Digital Technologies @ Home
Unplugged activities for students

SECONDARY STUDENTS

Logic Gates (7-8)
Maze Escape (7-8)
Unscrambling a secret message (7-8)
Indoor Scavenger Hunt (7-8, 9-10)
Spaceship Rescue (7-8, 9-10)
Cellular Automoji (7-8, 9-10)
Card Switches (7-8, 9-10)
Hamiltonian and Eulerian Paths and Circuits (7-8, 9-10)
Convenient Stores (7-8, 9-10, 11, 12)
TV Torment (7-8, 9-10, 11, 12)
LAN Party (9-10)

Digital Technologies @ Home
Unplugged activities for students

This activity is for: Years 7-8

Logic gates

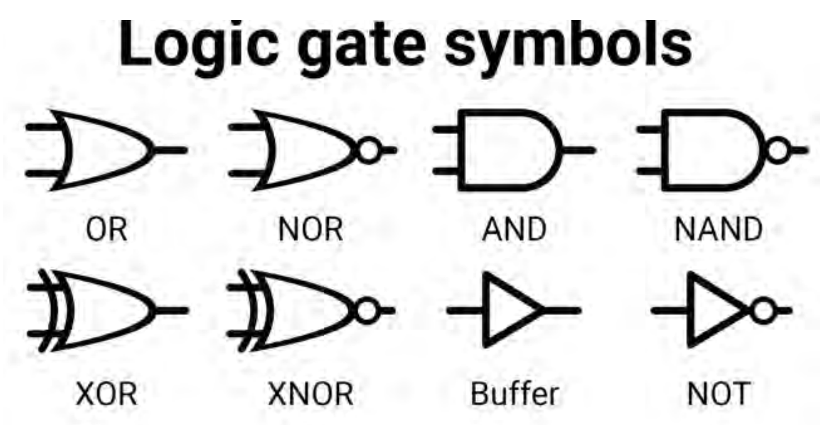
Trace logic gates following simple rules, to check if the light is on or off!

This activity teaches...

Logic gates are a representation of electronic circuit devices used in computers. Their inputs are binary, either on or off (or 1 or 0), and so is their output. Different logic gates have different rules that define how they work.

In this activity we'll learn the rules of logic gates, add logic gates together, and determine what their output will be based on their inputs.

This activity is aimed at lower secondary students and will take about 45 minutes to complete.



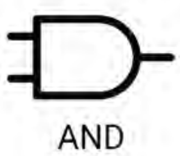
Getting started (read this with your child):

Logic gates have inputs, and outputs with just two states. On, and off (or 1 and 0).

If an input is on, we say it has a value of 1.
If an input is off, we say it has a value of 0.

Here are some gates!

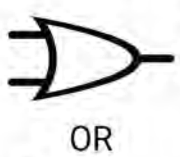
This is an AND gate. It's output will be 1 if the top **and** bottom inputs are 1.



We can also represent this with a truth table:

A	B	Out
0	0	0
0	1	0
1	0	0
1	1	1

This is an OR gate. It's output will be 1 if either the top **or** bottom input are 1.



We can also represent this with a truth table:

A	B	Out
0	0	0
0	1	1
1	0	1
1	1	1

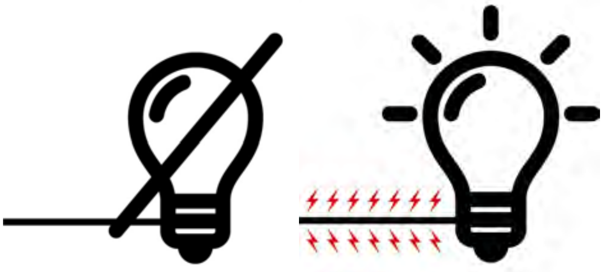
Let's try some activities with these logic gates.

Logic Gates

Solve these questions! Will the light be on or off?

For these questions, we'll ask if the light is on or off.

The light is on if there is power getting to it through the wire, and it's off if there isn't.

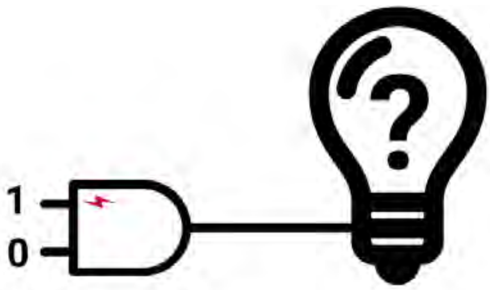


Off On!

The tricky bit is, we're going to send the power through the logic gates before it gets to the light.

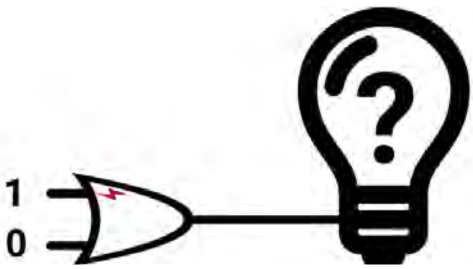
Question 1

Will this light be on or off?



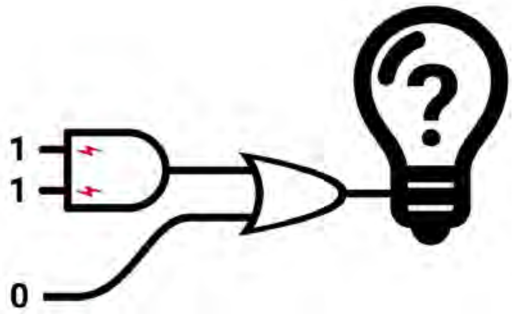
Question 2

Will the light be on or off?



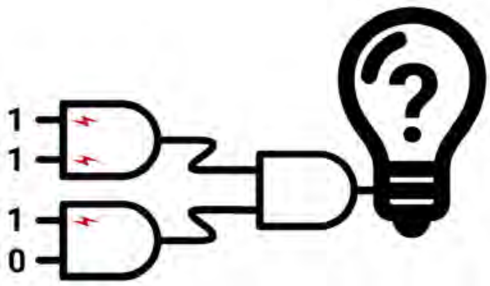
Question 3

Will the light be on or off?



