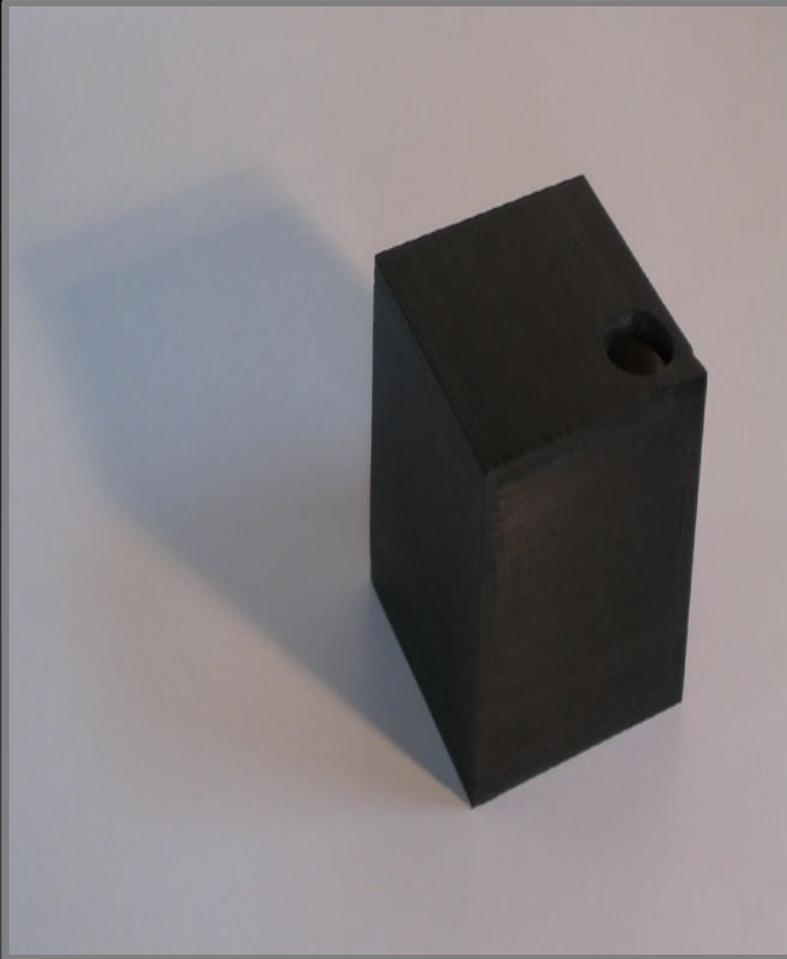
A block that searches for a particular number in a list.

Testing Your Idea



If there is anything wrong with your algorithm / program, your users WILL find it!

Types of Testing



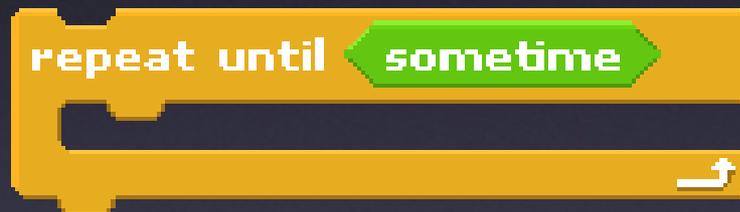
Black Box



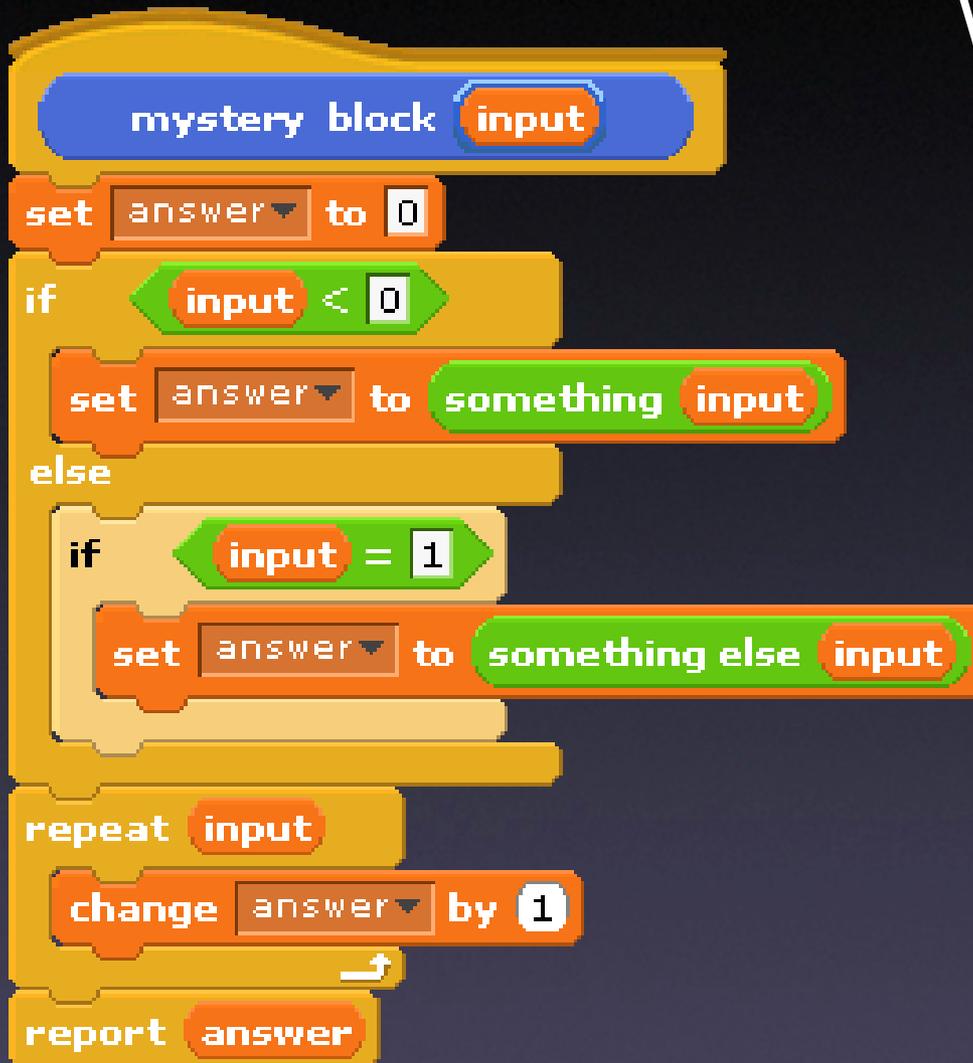
Glass Box

Testing is Really Complex

You want to test as many “paths of execution” as possible!



The Best Test



Which of the following sets of values for *num* would make the best test case for this block?

- a) 1, 2, 3
- b) -1, 1, 3
- c) 12, 100, -3
- d) -1, -2, -3
- e) no testing needed

Food for Thought:

You have a list of numbers sorted in increasing order.

If someone asked if a particular number exists in the list, what is the fastest way to determine whether it does or doesn't?