

national center for

women &

INFORMATION  
TECHNOLOGY™

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum



*Computer Science-in-a-Box: Unplug Your Curriculum* introduces fundamental building blocks of computer science—without using computers. Use it with students ages 9 to 14 to teach lessons about how computers work, while addressing critical mathematics and science concepts such as number systems, algorithms, and manipulating variables and logic. The National Center for Women & Information Technology (NCWIT) is pleased to offer *Computer Science-in-a-Box: Unplug Your Curriculum* in cooperation with the authors of *Computer Science Unplugged*. So unplug your computer, and get ready to explore computer science!



AVAYA

Bank of America



Microsoft



# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## INTRODUCTION

Computers are everywhere and we use them every day, but how many of us understand how they work, or how they think? How can people make them operate faster and better? Computer science is a fascinating subject that explores these very questions. Every student can benefit from an introduction to the science that is possibly most central to their lives — computer science.

*Computer Science-in-a-Box: Unplug Your Curriculum* introduces fundamental building blocks of computer science—without using computers at all. This selection of activities is designed for use with students ages 9 to 14.

Use *Computer Science-in-a-Box: Unplug Your Curriculum* to teach lessons that explain how computers work, and at the same time, address critical mathematics and science concepts from number systems and algorithms to manipulating variables and logic. Presenting these activities to your students will allow them to:

- » Gain fluency with decimal (base10) and binary (base 2) numerical systems
- » Understand how zeros and ones can be used to represent information such as digital images and numbers
- » Understand fundamental ideas of logic and apply logic to solve problems, such as sorting information into useful order quickly
- » Develop an understanding of algorithms that goes beyond basic operations of arithmetic
- » Understand that these concepts are harnessed by computer scientists to solve problems and innovate

These learning outcomes correspond to the Mathematics Focal Points advised by the National Council of Teachers of Mathematics (NCTM) for grades K-8:

- » the use of the mathematics to solve problems
- » an application of logical reasoning to justify procedures and solutions
- » involvement in the design and analysis of multiple representations to learn, make connections among, and communicate about the ideas within and outside of mathematics

The *ACM (Association for Computing Machinery) K-12 Computer Science Model Curriculum*, published by The Computer Science Teachers Association, recommends using *Computer Science Unplugged* activities to address key computing concepts appropriate for Level 1, Foundations of Computer Science.

In addition to giving students experience with applied math and science in a meaningful context, *Computer Science-in-a-Box: Unplug Your Curriculum* also presents opportunities to advance communication skills, problem-solving, and creative thinking. Look for instructional objectives associated with each activity in *Computer Science-in-a-Box: Unplug Your Curriculum*.

You don't have to be a computer expert to enjoy learning these principles with your students. Answers to all problems are provided, and each activity ends with a "what's it all about?" section that explains the relevance of the activities. To help you get ready, the authors have created videos of each activity in action. Look for video links throughout this resource. Links to all the videos can be found at [www.ncwit.org/unplugged](http://www.ncwit.org/unplugged).

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

*Computer Science Unplugged*, the curriculum from which these materials are excerpted, was written by three computer science lecturers and two school teachers, and is based on their direct experience in classrooms (see *About This Resource* for more information about *Computer Science Unplugged*). They found that many important concepts can be taught without using a computer—in fact, sometimes the computer is actually a distraction to the learning. So unplug your computer, and get ready to explore computer science!

## ABOUT THIS RESOURCE

The National Center for Women & Information Technology (NCWIT) is pleased to offer *Computer Science-in-a-Box: Unplug Your Curriculum* in cooperation with the authors of *Computer Science Unplugged*. *Computer Science Unplugged* was created by Tim Bell, Ian H. Witten, and Mike Fellows, and was illustrated by Matt Powell. *Computer Science Unplugged* was adapted for classroom use by Robyn Adams and Jane McKenzie. It was further adapted for this resource, *Computer Science-in-a-Box: Unplug Your Curriculum*, by Jane Krauss and reviewed by David Burkhart and Chris Stephenson.

*Computer Science-in-a-Box: Unplug Your Curriculum* excerpts seven activities from the larger work. *Computer Science Unplugged* is available in its entirety for free download, for personal and educational use, thanks to a generous grant by Google, Inc. Visit: [www.csunplugged.com](http://www.csunplugged.com).

It is likely that once you use these activities with students you will be inspired to do more. Extension activities for each lesson are available, along with many other lessons in the comprehensive curriculum *Computer Science Unplugged*, at [www.csunplugged.com](http://www.csunplugged.com).

*This material is distributed under a Creative Commons Attribution-NonCommercial-NoDerivatives license, which means you are free to copy, distribute, and display the book provided you make no changes to the content (including the attribution to the authors and these license terms); you may not use the book for commercial purposes, and you may not alter, transform, or build upon this work.*

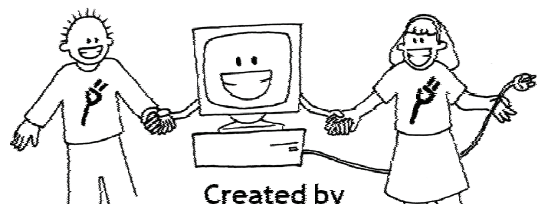
*Computer Science Unplugged* authors welcome comments and suggestions about these activities. Please contact them at the *Computer Science Unplugged* website: [www.csunplugged.com](http://www.csunplugged.com).

NCWIT works to help teachers cultivate the next generation of innovative thinkers and increase the participation of underrepresented populations in computer science and information technology. Visit the NCWIT website to learn more: [www.ncwit.org](http://www.ncwit.org).

We welcome your feedback, input, and user experiences about this resource. Please use the evaluation survey link at [www.ncwit.org/unplugged](http://www.ncwit.org/unplugged) to share your thoughts.

# COMPUTER SCIENCE *Unplugged*

**An enrichment and extension  
programme for primary-aged children**



Created by  
**Tim Bell, Ian H. Witten and Mike Fellows**

## CONTENTS

Introduction	i
About this Resource	ii
 Data: The Raw Material— <i>Representing Information</i>	 5
1. Count the Dots— <i>Binary Numbers</i>	6
2. Color by Numbers— <i>Image Representation</i>	14
3. Card Flip Magic— <i>Error Detection and Correction</i>	21
 Putting Computers to Work— <i>Algorithms</i>	 25
4. Lightest and Heaviest— <i>Sorting Algorithms</i>	27
5. Beat the Clock— <i>Sorting Networks</i>	34
6. The Orange Game— <i>Routing and Deadlock in Networks</i>	40
 Telling Computers What To Do— <i>Representing Procedures</i>	 43
7. Treasure Hunt— <i>Finite-state Automata</i>	45



# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## DATA: THE RAW MATERIAL (*READ ALOUD*)

### How can we store information in computers?

The word “computer” comes from the Latin *computare*, which means to calculate or add together. But computers today are more than just giant calculators. They can be a library, help us to write, find information for us, play music, and even show movies. So how do they store all this information? Believe it or not, the computer uses only two things: zero and one!

### What is the difference between data and information?

Data are the raw material, the numbers that computers work with. A computer converts its data into information (words, numbers and pictures) that you and I can understand.

### How can numbers, letters, words, and pictures be converted into zeros and ones?

Let’s learn how zeros and ones operate in the binary number system, and how computers interpret these data systems to do work and represent information.



## ACTIVITY 1: GETTING READY

### COUNT THE DOTS — *BINARY NUMBERS*

#### Summary

Data in computers is stored and transmitted as a series of zeros and ones. How can we represent words and numbers using just these two symbols?

#### Video Link

[www.ncwit.org/unplugged](http://www.ncwit.org/unplugged)

#### Time Needed

- » 45 minutes for introductory activity
- » 45 minutes additional worksheet extension activities

#### Curriculum Links

- » Mathematics: Numeral Systems: exploring numbers in other bases, representing numbers in base two.
- » Mathematics: Algebra: continuing a sequential pattern, and describing a rule for this pattern. Patterns and relationships in powers of two.

#### Skills

- » Counting
- » Matching
- » Sequencing

#### Ages

- » 9 and up

#### Materials

You will need to make one set of five large binary cards (see page 9) for the class demonstration. 8.5" x 11" cards with smiley face sticker dots work well, too.

#### Each student will need:

A set of five small cards.

**Photocopy Master:** Copy binary numbers (page 8) onto card and cut out. (The photocopy master has four sets. For a class of 24 students, print six pages for 24 sets.)

- » Worksheet Activity: Binary numbers (page 9)

#### For optional extension activities each student or small cooperative group will need:

- » Worksheet Activity: Working with binary (page 10)
- » Worksheet Activity: Counting higher than 31 (page 11)

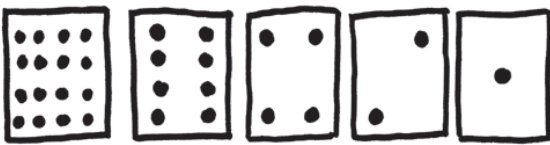
# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## BINARY NUMBERS — TEACHING PLAN

### Introduction to the *Binary Numbers Activity*

Present this activity to introduce and demonstrate binary system principles to the whole group. Give ample practice representing numbers with the cards and reading numbers represented in binary (0s and 1s).

For this activity, you will need a set of five cards, as shown below, with dots on one side and nothing on the other. Choose five students to hold the demonstration cards at the front of the class. The cards should be in the following order:



### Discussion

What do you notice about the number of dots on the cards? (*Each card has twice as many as the card to its right.*)

How many dots would the next card have if we carried on to the left? (32) The next...?

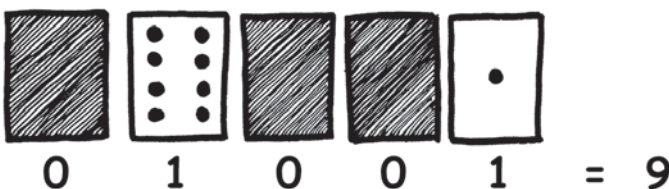
We can use these cards to make numbers by turning some of them face down and adding up the dots that are showing. Ask the students to make the number 6 (4-dot and 2-dot cards), then the number 15 (8-, 4-, 2- and 1-dot cards), then the number 21 (16, 4 and 1)...

Now try counting from zero onwards.

The rest of the class needs to look closely at how the cards change to see if they can see a pattern in how the cards flip (*each card flips half as often as the one to its right*). You may want to try this with more than one group.

After ample group practice, give students an opportunity to study the method and practice individually. Supply each student with a small set of five cards (page 9) and the *Worksheet Activity: Binary Numbers* (page 10) Monitor individual practice, and then assess the class by having them show numbers like 3, 12, 19, and the next bigger or smaller number.

Next, show how to interpret binary representations. Using the example below, show students that when a binary number card is not showing, it is represented by a zero. When it is showing, it is represented by a one. This is the binary number system.