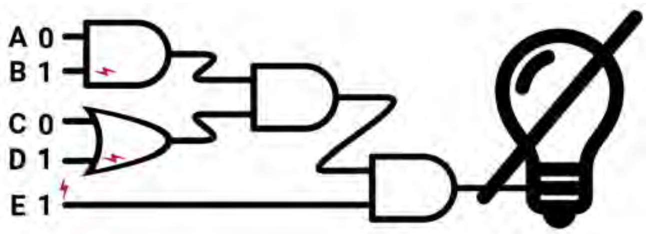
Question 4

Will the light be on or off?



Question 5

This light is off. Which switch do we need to flip to turn it on? A, or C?



New Logic Gates!

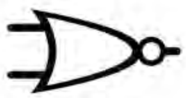
This is called a NAND gate. It's output will be 1 if both the top and bottom inputs are **not** 1. You can think of it as an AND gate, with the output reversed. That's why we call it a NAND gate, **N**ot **AND** gate.

NAND

We can also represent this with a truth table:

A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

This is an NOR gate. It's output will be 1 if neither the top **or** bottom input are 1.



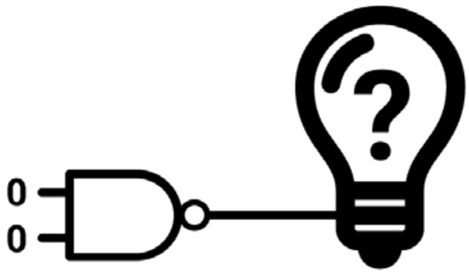
NOR

We can also represent this with a truth table:

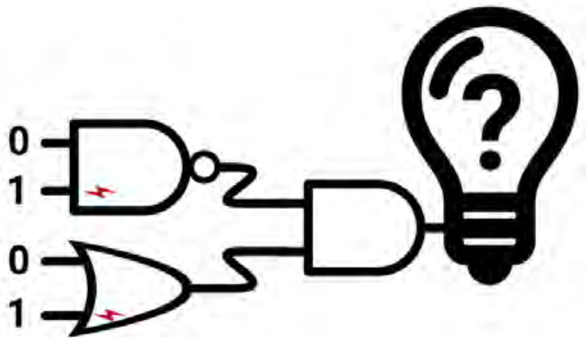
A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

Let's try some problems with these new gates.

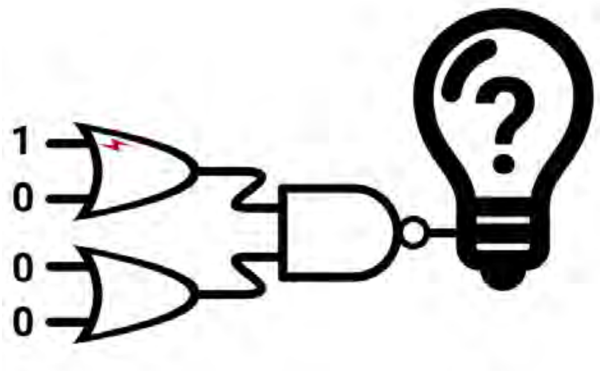
Question 6
Will the light be on or off?



Question 7
Will the light be on or off?



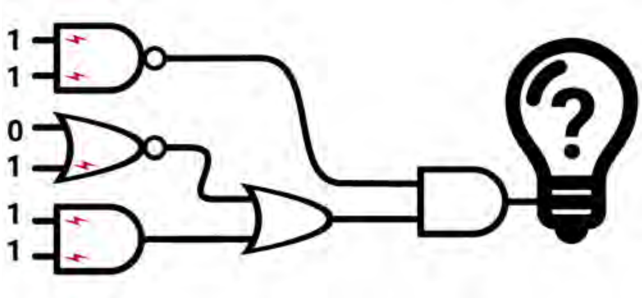
Question 8
Will the light be on or off?



What's the minimum number of switches that can be turned on for this light to be on? _____

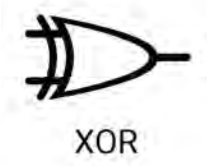
The maximum? _____

Question 9
Will the light be on or off?



XOR and XNOR gates

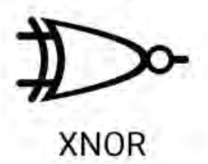
This is an exclusive OR gate, or XOR gate. It's output will be 1 if either the top **or** bottom inputs are 1, but not both.



We can also represent this with a truth table:

A	**B**	**Out**
0	0	0
0	1	1
1	0	1
1	1	0

This is an XNOR gate, or exclusive not or. It's output will be 1 if the output of an XOR gate would be 0.



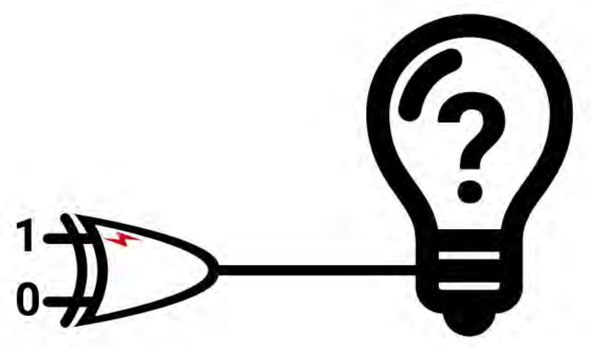
We can also represent this with a truth table:

A	**B**	**Out**
0	0	1
0	1	0
1	0	0
1	1	1

Let's try some problems with these new gates.

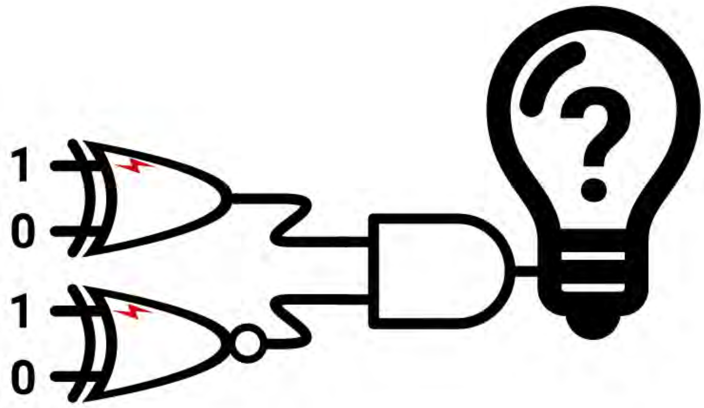
Question 10

Will the light be on or off?



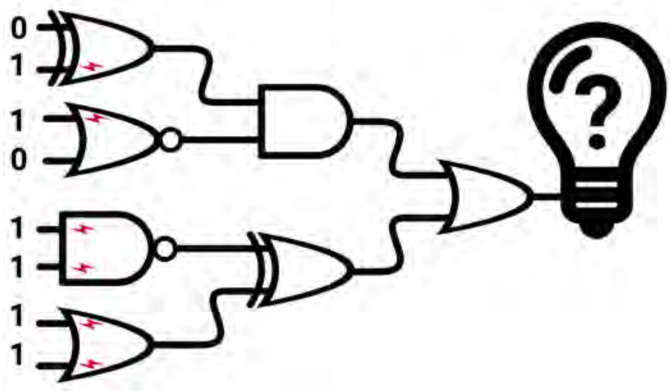
Question 11

Will the light be on or off?



Question 12

Will the light be on or off?



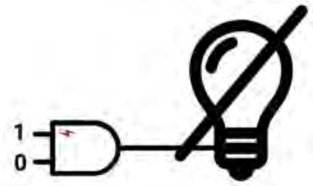
This page is for

Parents and carers

Answer key

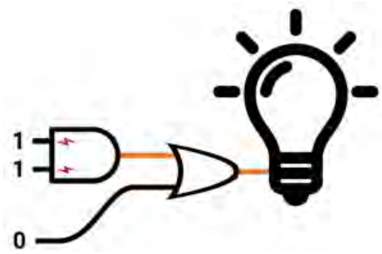
Choose if you want to print this for your kids or keep it to yourself!

1: Off

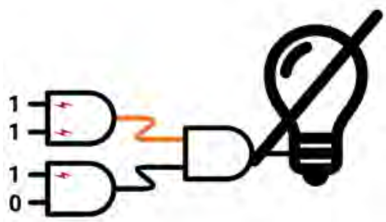


2: On

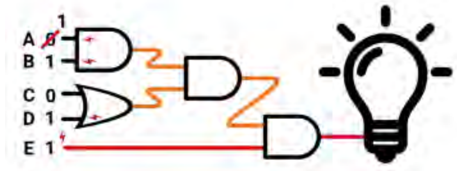
3: On



4: Off



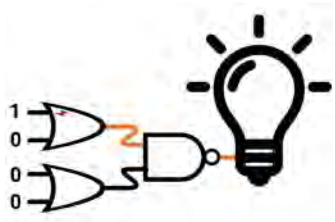
5: Flip A to turn it on



6: On

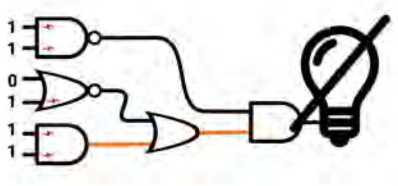
7: On

8: On



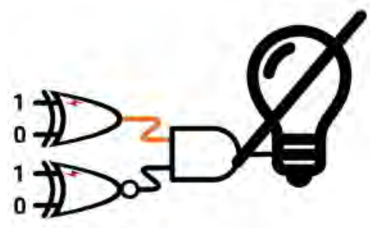
Minimum on switches: 0
Maximum on switches: 2

9: Off

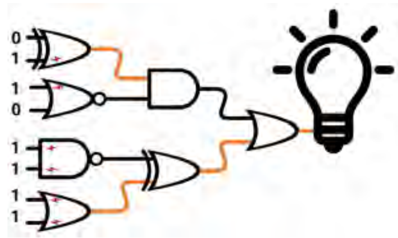


10: On

11: Off



12: On



Want more?

Here are some further activities, online resources, assessment ideas and curriculum references.

Adapting this activity

Older students can go on to learn about DeMorgan's theorem, which lets you transform a set of logic gates into equivalent gates, and boolean algebra, which is used in computing.

Keep the conversation going

- How does the number of logic gates used relate to the number of inputs?
- Can you think of a way to build a physical logic gate? Think water, or marbles. Some people have even made them using lego!

Keep learning

For High School students interested in learning about how computers communicate with encrypted messages, try this course: cmp.ac/crypto

For teachers creating a portfolio of learning or considering this task for assessment

Students can draw a truth table for a circuit of logic gates, rather than just one.

Is there any way to lessen the number of gates needed while getting the same result?

Linking it back to the Australian Curriculum: Digital Technologies

Algorithms
Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition) (ACTDIP019 - see cmp.ac/algorithms)

Refer to aca.edu.au/curriculum for more curriculum information.

This page is for

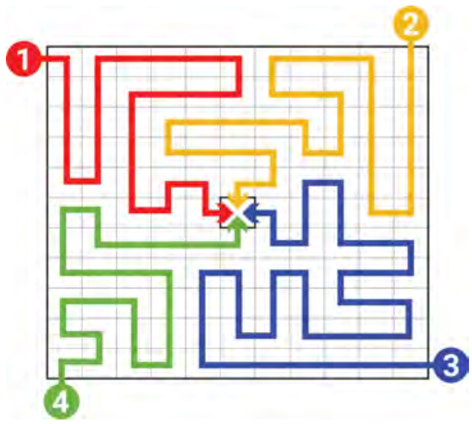
Teachers

Parents and carers

Digital Technologies @ Home
Unplugged activities for students

This activity is for: Years 5-8

Maze escape!