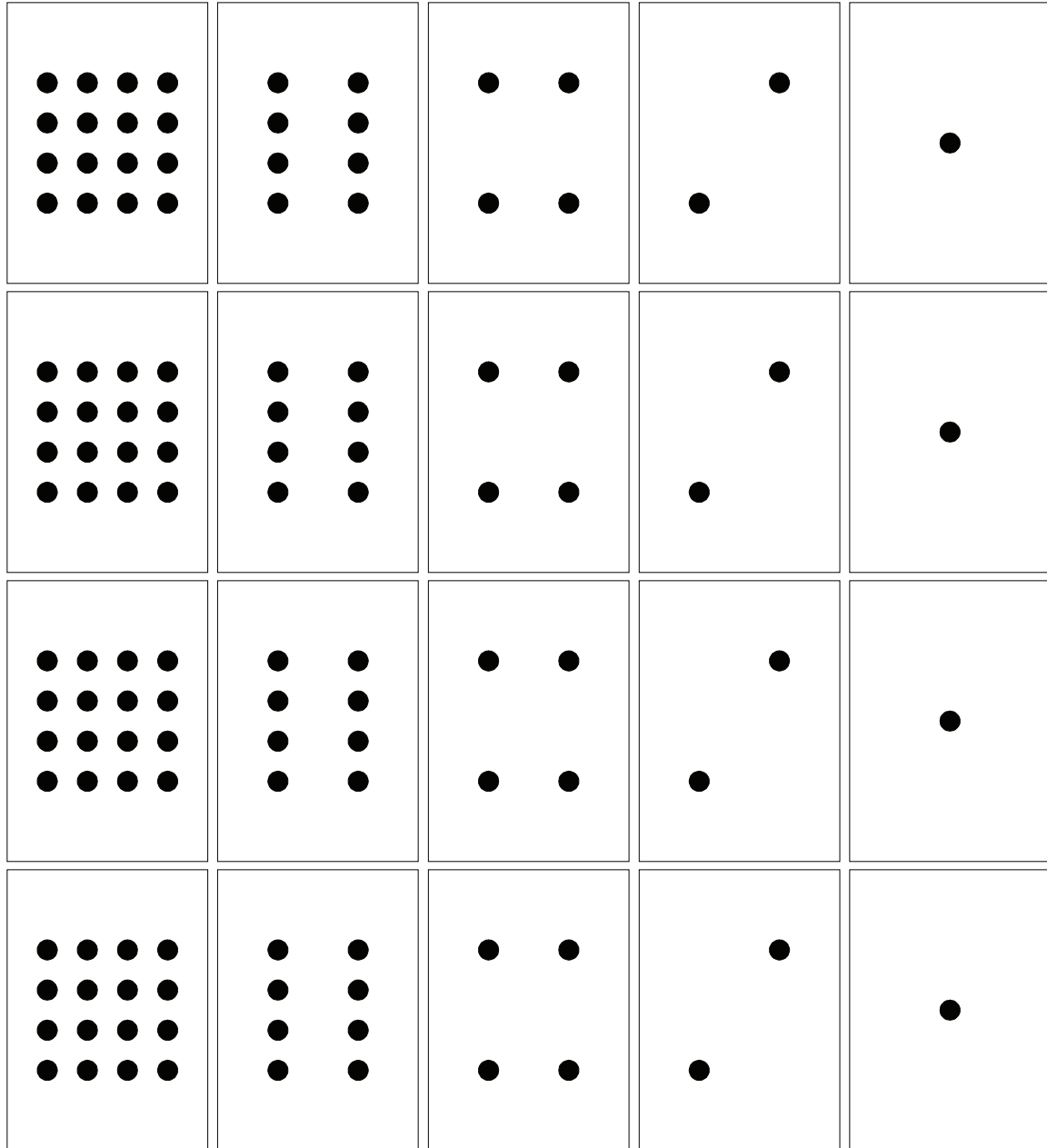


Ask the students to make 01001. What number is this in numerical form? (9) What would 17 be in binary? (10001)

Try a few more until they understand the concept. Use the extension activities for reinforcement. Have students do as many of them as they can.

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## PHOTOCOPY MASTER: *BINARY NUMBERS*





# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## WORKSHEET ACTIVITY: *BINARY NUMBERS FOR STUDENTS*

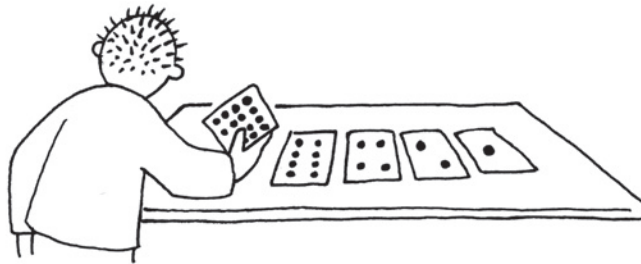
### Learning how to count

So, you thought you knew how to count? Well, here is a new way to do it!

Did you know that computers use only zero and one? Everything that you see or hear on the computer—words, pictures, numbers, movies, and even sound is stored using just those two numbers! These activities will teach you how to send secret messages to your friends using exactly the same method as a computer.

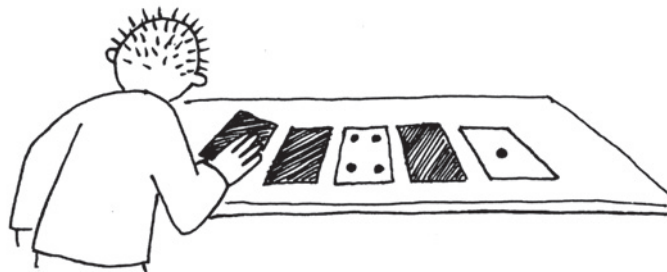
### Instructions

Cut out the cards on your sheet and lay them out with the 16-dot card on the left, as shown here:



Make sure the cards are placed in exactly the same order.

Now flip the cards so exactly 5 dots show—keep your cards in the same order!



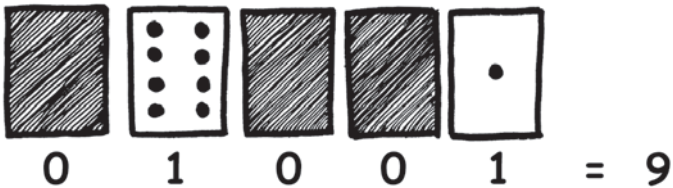
Find out how to make the numbers 3, 12, 19. Is there more than one way to make any number? What is the biggest number you can make? What is the smallest? Is there any number you can't make between the smallest and biggest numbers?

**Extra for Experts:** Try making the numbers 1, 2, 3, 4 in order. Can you work out a logical and reliable method for flipping the cards to increase any number by one?

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## WORKSHEET ACTIVITY: *WORKING WITH BINARY (EXTENSION)*

The binary system uses zero and one to represent whether a card is face-up or not. 0 shows that a card is hidden, and 1 means that it is face-up and you can see the dots. For example:



Can you work out what 10101 is? What about 11111?

What day of the month were you born? Write it in binary. Find out what your friend's birthdays are in binary.

Try to work out these coded numbers:

$$\begin{array}{ccccc} \boxed{\times} & \boxed{\checkmark} & \boxed{\times} & \boxed{\times} & \boxed{\checkmark} \\ (\checkmark=1, \times=0) \end{array} =$$

$$\begin{array}{ccc} \uparrow & \downarrow & \uparrow \\ (\uparrow=1, \downarrow=0) \end{array} =$$

$$\begin{array}{ccccc} \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc \\ (\odot=1, \bigcirc=0) \end{array} =$$

$$\begin{array}{ccc} \text{house icon} & \text{house icon} & \text{house icon} \\ (\text{house icon}=1, \text{house icon}=0) \end{array} =$$

$$\begin{array}{c} \text{sad face icon} \\ (\text{happy face}=1, \text{sad face}=0) \end{array} =$$

$$\begin{array}{cccc} \text{thumbs up} & \text{thumbs down} & \text{thumbs up} & \text{thumbs down} \\ (\text{thumbs up}=1, \text{thumbs down}=0) \end{array} =$$

$$\begin{array}{cccc} + & + & \times & + \\ (+=1, \times=0) \end{array} =$$

$$\begin{array}{ccccc} \curvearrowright & \curvearrowright & \curvearrowright & \curvearrowright & \curvearrowright \\ (\curvearrowright=1, \curvearrowright=0) \end{array} =$$

$$\begin{array}{ccccc} \blacktriangle & \blacktriangledown & \blacktriangle & \blacktriangledown & \blacktriangledown \\ (\blacktriangle=1, \blacktriangledown=0) \end{array} =$$

$$\begin{array}{ccccc} \spadesuit & \spadesuit & \spadesuit & \spadesuit & \spadesuit \\ (\spadesuit=1, \clubsuit=0) \end{array} =$$

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## WORKSHEET ACTIVITY: COUNTING HIGHER THAN 31 (EXTENSION)

Look at the binary cards again. If you were going to make the next card in the sequence, how many dots would it have? What about the next card after that? What is the rule that you are following to make your new cards? As you can see, only a few cards are needed to count up to very big numbers.

If you look at the sequence carefully, you can find a very interesting relationship:

**1, 2, 4, 8, 16 . . .**

Try adding:  $1 + 2 + 4 = ?$  What does it come to?

Now try  $1 + 2 + 4 + 8 = ?$

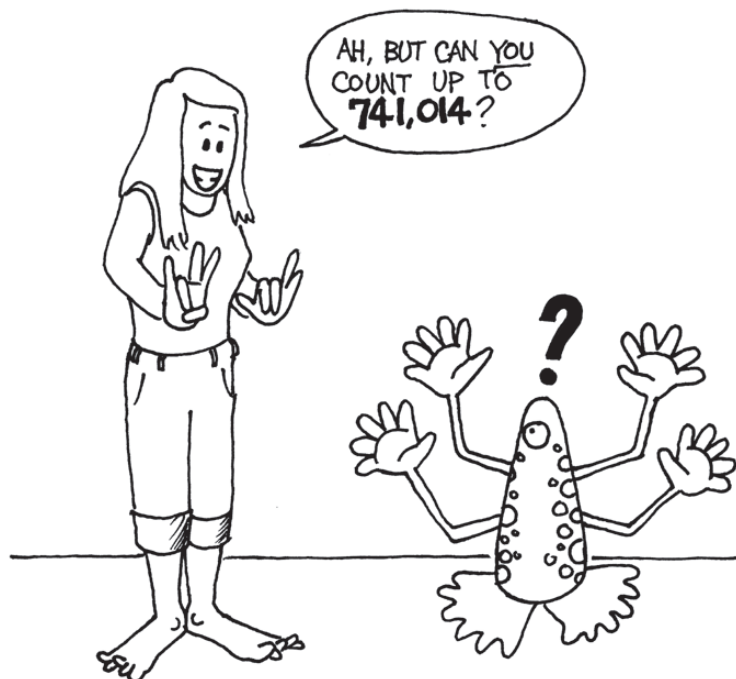
What happens if you add all the numbers up from the beginning?

Have you ever heard the expression, “let your fingers do the walking?” Well now you can let your fingers do the counting, but you can get much higher than ten—no, you don’t have to be an alien! If you use the binary system and let each finger on one hand represent one of the cards with dots, you can count from 0–31. That’s 32 numbers. (Don’t forget that zero is a number too!)

Try counting in order using your fingers. If a finger is up it is a one, and if it is down it is a zero.

You can actually get from 0–1023 if you use both hands! That’s 1024 numbers!

If you had really bendy toes (now you would have to be an alien) you could get even higher. If one hand can be used to count 32 numbers, and two hands can count to  $32 \times 32 = 1024$  numbers, what is the biggest number Miss Flexi-Toes can reach?



# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## WHAT'S IT ALL ABOUT? (BACKGROUND FOR THE TEACHER)

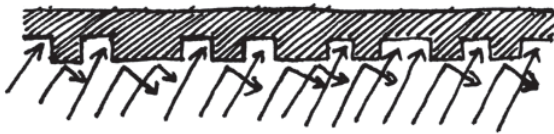
Computers today use the binary system to represent information. It is called binary because only two different digits are used. It is also known as base two (humans normally use base 10). Each zero or one is called a *bit* (binary digit). A bit is usually represented in a computer's main memory by a transistor that is switched on or off, or a capacitor that is charged or discharged.



When data must be transmitted over a telephone line or radio link, high- and low-pitched tones are used for the ones and zeros. On magnetic disks (floppy disks and hard disks) and tapes, bits are represented by the direction of a magnetic field on a coated surface, either North-South or South-North.



Audio CDs, CD-ROMs and DVDs store bits optically—the part of the surface corresponding to a bit either does or does not reflect light.