Many thanks to the Girls Programming Network who created this activity.

This activity teaches…

In this activity students create an algorithm, or set of instructions, to navigate through a maze. To create the algorithm, students choose from a finite set of commands. We want students to create the **best** set of instructions to get through the maze - in this activity best means the fewest number of instructions.

Creating good sets of instructions is a very important concept in computing. Solving problems takes a computer's time and energy: as we ask computers to solve bigger and bigger problems (like climate modelling and searching for life beyond earth) it's important to find the most efficient solution available.

The commands students can use to get through the maze include two key programming ideas: **branching**: ie IF something is true THEN do something) and **iteration:** keep doing something a fixed number of times or until a specified condition is no longer true.

This activity is targeted towards students in years **5 to 8** and will take **1 to 2 hours**.

Getting started (read this with your student):

We're going to write a program to move a character to the center of the maze! Use the printed commands to create an algorithm, or set of instructions, to get to the centre of the maze, following one of the lines. Your goal is to use as few commands as possible.

You can do this activity in a few ways: you can use a printed copy of the maze, or you can use masking tape (if you have space) on the floor, creating a room-sized maze following the picture shown. You can either do this with a partner, taking turns coming up with instructions and then following them, or you can do it alone and just use a coin or other item on the print out to follow the instructions, making sure you follow them exactly.

You can print off and cut out all the commands for this activity, or you can write instructions out on a piece of paper.

See a demonstration

cmp.ac/mazevid

Maze escape!

Can you get to the centre of the maze with the fewest instructions?

Step 1

Either use the maze on the next page for this activity, or if you have space, you can create this maze on your floor using masking tape.

Step 2

Cut out the commands on pages 4-9.

Step 3

Using only the printed commands, create an algorithm to get you, a partner or a counter to the centre of the maze, following one of the lines.

Step 4

With a partner, or alone, follow your Step 3 instructions exactly. Do they work? Do you need to change them a bit? Can you improve them at all?

Step 5

If your first set of instructions gets you to the middle, work out how many points you earn (see step 6), then make a new set of instructions for the next path. Keep going until you have done all 4 paths.

Step 6

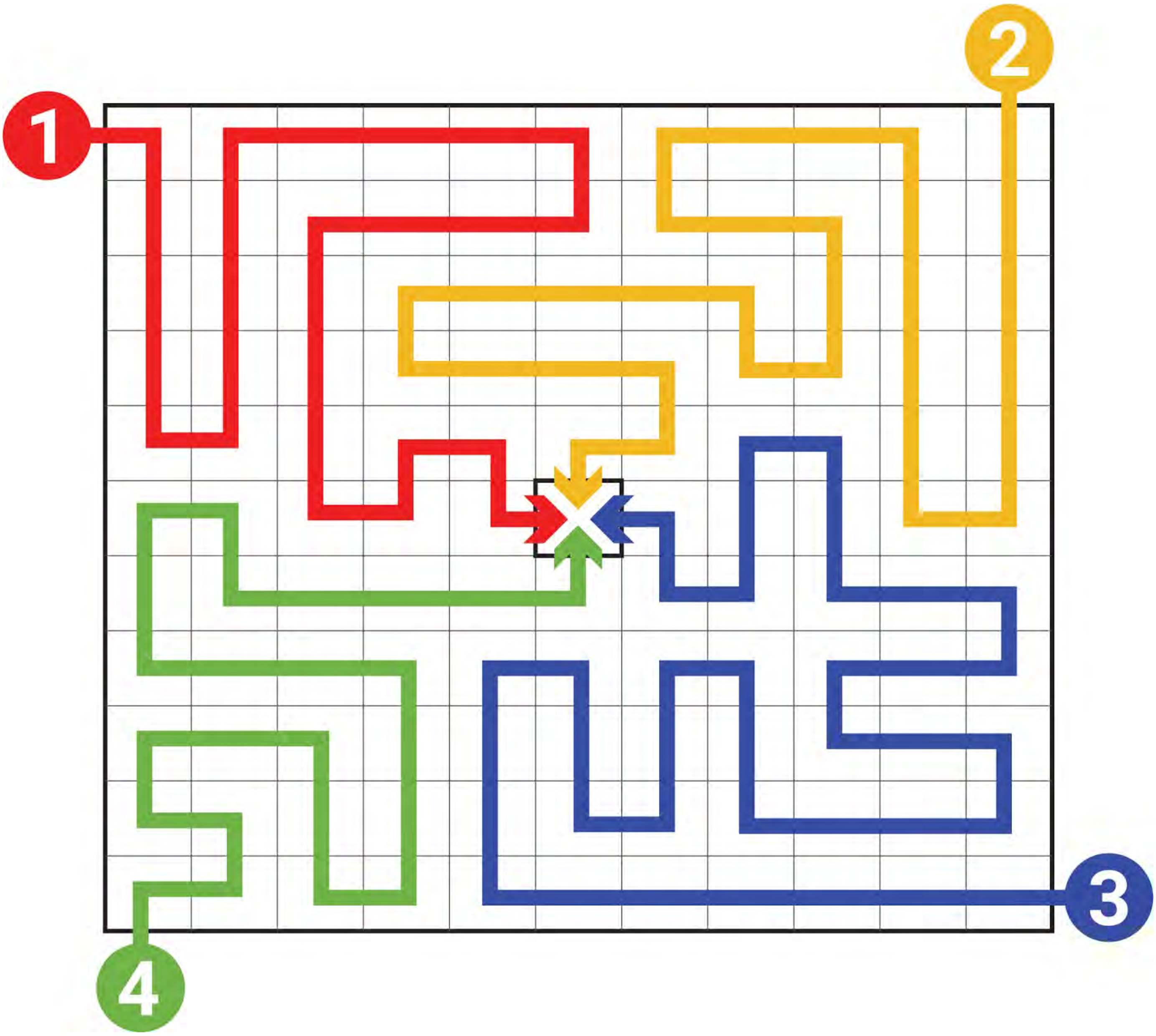
You earn points for each set of instructions. There are 4 paths to the centre of the maze. For each path, the less commands you use the more points you get!