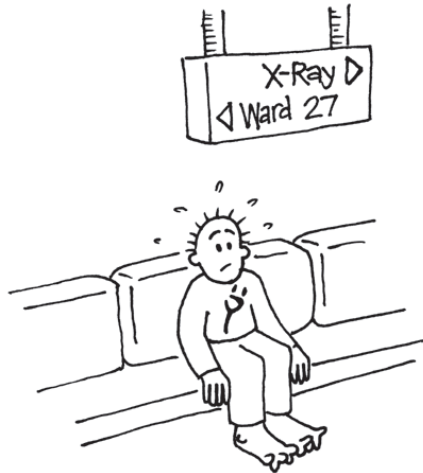


One bit on its own can't represent much, so they are usually grouped together in groups of eight, which can represent numbers from 0 to 255. A group of eight bits is called a byte.

The speed of a computer depends on the number of bits it can process at once. For example, a 32-bit computer can process 32-bit numbers in one operation, while a 16-bit computer must break 32-bit numbers down into smaller pieces, making it slower.

Ultimately bits and bytes are all that a computer uses to store and transmit numbers, text, and all other information. In some of the later activities we will see how other kinds of information can be represented on a computer.

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum



**BEWARE!**  
Miss Flexi-Toes is  
a trained professional!  
Not everyone's toes  
bend so easily!

## SOLUTIONS AND HINTS

### Binary Numbers (page 9)

**3** requires cards 2 and 1

**12** requires cards 8 and 4

**19** requires cards 16, 2 and 1

There is only one way to make any number.

The biggest number you can make is 31. The smallest is 0. You can make every number in between, and each has a unique representation.

**Experts:** To increase any number by one, flip all the cards from right to left until you turn one face up.

### Working with binary (page 10)

$10101 = 21$ ,  $11111 = 31$

### Counting higher than 31 (page 11)

If you add the numbers up from the beginning the sum will always be one less than the next number in the sequence.

Miss Flexi-toes can count  $1024 \times 1024 = 1,048,576$  numbers—from 0 to 1,048,575!



## ACTIVITY 2: GETTING READY

### COLOR BY NUMBERS—*IMAGE REPRESENTATION*

#### Summary

Computers store drawings, photographs, and other pictures using only numbers. The following activity demonstrates how they can do this.

**Video Link:** [www.ncwit.org/unplugged](http://www.ncwit.org/unplugged)

#### Time Needed

- » 45 minutes: Introductory Activity
- » 45 minutes each: Worksheet Activities

#### Curriculum Links

- » Mathematics: Geometry Grade 4 and up. Exploring Shape and Space.

#### Skills

- » Counting
- » Graphing

#### Ages

- » 9 and up

#### Materials

- » Overhead transparency made from Projection Master: Color by Numbers (page 16)

#### Each student will need:

- » Worksheet Activity: Kid Fax (page 17)
- » Worksheet Activity: Make your own picture (page 18)

## COLOR BY NUMBERS — TEACHING PLAN

### Introduction

### Discussion Questions

1. What do facsimile (fax) machines do?
2. In what situations would computers need to store pictures? (*A drawing program, a game with graphics, or a multi-media system.*)
3. How can computers store pictures when they can only use numbers?

(You may want to arrange for the students to send and/or receive faxes as a preparation for this activity)

### Demonstration using overhead transparency



Computer screens are divided up into a grid of small dots called pixels (picture elements).

In a black and white picture, each pixel is either black or white.

The letter “a” has been magnified above to show the pixels. When a computer stores a picture, all that it needs to know is which dots are black and which are white.

					1, 3
					4, 1
					1, 4
					0, 1, 3, 1
					0, 1, 3, 1
					1, 4

The picture above shows us how a picture can be represented by numbers. The first line consists of one white pixel, then three black, then one white. Thus the first line is represented as 1, 3.

The first number always relates to the number of white pixels. If the first pixel is black the line will begin with a zero.

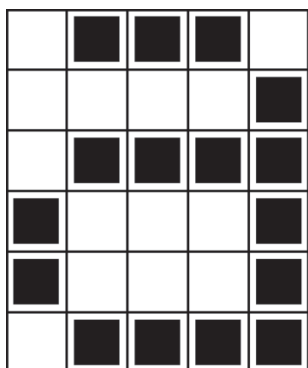
The worksheet on page 3 gives some pictures that the students can decode using the method just demonstrated.

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## OVERHEAD PROJECTION MASTER: *COLOR BY NUMBERS*



⬆ A letter “a” from a computer screen and a magnified view showing the pixels that make up the image



1, 3, 1

4, 1

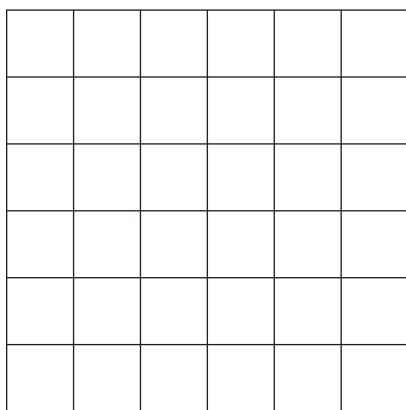
1, 4

0, 1, 3, 1

0, 1, 3, 1

1, 4

⬆ The same image coded using numbers

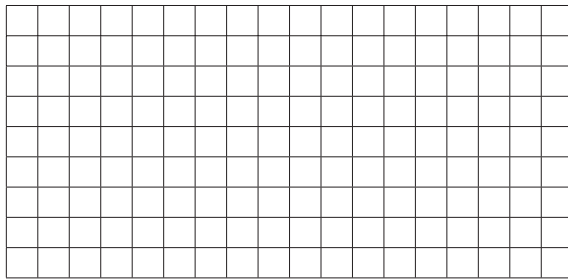


⬆ Blank grid (for teaching purposes)

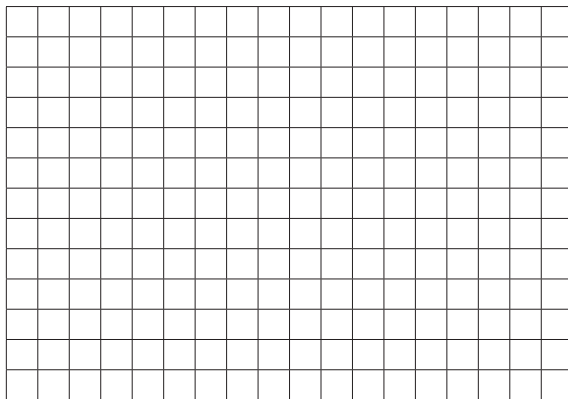
# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## WORKSHEET ACTIVITY: *KID FAX*

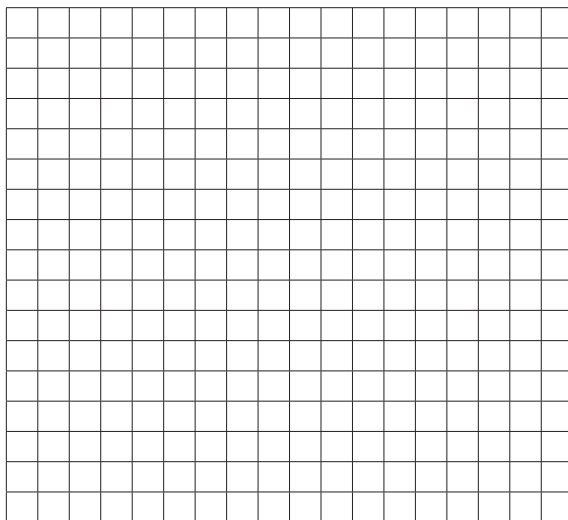
The first picture is the easiest and the last one is the most complex. It is easy to make mistakes and therefore it's a good idea to use a pencil to color with and have an eraser handy!



4, 11  
4, 9, 2, 1  
4, 9, 2, 1  
4, 11  
4, 9  
4, 9  
5, 7  
0, 17  
1, 15



6, 5, 2, 3  
4, 2, 5, 2, 3, 1  
3, 1, 9, 1, 2, 1  
3, 1, 9, 1, 1, 1  
2, 1, 11, 1  
2, 1, 10, 2  
2, 1, 9, 1, 1, 1  
2, 1, 8, 1, 2, 1  
2, 1, 7, 1, 3, 1  
1, 1, 1, 1, 4, 2, 3, 1  
0, 1, 2, 1, 2, 2, 5, 1  
0, 1, 3, 2, 5, 2  
1, 3, 2, 5



6, 2, 2, 2  
5, 1, 2, 2, 2, 1  
6, 6  
4, 2, 6, 2  
3, 1, 10, 1  
2, 1, 12, 1  
2, 1, 3, 1, 4, 1, 3, 1  
1, 2, 12, 2  
0, 1, 16, 1  
0, 1, 6, 1, 2, 1, 6, 1  
0, 1, 7, 2, 7, 1  
1, 1, 14, 1  
2, 1, 12, 1  
2, 1, 5, 2, 5, 1  
3, 1, 10, 1  
4, 2, 6, 2  
6, 6

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## WORKSHEET ACTIVITY: *MAKE YOUR OWN PICTURE*

Now that you know how numbers can represent pictures, why not try making your own coded picture for a friend? Draw your picture on the top grid, and when you've finished, write the code numbers beside the bottom grid. Cut along the dotted line and give the bottom grid to a friend to color in. (Note: you don't have to use the whole grid if you don't want to—just leave some blank lines at the bottom if your picture doesn't take up the whole grid.)

[illegible][illegible]A full page of blank graph paper with a uniform grid of small squares. The grid consists of 20 columns and 20 rows, creating a total of 400 small squares. The lines are thin and black, set against a white background. There are no margins or additional markings on the page.[illegible]



# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## WHAT'S IT ALL ABOUT? *BACKGROUND FOR THE TEACHER*

A fax machine is really just a simple computer that scans a black and white page into about  $1000 \times 2000$  pixels, which are sent using a modem to another fax machine, which prints the pixels out on a page. Often fax images have large blocks of white (e.g. margins) or black pixels (e.g. a horizontal line). Color pictures also have a lot of repetition in them. To save on the amount of storage space needed to keep such images, programmers can use a variety of compression techniques. The method used in this activity is called “run-length coding”, and is an effective way to compress images. If we didn’t compress images it would take much longer to transmit pictures and require much more storage space. This would make it infeasible to send faxes or put photos on a web page. For example, fax images are generally compressed to about one-seventh of their original size. Without compression they would take seven times as long to transmit!

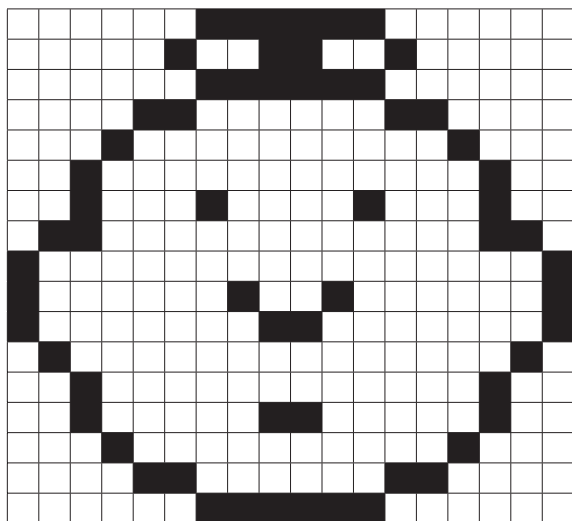
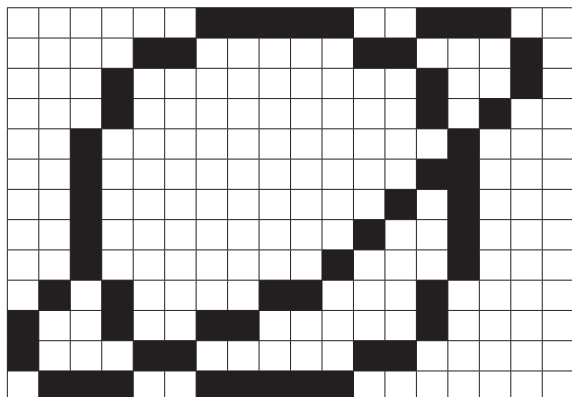
Photographs and pictures are often compressed to a tenth or even a hundredth of their original size (using a different technique). This allows many more images to be stored on a disk, and it means that viewing them over the web will take a fraction of the time.