- You get 10 points per path if your instructions have less than 12 commands.
- You get 8 points per path if your instructions have between 12 and 20 commands.
- You get 5 points per path if your code has more than 20 commands.

If you write a program that works on any of the paths and will get you (or your partner) to the centre, you get a **bonus 10 points**. Get **another bonus 10 points** if your instructions solve all the puzzles in **less than 7 lines**.

Maze escape!

Can you get to the centre of the maze with the fewest instructions?



Maze escape!

Can you get to the centre of the maze with the fewest instructions?

Print off these commands

For ____ counts: →
For ____ counts: →
For ____ counts: →
For ____ counts: →

Maze escape!

Can you get to the centre of the maze with the fewest instructions?

Code snippets - while loop

While not at the end of maze: →
While not at the end of maze: →
While not at the end of maze: →
While not at the end of maze: →

Maze escape!

Can you get to the centre of the maze with the fewest instructions?

Code snippets - while loop

While there is path ahead: →
While there is path ahead: →
While there is path ahead: →
While there is path ahead: →

This page is for

Students

Maze escape!

Can you get to the centre of the maze with the fewest instructions?

Code snippets - if

If there is a path to the left: →
If there is a path to the left: →
If there is a path to the right: →
If there is a path to the right: →

Maze escape!

Can you get to the centre of the maze with the fewest instructions?

Code snippets

If there is a path in front: →
If there is a path in front: →
If there is a path in front: →
If there is a path in front: →

Maze escape!

Can you get to the centre of the maze with the fewest instructions?

Turn to the right	Turn to the left
Turn to the right	Turn to the left
Turn to the right	Turn to the left
Turn to the right	Turn to the left
Turn to the right	Turn to the left
Step forward	Step forward
Step forward	Step forward
Step forward	Step forward
Step forward	Step forward

Answer key

Choose if you want to print this for your kids or keep it to yourself!

There are many possible solutions to this problem. The most efficient solution uses **6 commands** that solves all mazes:

```
While not at the end of maze:
→ If there is a path to the left:
    → Turn to the left
    If there is a path to the right:
    → Turn to the right
    Step forward
```

The reason we don't need the "If there is a path in front" before stepping forward - is because we've checked if the line turns already, it will *always* be forward.

Here is an example to follow line 1 that would only earn 5 points:

```
Step forward
Turn to the right
For 4 counts:
→ Step forward
Turn to the left
Step forward
Turn to the left
For 4 counts:
→ Step forward
Turn to the right
For 4 counts:
→ Step forward
Turn to the right
Step forward
Turn to the right
```

```
(continued):
For 3 counts:
→ Step forward
Turn to the left
For 4 counts:
→ Step forward
Turn to the left
Step forward
Turn to the left
Step forward
Turn to the right
Step forward
Turn to the right
Step forward
Turn to the left
Step forward
```

There are many possible solutions! The best way is to run through someone's program to see if it gets you to the end!

Want more?

Here are some further activities, online resources, assessment ideas and curriculum references.

Adapting this activity

Once students understand how to complete this activity, ask them to perform similar steps using a map - or the steps to get to their bedroom.

Keep the conversation going

- What were some strategies that you used to solve the problem?
- How does it relate to other types of instructions like directions or navigation?
- Did you see any repeated patterns within the solution? In programming we have functions that could repeat a certain pattern - even if they don't come directly after each other.

Keep learning

To move this type of thinking into the computer, students can write their own programs that follow these types of steps! We recommend trying the Blockly Turtle course:
cmp.ac/blocklyturtle

For younger students who would like to create a maze using the Scratch programming language, there is a printable step by step guide available here:
cmp.ac/scratchmaze

For teachers creating a portfolio of learning or considering this task for assessment

Ask students to submit their best algorithm for the 4 paths in this worksheet.
You could also ask them to design their own maze and best algorithm.

Linking it back to the Australian Curriculum: Digital Technologies

Algorithms
Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition) (ACTDIP019 - see cmp.ac/algorithms)

Refer to aca.edu.au/curriculum for more curriculum information.

Digital Technologies @ Home
Unplugged activities for students

This activity is for: Years 5-8

Unscrambling a secret message

This activity teaches…

We can make a message secret by changing the letters to create an *encrypted message*. This change process is called a *cipher*. Anyone who knows the cipher can reverse it, *decrypting* the secret message to get the original back. The message is still there, but hidden, because we've changed how we *represent* it.

Simple ciphers, such as Pig Latin[1] or Caesar Cipher[2], can be *cracked* to find the original message and the cipher.

We use *secure* ciphers (that are extremely hard to crack) to protect communication on the Internet, e.g. to stop hackers getting our credit card details when we shop online. Without encryption, every message we send is at risk.

In this activity, the message is encrypted by swapping letters (a *substitution cipher*). There is no pattern to how they are swapped, except that each letter always swaps to the same one. Here, the cipher encrypts every *G* by swapping it to an *A*, so to decrypt the message, we must swap every *A* back to an *G*.

You can crack the substitution cipher to find the original message by looking for familiar words and letter patterns in the encrypted message. Good luck!

This activity will take up to **60 minutes** . Print pages **2** for students. The answer appears on page **3**. If you are a teacher, read through page **4** for further information.

Getting started (read this with your child):

We're going to unscramble a hidden message, using our understanding of English and commonly used words, and learn about one way to send hidden messages. We're going to crack a cipher!