A programmer can choose which compression technique best suits the images he or she is transmitting.



## SOLUTIONS AND HINTS

### Answers to Kid Fax Worksheet



## ACTIVITY 3: GETTING READY

### CARD FLIP MAGIC—*ERROR DETECTION AND CORRECTION*

#### Summary

When data are stored on a disk or transmitted from one computer to another, we usually assume that they don't get changed in the process. But sometimes things go wrong and the data are changed accidentally. This activity uses a magic trick to show how to detect when data have been corrupted, and how to correct them.

**Video Link:** [www.ncwit.org/unplugged](http://www.ncwit.org/unplugged)

#### Time Needed

- » 45 minutes for Magic Trick Activity
- » 45 minutes for Extension Activities

#### Curriculum Links

- » Mathematics: Number Grade 4 and up. Exploring computation and estimation
- » Algebra Grade 4 and up. Exploring patterns and relationships

#### Skills

- » Counting
- » Recognition of odd and even numbers
- » Visual representation of data

#### Ages

- » 9 years and up

#### Materials

- » A set of 36 “fridge magnet” cards, colored on one side only
- » A metal board (a whiteboard works well) for the demonstration

#### Each pair of students will need:

- » 36 identical cards, colored on one side only

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

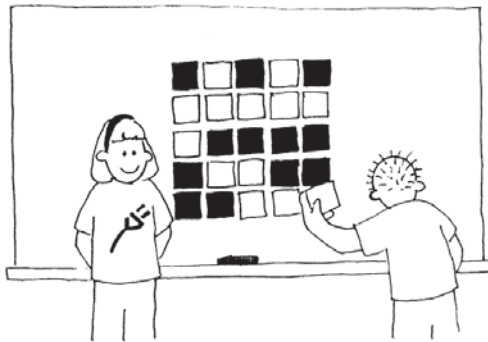
## THE “MAGIC TRICK” (TEACHING PLAN)

### Demonstration

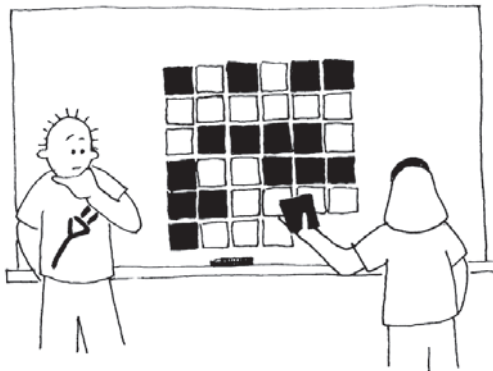
Here’s your chance to be a magician!

You will need a pile of identical, two-sided cards. (To make your own cut up a large sheet of cardstock that is colored on one side only). For the demonstration it is easiest to use flat magnetic cards that have a different color on each side—fridge magnets are ideal.

1. Choose a student to lay out the cards in a  $5 \times 5$  square, with a random mixture of sides showing.



Casually add another row and column, “just to make it a bit harder”.



These cards are the key to the trick. You must choose the extra cards to ensure that there is an even number of colored cards in each row and column.

2. Get a student to flip over one card only while you cover your eyes. The row and column containing the changed card will now have an odd number of colored cards, and this will identify the changed card.

Can the students guess how the trick is done?

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## Teach the trick to the students:

1. Working in pairs, the students lay out their cards  $5 \times 5$ .
2. How many colored cards are there in each row and column? Is it an odd or even number? Remember, 0 is an even number.
3. Now add a sixth card to each row, making sure the number of colored cards is always even. This extra card is called a “parity” card.
4. Add a sixth row of cards along the bottom, to make the number of cards in each column an even number.
5. Now flip a card. What do you notice about the row and column? *(They will have an odd number of colored cards.)* Parity cards are used to show you when a mistake has been made.
6. Now take turns to perform the “trick”.

## Extension Activities:

1. Try using other objects. Anything that has two “states” is suitable. For example, you could use playing cards, coins (heads or tails) or cards with 0 or 1 printed on them (to relate to the binary system).
2. What happens if two, or more, cards are flipped? *(It is not always possible to know exactly which two cards were flipped, although it is possible to tell that something has been changed. You can usually narrow it down to one of two pairs of cards. With four flips it is possible that all the parity bits will be correct afterwards, and so the error could go undetected.)*
3. Another interesting exercise is to consider the lower right-hand card. If you choose it to be the correct one for the column above, then will it be correct for the row to its left? *(The answer is yes, always.)*
4. In this card exercise we have used even parity—using an even number of colored cards. Can we do it with odd parity? *(This is possible, but the lower right-hand card only works out the same for its row and column if the numbers of rows and columns are both even or both odd. For example, a  $5 \times 9$  layout will work fine, or a  $4 \times 6$ , but a  $3 \times 4$  layout won't.)*



# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

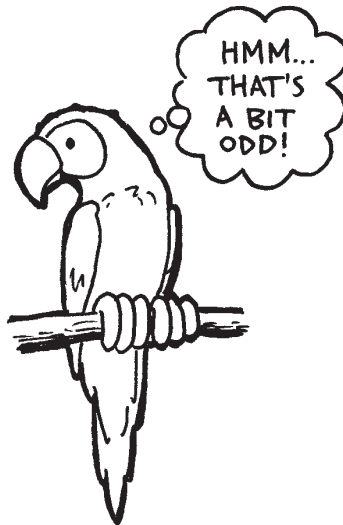
## WHAT'S IT ALL ABOUT? (BACKGROUND FOR THE TEACHER)

Imagine you are depositing \$10 cash into your bank account. The teller types in the amount of the deposit, and it is sent to a central computer. But suppose some interference occurs on the line while the amount is being sent, and the code for \$10 is changed to \$1,000. No problem if you are the customer, but clearly a problem for the bank!

It is important to detect errors in transmitted data. So a receiving computer needs to check that the data coming to it have not been corrupted by some sort of electrical interference on the line. Sometimes the original data can be sent again when an error has been transmitted, but there are some situations when this is not feasible, for example if a disk or tape has been corrupted by exposure to magnetic or electrical radiation, by heat, or by physical damage. If data are received from a deep space probe, it would be very tedious to wait for retransmission if an error had occurred! (It takes just over half an hour to get a radio signal from Jupiter when it is at its closest to Earth!)

We need to be able to recognize when the data have been corrupted (*error detection*) and to be able to reconstruct the original data (*error correction*).

The same technique that was used in the “card flip” game is used on computers. By putting the bits into imaginary rows and columns, and adding parity bits to each row and column, we can not only detect if an error has occurred, but *where* it has occurred. The offending bit is changed back, and so we have performed error correction.



Of course computers often use more complex error control systems that are able to detect and correct multiple errors. The hard disk in a computer has a large amount of its space allocated to correcting errors so that it will work reliably even if parts of the disk fail. The systems used for this are closely related to the parity scheme.

And to finish, here's a joke that is better appreciated after doing this activity:

**Q:** What do you call this: “Pieces of nine, pieces of nine”?

**A:** A parrot error.

## PART II

### PUTTING COMPUTERS TO WORK—*ALGORITHMS*

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## PUTTING COMPUTERS TO WORK (*READ ALOUD*)

Computers operate by following a list of instructions set for them. These instructions enable them to sort, find, and send information. To do these things as quickly as possible, you need good methods for finding things in large collections of data, and for sending information through networks.

An “algorithm” is a set of instructions for completing a task. The idea of an algorithm is central to computer science. Algorithms are how we get computers to solve problems. Some algorithms are faster than others, and many of the algorithms that have been discovered have made it possible to solve problems that previously took an infeasible length of time—for example, finding millions of digits in pi, or all the pages that contain your name on the Internet, or finding out the best way to pack parcels into a container, or finding out whether or not very large (100-digit) numbers are prime.

The word “algorithm” is derived from the name of Mohammed ibn Musa Al-Khowarizmi—Mohammed, son of Moses, from Khowarizm—who joined an academic center known as the House of Wisdom in Baghdad around 800 AD. His works transmitted the Hindu art of reckoning to the Arabs, and then to Europe.

## ACTIVITY 4: GETTING READY

### LIGHTEST AND HEAVIEST—*SORTING ALGORITHMS*

#### Summary

Computers are often used to put lists into some sort of order—for example, names into alphabetical order, appointments or e-mail by date, or items in numerical order. Sorting lists helps us find things quickly, and also makes extreme values easy to see. If you sort the grades for a class test into numerical order, the lowest and highest grades become obvious.

If you use the wrong method, it can take a long time to sort a large list into order, even on a fast computer. Fortunately, several fast methods are known for sorting. In this activity students will discover different methods for sorting, and see how a clever method can perform the task much more quickly than a simple one.

**Video Link:** [www.ncwit.org/unplugged](http://www.ncwit.org/unplugged)

#### Time Needed

- » 45 minutes – Sorting weights with selection sort
- » 45 minutes – Quick sort and other sorting methods

#### Curriculum Links

- » Mathematics: Measurement Grade 6 and up. Carrying out practical weighing tasks

#### Skills

- » Using balance scales
- » Ordering
- » Comparing

#### Ages

- » 12 and up

#### Materials

##### Each group of students will need:

- » Sets of 8 containers of the same size but different weights (*e.g.* milk cartons or film canisters filled with sand)
- » Balance scales
- » Worksheet Activity: Sorting Weights (page 29)
- » Worksheet Activity: Divide and Conquer (page 30)

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## LIGHTEST AND HEAVIEST (*TEACHING PLAN*)

### Discussion

Computers often have to sort lists of things into order. Brainstorm all the places where putting things into order is important. What would happen if these things were not in order?

Computers usually only compare two values at once. The activity on the next page uses this restriction to give students an idea of what this is like.

### Activity

1. Divide the students into groups.
2. Each group will need a copy of the activity sheet on page 29, and its own weights and scales.
3. Have the students do the activity, then discuss the result.



# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## WORKSHEET ACTIVITY: *SORTING WEIGHTS*

**Aim:** To find the best method of sorting a group of unknown weights into order.

**You will need:** Sand or water, 8 identical containers, a set of balance scales

**What to do:**

1. Fill each container with a different amount of sand or water. Seal tightly.
2. Mix them up so that you no longer know the order of the weights.
3. Find the lightest weight. What is the easiest way of doing this?

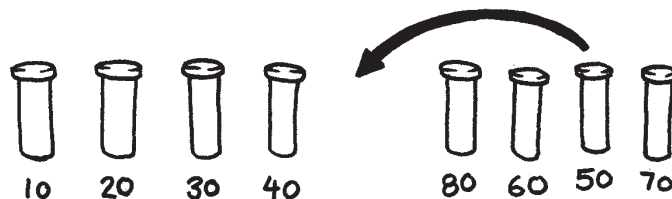
**Note:** You are only allowed to use the scales to find out how heavy each container is. Only two weights can be compared at a time.

4. Choose three weights at random and sort them into order from lightest to heaviest using only the scales.  
How did you do this? What is the minimum number of comparisons you can make? Why?
5. Now sort all of the objects into order from lightest to heaviest.