[1] https://en.wikipedia.org/wiki/Pig_Latin
[2] https://cryptii.com/pipes/caesar-cipher

THE UNIVERSITY OF SYDNEY | Australian Computing Academy | Girls' Programming Network | GROK LEARNING

This page is for Students

Can you unscramble the message?

The letters have been jumbled up. Use the grid at the bottom to write down your answer for each letter. Use your knowledge of words to figure out what the message says.

The first word has only one letter. How many one letter words do you know? It can't be **A** because the answer grid shows us that when you see a **G** in the scrambled message, this becomes **A** in the unscrambled message (the top line). So it must be **I**. Go ahead and find all the other letters **Y** in the scrambled message and write **I** above them. Also write **I** above **Y** in the answer grid at the bottom of the page.

Look for other parts of words that you recognize. On the third line the letters **BCGJ** are unscrambled to be _**HAT.** Think of a word that ends in **HAT.** It might be **THAT,** but we already know that **J** in the scrambled message becomes **T,** and this scrambled word starts with **B.** Once you have figured out what to change letter **B** to, write it above **B** everywhere you see it in the message and also in the answer grid, then look for other parts of words you recognise.

Unscramble this message

							T				A	T		M	Y
Y		Z	Y	P	W		J	H		W	G	J		A	L

				T	B						T	H	
	B	W	W	J	R	Y	N		B	Y	J	C	

	H					.			H	A	T			
E	C	W	W	V	W	.		B	C	G	J		Y	V

T	H			B			T		B			A		-
J	C	W		R	W	V	J		R	D	W	G	P	-

	A		T								U	?	
I	G	V	J		I	H	D		L	H	M	?	

Write your answers here

M		H				A			T			U
A	B	C	D	E	F	G	H	I	J	K	L	M

				B								
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

This page is for

Teachers

Parents and carers

Answer key

Print this for yourself to check your child's answers.

The unscrambled message

I		L	I	K	E		T	O		E	A	T		M	Y
Y		Z	Y	P	W		J	H		W	G	J		A	L

	W	E	E	T	B	I	X		W	I	T	H	
	B	W	W	J	R	Y	N		B	Y	J	C	

C	H	E	E	S	E	.		W	H	A	T		I	S
E	C	W	W	V	W	.		B	C	G	J		Y	V

T	H	E		B	E	S	T		B	R	E	A	K	-
J	C	W		R	W	V	J		R	D	W	G	P	-

F	A	S	T		F	O	R		Y	O	U	?	
I	G	V	J		I	H	D		L	H	M	?	

The completed answer key

M	W	H	R	C	J	A	N	F	T	P	Y	U
A	B	C	D	E	F	G	H	I	J	K	L	M

X	Q	K	D	B	Z	O	G	S	E	V	I	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

More information

Here are some further activities, online resources, assessment ideas and curriculum references.

Teachers

Adapting this activity

Once students understand how to complete this activity, ask them to prepare a hidden message for a family member, or in an online class, for a classmate. It helps to have short words in the message if you want to crack them.

Keep learning

For year 7 to 10 students interested in learning more about how computers communicate with encrypted messages, try this course:
cmp.ac/crypto

An additional hands-on lesson plan further exploring cryptography is available to download at
cmp.ac/cipherwheels

For teachers creating a portfolio of learning or considering this task for assessment

Ask students to create their own scrambled message using a substitution cipher that they create.

Students could also explore when cryptography is necessary for securing data by making a list of data they frequently send online: examples include messages, searches eg, for netflix, bus timetables, recipes, banking transactions, purchase of movie tickets, submitting school work. Students can then sort these transmissions into activities where encryption is (i) necessary (ii) a good idea (iii) unnecessary.

Linking it back to the Australian Curriculum: Digital Technologies

Digital systems
Investigate how data is transmitted and secured in wired, wireless and mobile networks, and how the specifications affect performance (ACTDIK023 - see cmp.ac/systems)

Cryptography is an important part of investigating the transmission of data.

Cryptography

Cryptography allows a message to be securely stored and transmitted.

Students explain why cryptography is necessary for securing data (e.g. transmitting credit card details over the web) and explore simple encryption and decryption algorithms (e.g. rot13 and XOR).

Refer to aca.edu.au/curriculum for more curriculum information.

Digital Technologies @ Home
Unplugged activities for students

This activity is for: Years 7-10

Indoor Scavenger Hunt

With thanks to David Schulz, year 10 student, for providing this content.

This activity teaches…

In this activity, students solve seven puzzles to navigate their way around the house to find a hidden treasure. Puzzles are solved by applying a range of data representation, binary number, pixel graphic, morse code and simple cryptography skills.

It is targeted towards secondary students and is expected to take 15 to 45 minutes.

Care has been taken to choose rooms that are representative of typical Australian households, However, if your settings are different, we ask parents and carers to be creative. For example, if you don't have a study, then designate an area in your house or unit (for example a wardrobe) and call it study.

Encryption and data representation are important ideas in computing. Encryption allows computers to communicate securely with one another. We use secure ciphers (which are much harder to crack than the ones in this activity) to protect communication on the Internet, e.g. to stop hackers getting our credit card details when we shop online. Without encryption, every message we send is at risk.

Many ideas can be communicated using symbols and conventions. In this activity, dots, dashes, pixel graphics and strings of 1s and 0s are all used to communicate important information. Computers also use conventions to store different kinds of data, including using binary numbers.

You will need…

A treasure. Easter eggs, a toy, or any other appropriate prize will do nicely.

Hide the treasure in the child's / children's bedroom.

Cut out the puzzles on pages 3 and 4 and hide them in rooms as follows.

Puzzle 1	Hand it to the child / children
Puzzle 2	Kitchen
Puzzle 3	Study
Puzzle 4	Parent's Bedroom
Puzzle 5	Living Room
Puzzle 6	Bathroom
Puzzle 7	Dining Room

Indoor Scavenger Hunt

Getting started (read this with your child):

A treasure is hidden somewhere in the house/unit. To solve this puzzle, you must decode the provided clues by correctly decoding/translating them. Each solution will lead you to the next room. Sometimes, the solution to a problem is a number, which maps to a room as shown in the following table.

1	Parent's Bedroom
2	Bathroom
3	Kitchen
4	Dining Room
5	Children's Bedroom
6	Living Room
7	Study

When numbers are provided in a puzzle, translate them to their corresponding letter. For example, 123 = ABC (Number decoding).

Remember! You **must** show your working, and the rooms which you travel through to successfully complete the challenge.

This page is for

Students

Indoor Scavenger Hunt

A treasure is hidden in the house. Solve the puzzles to find it.

Puzzle 1

Here is your first secret message:

dash-dot-dash
dot-dot
dash
dash-dot-dash-dot
dot-dot-dot-dot
dot
dash-dot

Solve it and go to the stated room.

A ●▬	J ●▬▬▬	S ●●●
B ▬●●●	K ▬●▬	T ▬
C ▬●▬●	L ●▬●●	U ●●▬
D ▬●●	M ▬▬	V ●●●▬
E ●	N ▬●	W ●▬▬
F ●●▬●	O ▬▬▬	X ▬●●▬
G ▬▬●	P ●▬▬●	Y ▬●▬▬
H ●●●●	Q ▬▬●▬	Z ▬▬●●
I ●●	R ●▬●	

Puzzle 2

These letters seem to make no sense at all. It must be a code. You figure it out and make your way to the answer.

Brxu remhfwlyh olhv lq wkh vwxgb...

Tip: A Caesar Cipher is a way of encrypting a message. To encrypt a message, letters are shifted to the right by a shift key. For example if the shift key is 1, the message ABC becomes BCD. To decrypt BCD we shift each letter one to the left, and it becomes ABC.

The shift key for this message is the **number** of the room you are in.

Puzzle 3

Here is another message for you: **Your forerunners nod off hereabouts**. You realise that your parents scrambled the message and changed the words around, but only so that they seem different - the words used are all synonyms for our normal name for the next location.

Puzzle 4

Here is the fourth puzzle for you:

20-8-9-19
15-14-5 9-19
6-1-9-18-12-25
5-1-19-25
25-15-21
23-1-20-3-8
20-22
8-5-18-5

Each of the groups of numbers is a word. Solve the puzzle to continue your journey.

Puzzle 5

A simple riddle is told, and you listen intently: "Kids love going here for a swim, but this isn't the pool. Dogs also love it, because they can sometimes drink out of something else in this room." Go to that room.

Puzzle 6

What will you do with this puzzle?
010101010 – 010100110 = ? As the equation becomes clear, you hurry to the room, but only after you've quickly taken a sip of water.
Tip: If you need an extra hint to solve this, ask for a clue.

Puzzle 7

Make sense of this cryptic sequence of 1's and 0's: **11111100001000011111000010000111111**.

Use this 5 x 7 table to solve this puzzle. Tip: If you need an extra hint to solve this, ask for a clue.

Puzzle 8

This is the final room. The treasure is here somewhere.

Answer key

Choose if you want to print this for your kids or keep it to yourself!

Answers

Puzzle 1
KITCHEN

Puzzle 2
The shift is 3, because the riddle is being solved in the kitchen.

By shifting the letters in **Brxu remhfwlyh olhv lq wkh vwxgb** by three to the left (remember, we are uncoding), the message reads : **Your objective lies in the study**

Puzzle 3
Your parents sleep here = Parent's bedroom

Puzzle 4
Each number gets converted to its corresponding letter in the alphabet. For example, 123 = ABC. The resulting message is: **this one is fairly easy, you watch tv here** = Living Room

Puzzle 5
Bathroom

Puzzle 6
The two strings of 1's and 0's represent binary numbers:

Clue: If the child is stuck, point out that these are binary numbers.

010101010 = 170
010100110 = 166

170 - 166 = 4. The solution is the Dining Room

Alternatively, observe that the first five bits of both numbers are identical and result to 0 in a subtraction. You can therefore simplify both numbers to 1010 - 0110 = ten - six = four.

This page is for

Teachers

Parents and carers

Answer key

Choose if you want to print this for your kids or keep it to yourself!

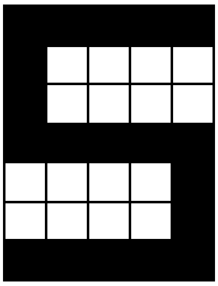
Puzzle 7

Pixel graphics.

Clue: If the child is stuck, suggest to Split the binary number into seven groups of five bits

11111
10000
10000
11111
00001
00001
11111

Enter them into the 5*7 matrix. A 1 corresponds to a black pixel. The resulting number is 5, which is the child's / children's Bedroom.



Want more?

Here are some further activities, online resources, assessment ideas and curriculum references.

Adapting this activity

This activity can be a springboard for students to create their own clues for siblings or parents to continue the scavenger hunt. It could also be used for a collaborative class activity where students can create clues to send a secret message around the class, translating it from a cipher to morse code, number translation, etc.

Older students may enjoy exploring more complex methods of encryption such as hashing algorithms: https://brilliant.org/wiki/secure-hashing-algorithms/ (which wouldn't work well in this activity as they can't be decrypted.)

Keep learning

For High School students interested in learning more about how computers communicate with encrypted messages, try this course: cmp.ac/crypto.

For students who would like to explore data representation, and how computers store images, text and music using 1s and 0s, try this course: cmp.ac/pythondatarep

For teachers creating a portfolio of learning or considering this task for assessment

Ask students to create their own scavenger hunt clues based on the activities in this worksheet.

Linking it back to the Australian Curriculum: Digital Technologies

Data representation
Investigate how digital systems represent text, image and audio data in binary (ACTDIK024 - see cmp.ac/datarep)

Refer to aca.edu.au/curriculum for more curriculum information.