When you think you have finished, check your ordering by re-weighing each pair of objects standing together.

### Selection Sort

One method a computer might use is called *selection sort*. This is how selection sort works. First find the lightest weight in the set and put it to one side. Next, find the lightest of the weights that are left, and remove it. Repeat this until all the weights have been removed.



Count how many comparisons you made.

**Extra for Experts:** Show how you can calculate mathematically how many comparisons you need to make to sort 8 objects into order. What about 9 objects? 20?

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## WORKSHEET ACTIVITY: *DIVIDE AND CONQUER*

### Quicksort

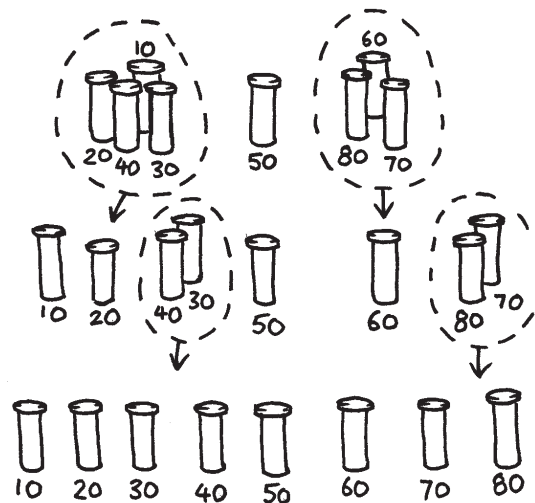
Quicksort is a lot faster than selection sort, particularly for larger lists. In fact, it is one of the best methods known. This is how quicksort works.

Choose one of the objects at random, and place it on one side of the balance scales.

Now compare each of the remaining objects with it. Put those that are lighter on the left, the chosen object in the middle, and the heavier ones on the right. (By chance you may end up with many more objects on one side than on the other.)

Choose one of the groups and repeat this procedure. Do the same for the other group. Remember to keep the one you know in the centre.

Keep repeating this procedure on the remaining groups until no group has more than one object in it. Once all the groups have been divided down to single objects, the objects will be in order from lightest to heaviest.



How many comparisons did this process take?

You should find that quicksort is a more efficient method than selection sort unless you happen to have chosen the lightest or heaviest weight to begin with. If you were lucky enough to have chosen the middle weight, you should have taken only 14 comparisons, compared with the 28 for selection sort. At any rate the quicksort method will never be any worse than selection sort and may be much better!

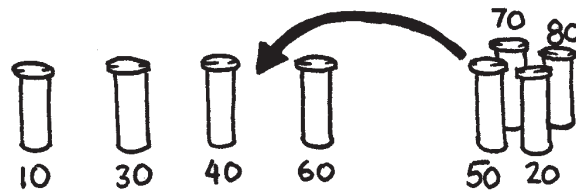
**Extra for Experts:** If quicksort accidentally always chose the lightest object, how many comparisons would it use?

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

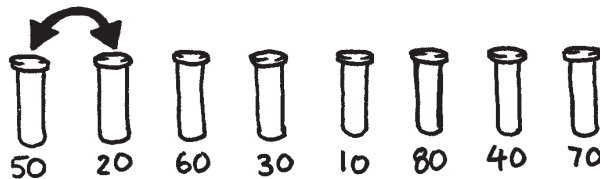
## VARIATIONS AND EXTENSIONS

Many different methods for sorting have been invented. You could try sorting your weights using these:

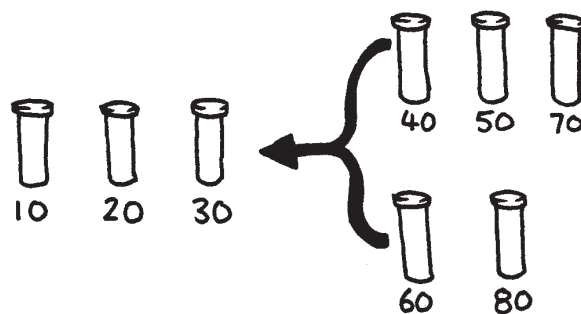
*Insertion sort* works by removing each object from an unsorted group and inserting it into its correct position in a growing list (see picture below). With each insertion the group of unsorted objects shrinks and the sorted list grows, until eventually the whole list is sorted. Card players often use this method to sort a hand into order.



*Bubble sort* involves going through the list again and again, swapping any objects side-by-side that are in the wrong order. The list is sorted when no swaps occur during a pass through the list. This method is not very efficient, but some people find it easier to understand than the others.



*Mergesort* is another method that uses “divide and conquer” to sort a list of items. First, the list is divided at random into two lists of equal size (or nearly equal if there are an odd number of items). Each of the two half-size lists is sorted, and the two lists are merged together. Merging two sorted lists is easy—you repeatedly remove the smaller of the two items at the front of the two lists. In the figure below, the 40- and 60-gram weights are at the front of the lists, so the next item to add is the 40-gram weight. How do you sort the smaller lists? Simple—just use mergesort! Eventually, all the lists will be cut down into individual items, so you don’t need to worry about knowing when to stop.



# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## WHAT'S IT ALL ABOUT? (*BACKGROUND FOR THE TEACHER*)

Information is much easier to find in a sorted list. Telephone directories, dictionaries, and book indexes all use alphabetical order, and life would be far more difficult if they didn't. If a list of numbers (such as a list of expenses) is sorted into order, the extreme cases are easy to see because they are at the beginning and end of the list. Duplicates are also easy to find, because they end up together.

Computers spend a lot of their time sorting things into order, so computer scientists have to find fast and efficient ways of doing this. Some of the slower methods such as insertion sort, selection sort, and bubble sort can be useful in special situations, but fast ones such as quicksort are usually used.

Quicksort uses a concept called recursion. This means you keep dividing a list into smaller parts, and then performing the same kind of sort on each of the parts. This particular approach is called *divide and conquer*. The list is divided repeatedly until it is small enough to conquer. For quicksort, the lists are divided until they contain only one item. It is trivial to sort one item into order! Although this seems very involved, in practice it is dramatically faster than other methods.

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## SOLUTIONS AND HINTS

The best way to find the lightest weight is to go through each object in turn, keeping track of the lightest one so far. That is, compare two objects, and keep the lighter one. Now compare that with another, keeping the lighter from the comparison. Repeat until all the objects have been used.

Compare the weights on the balance scales. This can easily be done with three comparisons, and sometimes just two will suffice—if the students realize that the comparison operator is transitive (that is, if A is lighter than B and B is lighter than C, then A must be lighter than C).

### Experts:

Here is a short-cut for adding up the number of comparisons that selection sort makes.

To find the minimum of two objects you need one comparison, three needs two, four needs three, and so on. To sort eight objects using selection sort takes seven comparisons to find the first one, six to find the next, five to find the next and so on. That gives us:

$$7 + 6 + 5 + 4 + 3 + 2 + 1 = 28 \text{ comparisons.}$$

$n$  objects will take  $1 + 2 + 3 + 4 + \dots + n - 1$  comparisons to sort.

Adding up these numbers is easy if we regroup them.

For example, to add up the numbers  $1 + 2 + 3 + \dots + 20$ , regroup them as

$$(1 + 20) + (2 + 19) + (3 + 18) + (4 + 17) + (5 + 16) +$$

$$(6 + 15) + (7 + 14) + (8 + 13) + (9 + 12) + (10 + 11)$$

$$= 21 \times 10$$

$$= 210$$

In general, the sum  $1 + 2 + 3 + 4 + \dots + n - 1 = n(n - 1)/2$ .

## ACTIVITY 5: GETTING READY

### BEAT THE CLOCK—*SORTING NETWORKS*

#### Summary

Even though computers are fast, there is a limit to how quickly they can solve problems. One way to speed things up is to use several computers to solve different parts of a problem. In this activity we use sorting networks which do several sorting comparisons at the same time.

**Video Link:** [www.ncwit.org/unplugged](http://www.ncwit.org/unplugged)

#### Time Needed

- » 45 minutes – Sorting Networks activity
- » 45 minutes each – Extension Activities (optional)

#### Curriculum Links

- » Mathematics: Number Grade 6 and up. Exploring number: Greater than, less than

#### Skills

- » Comparing
- » Ordering
- » Developing algorithms
- » Cooperative problem solving

#### Ages

- » 12 years and up

#### Materials

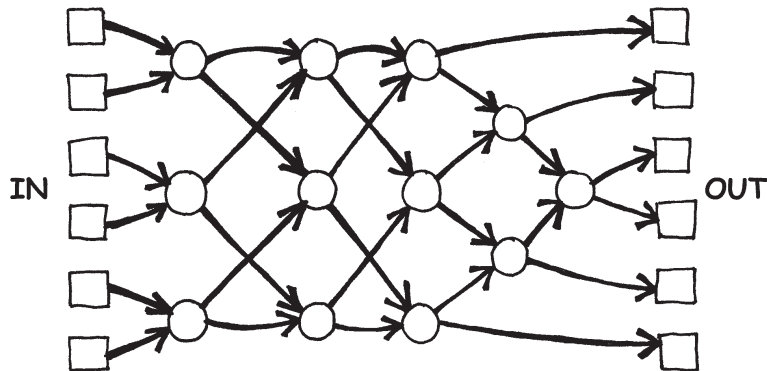
**This is an outdoor group activity.**

- » Chalk
- » Two sets of six cards
- » Copy Photocopy Master: Sorting Networks (page 36) onto card and cut out
- » Stopwatch

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## SORTING NETWORKS

Prior to the activity use chalk to mark out this network on a court.

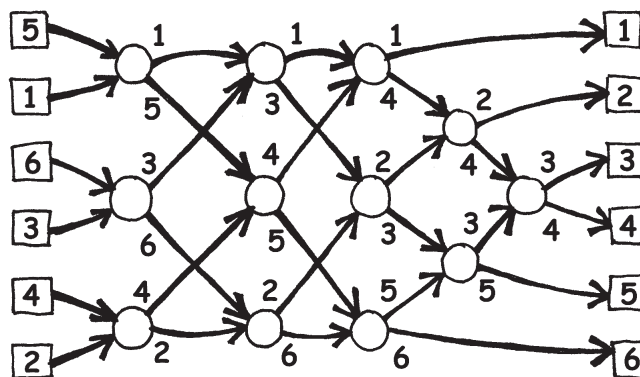


### Instructions for Students

This activity will show you how computers sort random numbers into order using a thing called a sorting network.

1. Organize yourselves into groups of six. Only one team uses the network at a time.
2. Each team member takes a numbered card.
3. Each member stands in a square on the left hand (in) side of the court. Your numbers should be in jumbled order.
4. Move along the lines marked, and when you reach a circle you must wait for someone else to arrive.
5. When another team member arrives in your circle compare your cards. The person with the smaller number takes the exit to their left. If you have the higher number on your card take the right exit.
6. Are you in the right order when you get to the other end of the court?

If a team makes an error the students must start again. Check that you have understood the operation of a node (circle) in the network, where the smaller value goes left and the other goes right. For example:





# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## PHOTOCOPY MASTER: *SORTING NETWORKS*

1

2

3

4

5

6

156

221

289

314

422

499

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

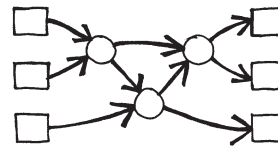
## Variations

1. When the students are familiar with the activity use a stopwatch to time how long each team takes to get through the network.
2. Use cards with larger numbers (e.g. the three-digit ones in the photocopy master).
3. Make up cards with even larger numbers that will take some effort to compare, or use words and compare them alphabetically.

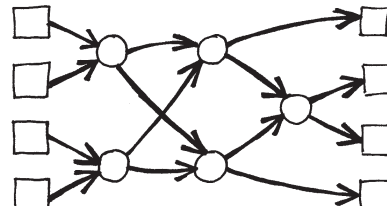
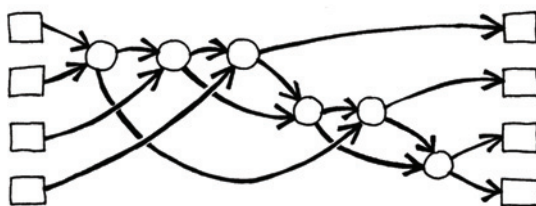
## Extension Activities

1. What happens if the smaller one goes right instead of left and vice versa? (The numbers will be sorted in reverse order.) Does it work if the network is used backwards? (It will not necessarily work, and the students should be able to find an example of an input that comes out in the wrong order.)

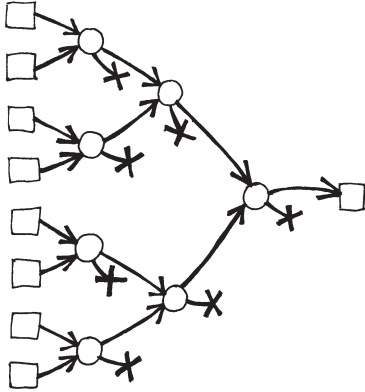
2. Try to design smaller or larger networks. For example, here is a network that sorts just three numbers. The students should try to come up with this on their own.



3. Below are two different networks that will sort four inputs. Which is the faster? *(The second one is. Whereas the first requires all comparisons to be done serially, one after the other, the second has some being performed at the same time. The first network is an example of serial processing, whereas the second uses parallel processing to run faster.)*



# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum



4. Try to make a larger sorting network.
5. Networks can also be used to find the minimum or maximum value of the inputs. For example, here is a network with eight inputs, and the single output will contain the minimum of the inputs (the other values will be left at the dead ends in the network).
6. What processes from everyday life can or can't be accelerated using parallelism? For example, cooking a meal would be a lot slower using only one cooking element, because the items would have to be cooked one after another. What jobs can be completed faster by employing more people? What jobs can't?

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## WHAT'S IT ALL ABOUT? (*BACKGROUND FOR THE TEACHER*)

As we use computers more and more, we want them to process information as quickly as possible.

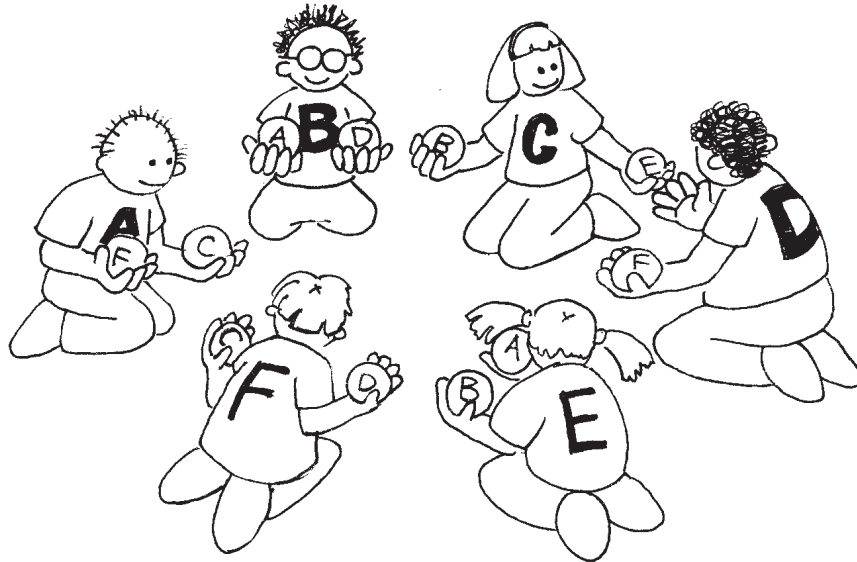
One way to increase the speed of a computer is to write programs that use fewer computational steps.

Another way to solve problems faster is to have several computers work on different parts of the same task at the same time. For example, in the six-number sorting network, although a total of 12 comparisons are used to sort the numbers, up to three comparisons are performed simultaneously. This means that the time required will be that needed for just five comparison steps. This parallel network sorts the list more than twice as quickly as a system that can only perform one comparison at a time.

Not all tasks can be completed faster by using parallel computation. As an analogy, imagine one person digging a ditch ten meters long. If ten people each dug one meter of the ditch the task would be completed much faster. However, the same strategy could not be applied to a ditch ten meters deep—the second meter is not accessible until the first meter has been dug. Computer scientists are still actively trying to find the best ways to break problems up so that they can be solved by computers working in parallel.

## ACTIVITY 6: GETTING READY

### THE ORANGE GAME—ROUTING AND DEADLOCK IN NETWORKS



#### Summary

When you have a lot of people using one resource (such as cars using roads, or messages getting through the Internet), there is the possibility of “deadlock.” A way of working cooperatively is needed to keep this from happening.

#### Curriculum Links

- » Mathematics: Developing logic and reasoning

#### Skills

- » Cooperative problem solving
- » Logical reasoning

#### Ages

- » 9 years and up

#### Materials

##### Each student will need:

- » Two oranges or tennis balls
- » Name tag or sticker

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## THE ORANGE GAME (TEACHING PLAN)

### Introduction

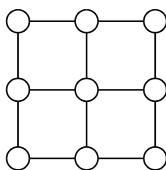
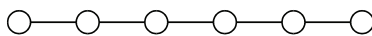
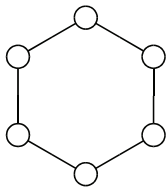
This is a cooperative problem-solving game. The aim is for each person to end up holding the oranges labeled with his or her own letter.

1. Groups of five or more students sit in a circle.
2. The students are labeled with a letter of the alphabet (using name tags or stickers). There are two oranges with each student's letter on them, except for one student, who only has one corresponding orange, to ensure that there is always an empty hand.
3. Distribute the oranges randomly to the students in the circle. Each student has two oranges, except for one student who has only one. (No student should have an orange with their letter on it.)
4. The students pass the oranges around until each student gets the oranges labeled with their letter of the alphabet. You must follow two rules:
  - a) Only one orange may be held in a hand.
  - b) An orange can only be passed to an empty hand of an immediate neighbor in the circle. (A student can pass either of their two oranges to their neighbor.)

Students will quickly find that if they are “greedy” (holding onto their own oranges as soon as they get them) then the group might not be able to attain its goal. It may be necessary to emphasize that individuals don’t “win” the game, but that the puzzle is solved when everyone has their oranges.

### Follow up Discussion

What strategies did the students use to solve the problem?



Where in real life have you experienced deadlock? *(Some examples might be a traffic jam, getting players around bases in baseball, or trying to get a lot of people through a doorway at once.)*

### Extension Activities

Try the activity with a smaller or larger circle.

- » Have the students come up with new rules.
- » Carry out the activity without any talking.
- » Try different configurations such as sitting in a line, or having more than two neighbors for some students. Some suggestions are shown here.

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## WHAT'S IT ALL ABOUT? (*BACKGROUND FOR THE TEACHER*)

Routing and deadlock are problems in many networks, such as road systems and telephone and computer systems. Engineers spend a lot of time figuring out how to solve these problems—and how to design networks that make the problems easier to solve.

Routing, congestion, and deadlock can present frustrating problems in many different networks. Just think of rush-hour traffic! It has happened several times in New York City that the traffic in the streets has become so congested that it deadlocks: no one can move their car! Sometimes when the computers are “down” in businesses (such as banks) the problem is caused by a communication network deadlock. Designing networks so that routing is easy and efficient and congestion is minimized is a difficult problem faced by many kinds of engineers.

Sometimes more than one person wants the same data at the same time. If a piece of data (such as a customer's bank balance) is being updated, it is important to “lock” it during the update. If it is not locked, someone else could update it at the same time and the balance might be recorded incorrectly. However, if this locking is interfered with by the locking of another item, deadlock may occur.

One of the most exciting developments in computer design is the advent of parallel computing, where hundreds or thousands of PC-like processors are combined (in a network) to form a single powerful computer. Many problems like the Orange Game must be played on these networks continuously (but much faster!) in order for these parallel computers to work.



# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## PART III

### TELLING COMPUTERS WHAT TO DO— *REPRESENTING PROCEDURES*

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

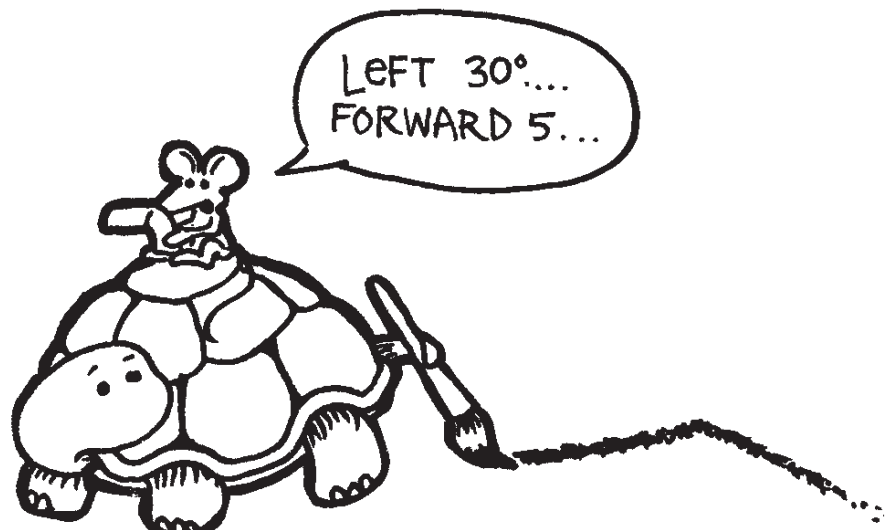
## TELLING COMPUTERS WHAT TO DO (*READ ALOUD*)

Computers follow instructions—millions of instructions—every second. To tell a computer what to do, all you have to do is give it the right instructions. But that’s not as easy as it sounds!

When we are given instructions we use common sense to interpret what is meant. If someone says “go through that door,” they don’t mean to actually smash through the door—they mean go through the doorway, and if necessary, open the door first! Computers are different. When they are attached to mobile robots you need to be careful to take safety precautions to avoid them causing damage and danger by interpreting instructions literally—like trying to go through doors. Dealing with something that obeys instructions literally, without “thinking,” takes some getting used to.

The Treasure Hunt activity in this section gives us some idea of what it is like to communicate to literal-minded machines using a fixed set of instructions.

Treasure Hunt will teach us about a “machine” that computers use to recognize words, numbers, or strings of symbols that the computer can work with. These “machines” are called finite-state automata.



## ACTIVITY 7: GETTING READY

### TREASURE HUNT—*FINITE-STATE AUTOMATA*

#### Summary

Computer programs often need to process a sequence of symbols such as letters or words in a document, or even the text of another computer program. Computer scientists often use a finite-state automaton to do this. A finite-state automaton (FSA) follows a set of instructions to see if the computer will recognize the word or string of symbols. We will be working with something equivalent to a FSA—treasure maps!

#### Time Needed

- » 45 minutes – Demonstration Activity
- » 45 minutes – Extension Worksheet Activities

#### Curriculum Links

- » Mathematics: Developing logic and reasoning—using words and symbols to describe and continue patterns
- » Social Studies – Geography, mapping
- » English

#### Skills

- » Simple map reading
- » Recognizing patterns
- » Logic
- » Following instructions

#### Ages

- » 9 and up

#### Materials

##### You will need:

- » One set of island cards (the instructions must be kept hidden from those trying to draw the map!)

**Photocopy Master:** Copy the Island Cards (pages 51-54) and cut out. Fold along the dotted line and glue, so that the front of the card has the name of the island, and the back has the instructions.

##### Each student will need:

- » Worksheet Activity: Find your Way to the Riches on Treasure Island (page 49)
- » Pen or pencil

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## TREASURE ISLAND (TEACHING PLAN)

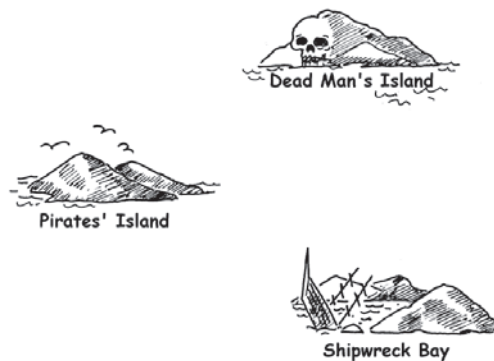
### Introduction

Your goal is to find Treasure Island. Friendly pirate ships sail along a fixed set of routes between the islands in this part of the world, offering rides to travelers. Each island has two departing ships, A and B, which you can choose to travel on. You need to find the best route to Treasure Island. At each island you may ask for either Ship A or Ship B (not both). The person at the island will tell you where your ship will take you next, but the pirates don't have a map of all the islands available. Use your map to keep track of where you are going and which ship you have travelled on.

### Demonstration

(**Note:** This is a different map from the actual activity.)

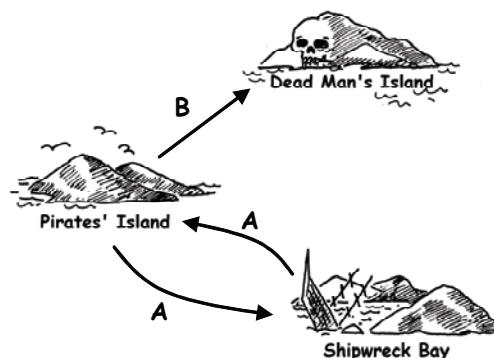
Using an overhead projector or blackboard, draw a diagram of three islands as shown here:



Copy the three cards on the next two pages, and have one student hold each card. Note that the routes on these cards are different from those in the main activity.

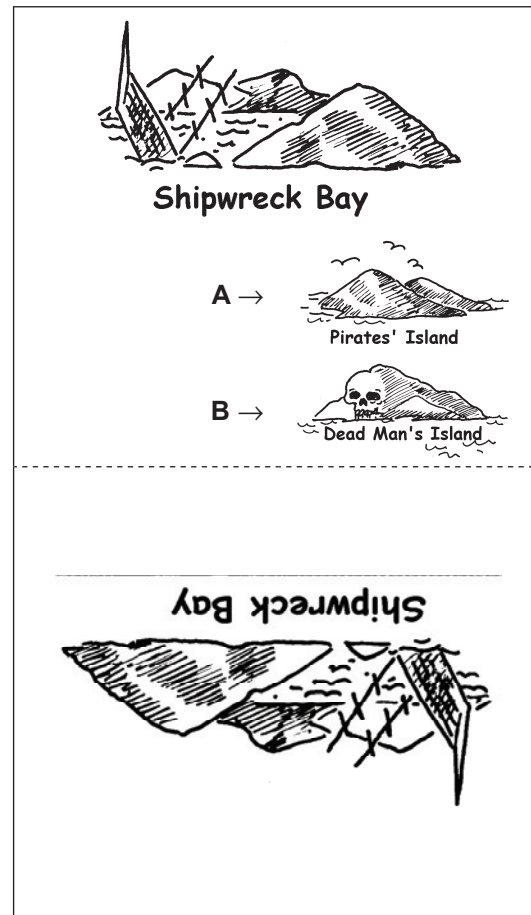
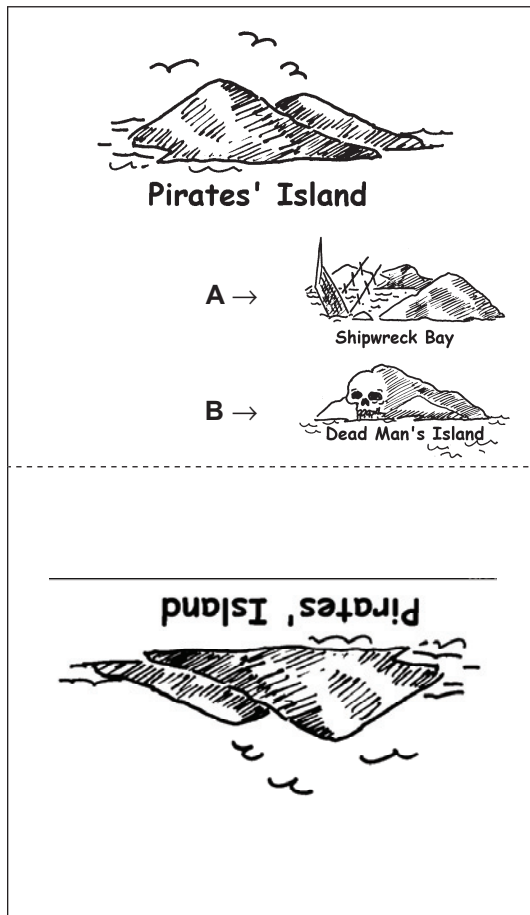
Starting at Pirates' Island, ask for Ship A. The student should direct you to Shipwreck Bay. Mark the route in on the map. At Shipwreck Bay ask for Ship A again. You will be directed back to Pirates' Island. Mark this on the map. This time ask for Ship B. Mark this on the map. This route goes to Dead Man's Island, at which stage you will be stuck!

Your final map should look like this:



# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## CARDS FOR DEMONSTRATION ACTIVITY



## CARDS FOR DEMONSTRATION ACTIVITY



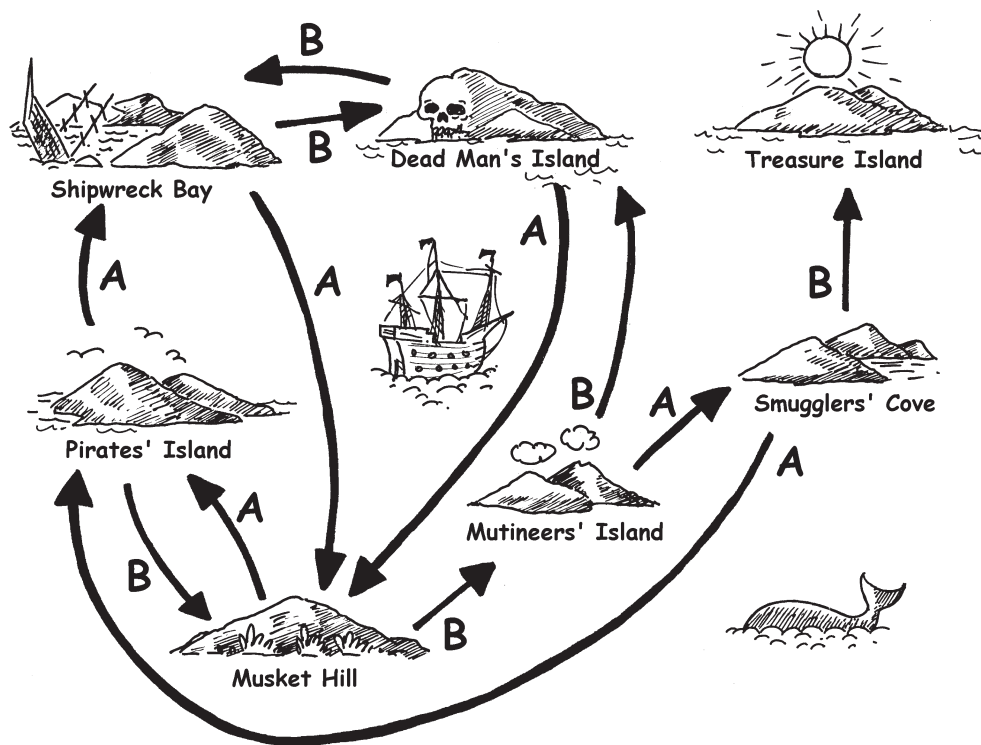
# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## ACTIVITY

Choose seven students to be “islands”. The students will hold cards identifying their island, with the secret instructions on the back. Position them randomly around the room or playground. The rest of the students are given the blank map and have to navigate a route from Pirates’ Island to Treasure Island, marking it carefully on their maps. (It is a good idea to send the students off one at a time so they cannot hear the routes in advance.)

**Fast Finishers:** Try to find more than one route.

The complete map looks like this:



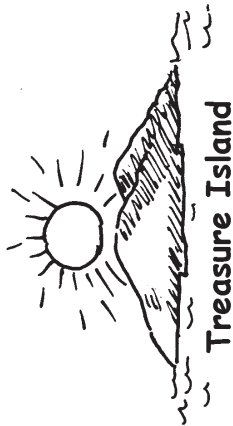
## Follow-up discussion

What is the quickest route? What would be a very slow route? Some routes may involve loops. Can you find an example of this? (For example, BBBABAB and BBBABBABAB both get to Treasure Island.)



# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## WORKSHEET ACTIVITY: *FIND YOUR WAY TO THE RICHES ON TREASURE ISLAND*



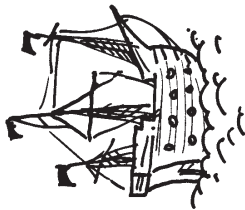
Treasure Island



Smugglers' Cove



Dead Man's Island



Mutineers' Island



Shipwreck Bay



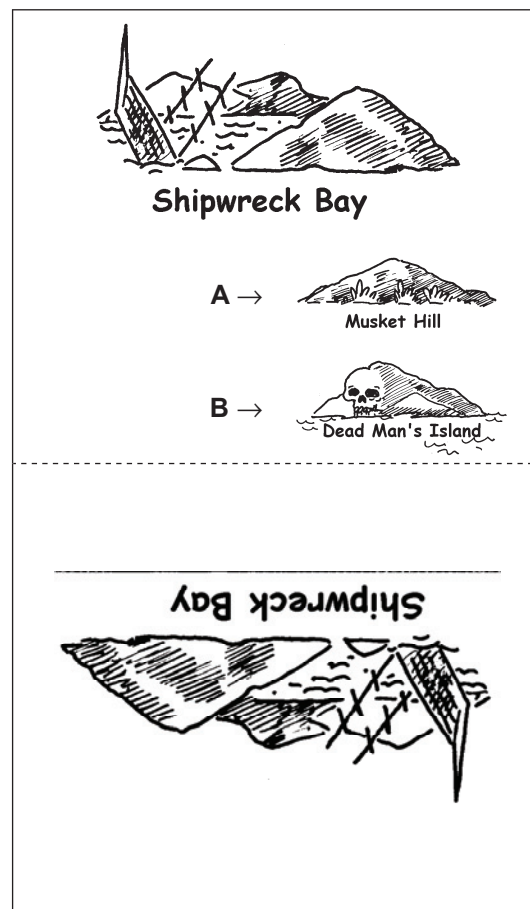
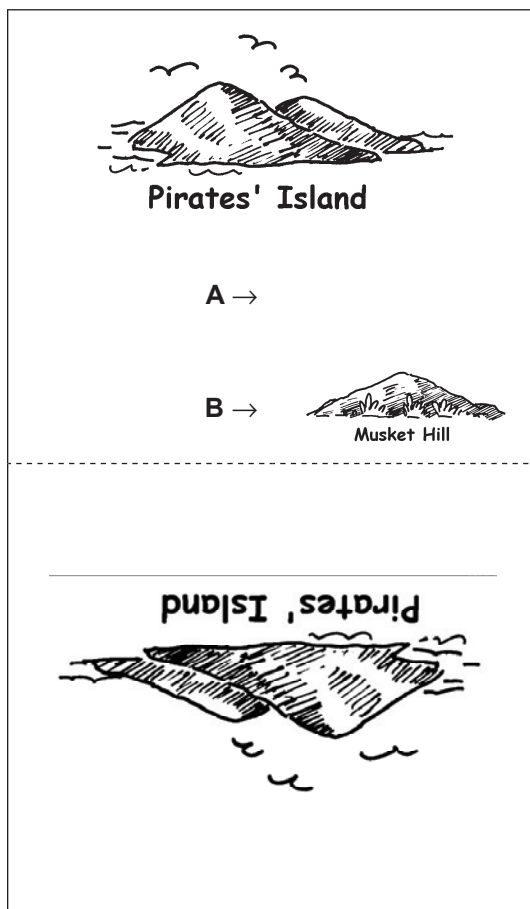
Pirates' Island



Musket Hill

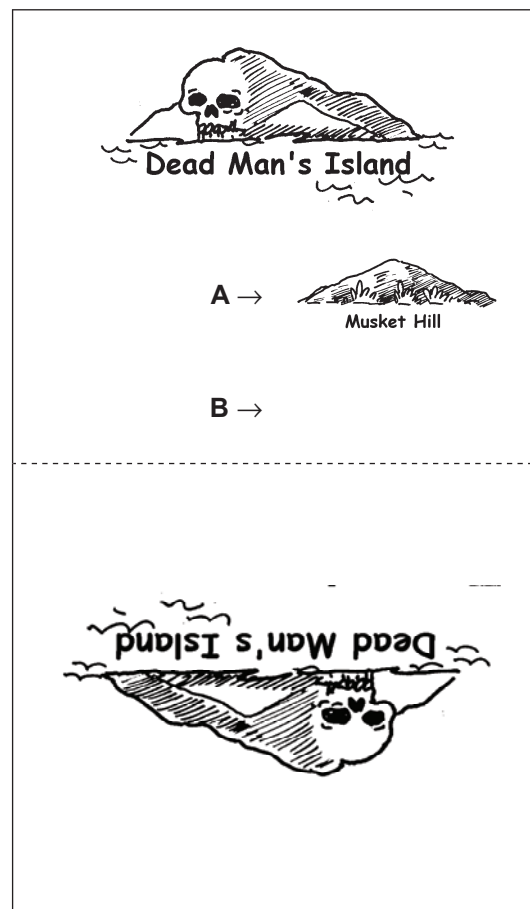
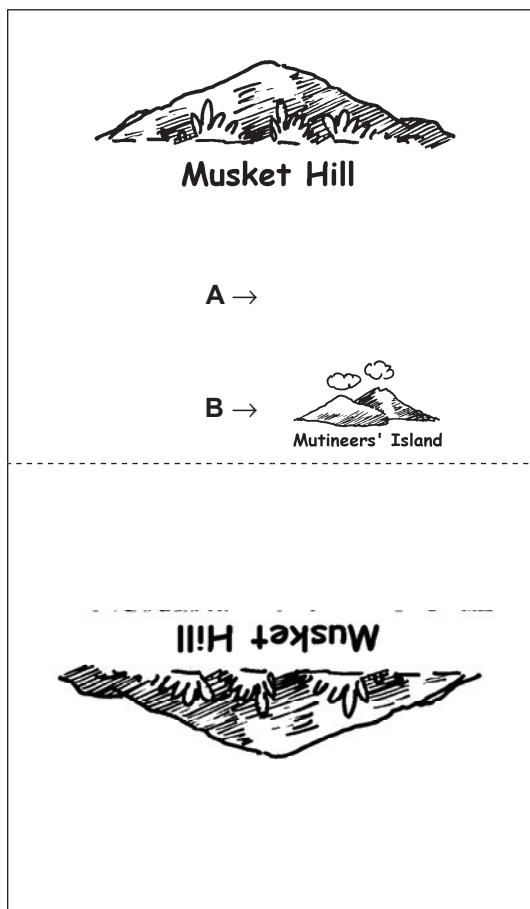
# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

PHOTOCOPY MASTER: *ISLAND CARDS* (1/4)



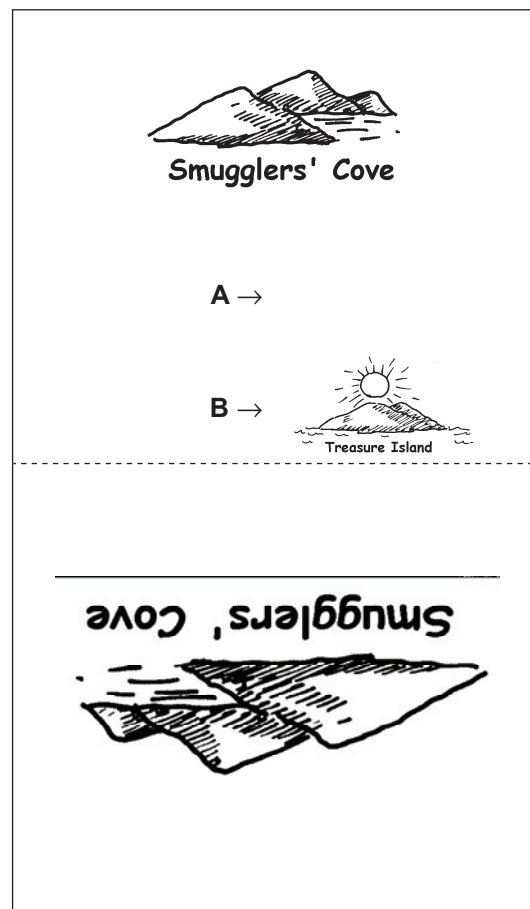
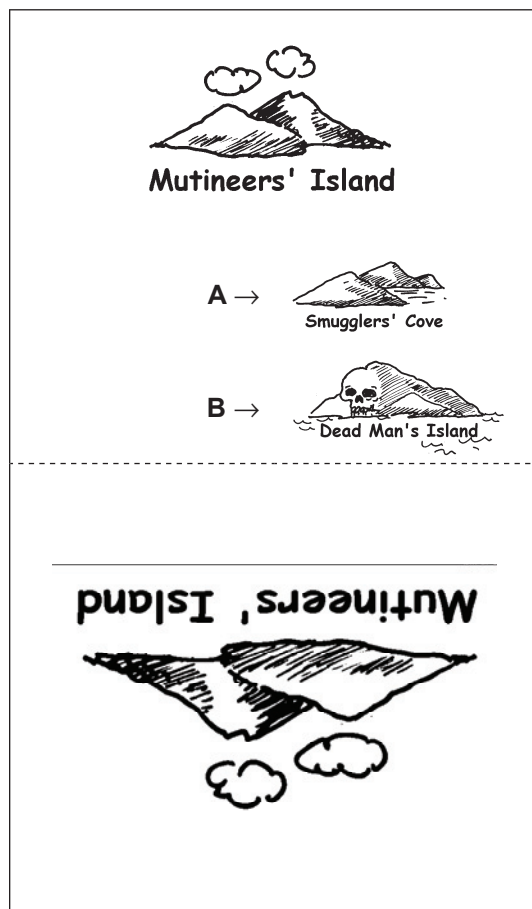
# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

PHOTOCOPY MASTER: *ISLAND CARDS (2/4)*

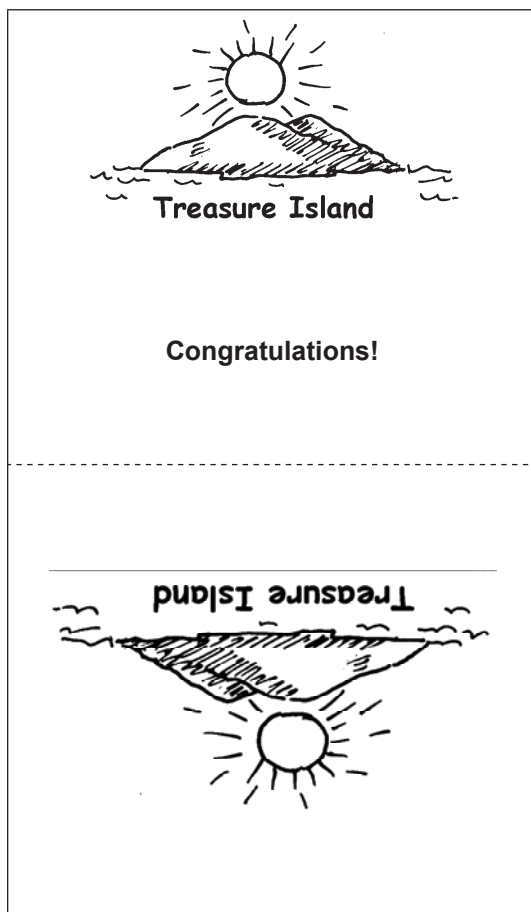


# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

PHOTOCOPY MASTER: *ISLAND CARDS (3/4)*



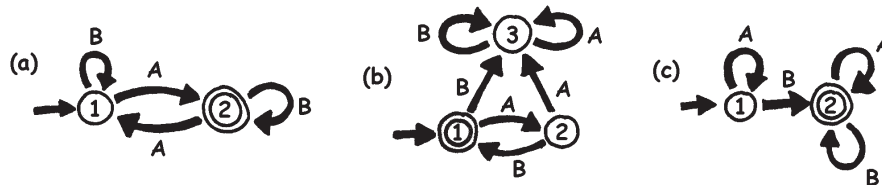
PHOTOCOPY MASTER: *ISLAND CARDS (4/4)*



# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## FINITE-STATE AUTOMATA

Another way of drawing a map is like this:



The islands are shown as numbered circles, and the final island (with the treasure) has a double circle. What routes can we travel around to get to the final island?

Note: Map (a) will finish at the double circle (island 2) only if the sequence has an odd number of As (for example, AB, BABAA, or AAABABA).

Map (b) only gets to the double circle with an alternating sequence of As and Bs (AB, ABAB, ABABAB, ...).

Map (c) requires that the sequence contains at least one B (the only sequences *not* suitable are A, AA, AAA, AAAA, ...).

# COMPUTER SCIENCE-IN-A-BOX: Unplug Your Curriculum

## WHAT'S IT ALL ABOUT? (BACKGROUND FOR THE TEACHER)

Finite-state automata are used in computer science to help a computer process a sequence of characters or events.

A simple example is when you dial up a telephone number and you get a message that says “Press 1 for this ... Press 2 for that ... Press 3 to talk to a human operator.” Your key presses are inputs for a finite state automaton at the other end of the phone. The dialogue can be quite simple, or very complex. Sometimes you are taken around in circles because there is a peculiar loop in the finite-state automaton. If this occurs, it is an error in the design of the system—and it can be extremely frustrating for the caller!

Another example is when you get cash from a bank cash machine. The program in the machine's computer leads you through a sequence of events. Inside the program all the possible sequences are kept as a finite-state automaton. Every key you press takes the automaton to another state. Some of the states have instructions for the computer on them, like “dispense \$100 of cash” or “print a statement” or “eject the cash card”.

Some computer programs really do deal with English sentences using maps like we've seen. They can both generate sentences themselves, and process sentences that the user types in. In the 1960s a computer scientist wrote a famous program called “Eliza” (after Eliza Doolittle) that had conversations with people. The program pretended to be a psychotherapist, and came out with leading questions like “Tell me about your family” and “Do go on.” Although it didn't “understand” anything, it was sufficiently plausible—and its human users were sufficiently gullible—that some people really did think they were talking to a human psychotherapist.

Although computers are not really very good at understanding natural language, they can readily process artificial languages. One important type of artificial language is the programming language. Computers use finite-state automata to read in programs and translate them into the form of elementary computer instructions, which can then be “executed” directly by the computer.