Digital Technologies @ Home
Unplugged activities for students

This activity is for: Years 7-10

Spaceship Rescue

This activity teaches algorithms

We use algorithms to solve all sorts of problems around us. Algorithms are sequences of steps, or procedures, that lead us from a starting position to a goal. Some algorithms can be described easily (think about the recipe for making a cake), whilst others are harder to describe (think about a Sudoku puzzle).

The algorithm in this activity is somewhere between a cake recipe and a Sudoku puzzle. Two parts of it are purely procedural, whilst some other parts require trial and error, heuristics, or gut feeling. Rest assured, it is completely based on the laws and logic and can be described as a computer algorithm, just like a Sudoku puzzle can be solved by an algorithm. Students interested in taking this activity further can try to create a computer game that implements the ideas in this activity.

This activity will take up to **60 minutes**. If you are a teacher, read through page 7 for further information. Instructions for students are on pages 4 and 5.

You will need…

- Printouts of the spaceship grid on page 2 and the markers on page 3. (If you don't want to cut out markers, you can use a pencil to record where you think a spaceship may or may not be).

Getting started (read this with your child):

Attention Space Commanders! NASA has lost four spaceships in deep space. Probes have been deployed to gather data. A probe can see any number of spaceships along its line of sight. The maximum number it can see in this situation is four, since there are four spaceships. Your job is to triangulate the data to find the missing spaceships and save the astronauts. Are you up for the challenge?

Clarifying terms:

In trigonometry and geometry, **triangulation** is the process of determining the location of a point by forming triangles to it from known points.[1]

See a demonstration

cmp.ac/spaceshipvid

[1] Wikipedia, Triangulation, https://en.wikipedia.org/wiki/Triangulation, last accessed: 31.3.2020

Spaceship Rescue

NASA has lost 4 spaceships in deep space ... Probes have been deployed to gather data … NASA needs your help to find the spaceships and save the astronauts.

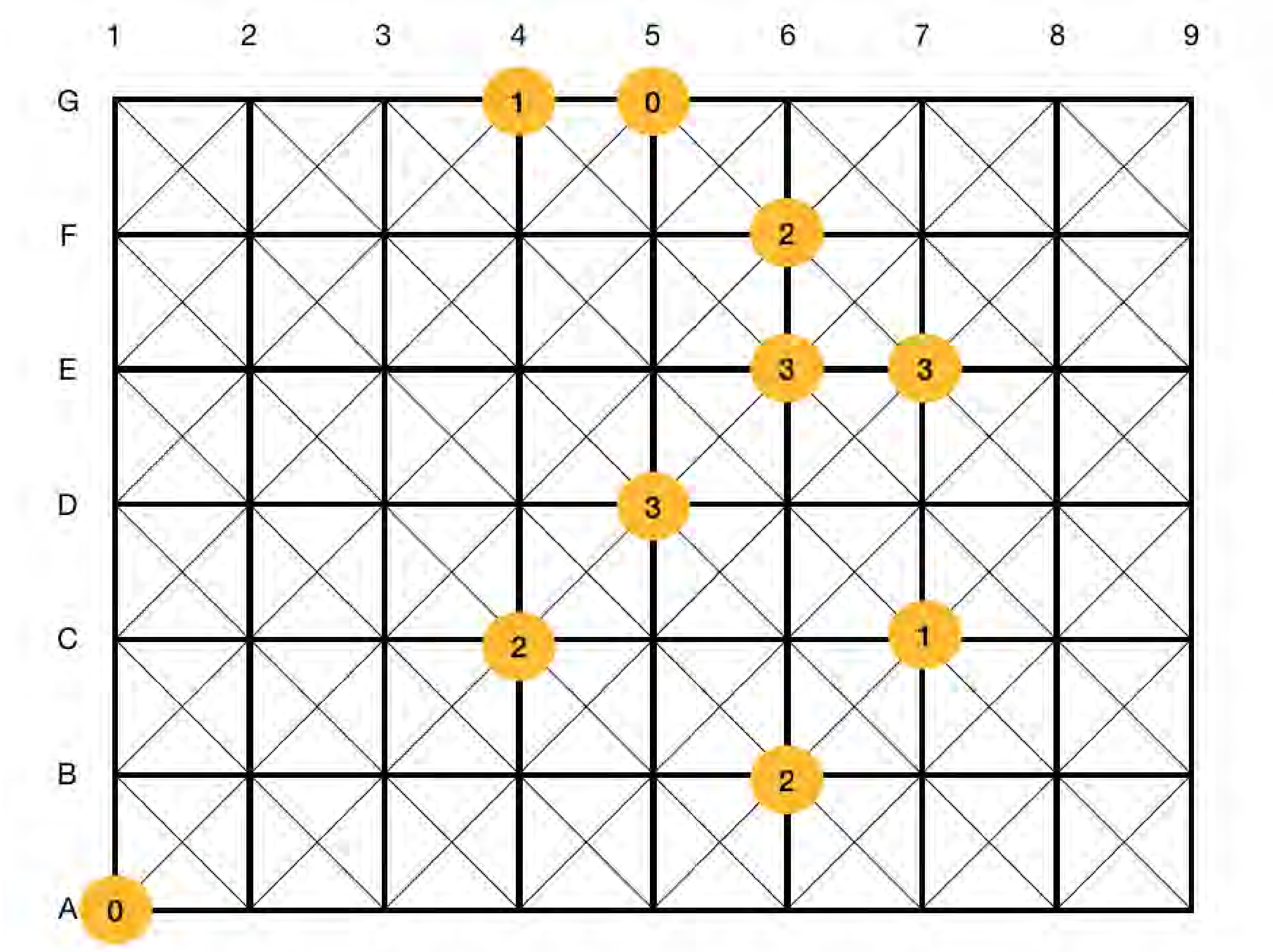
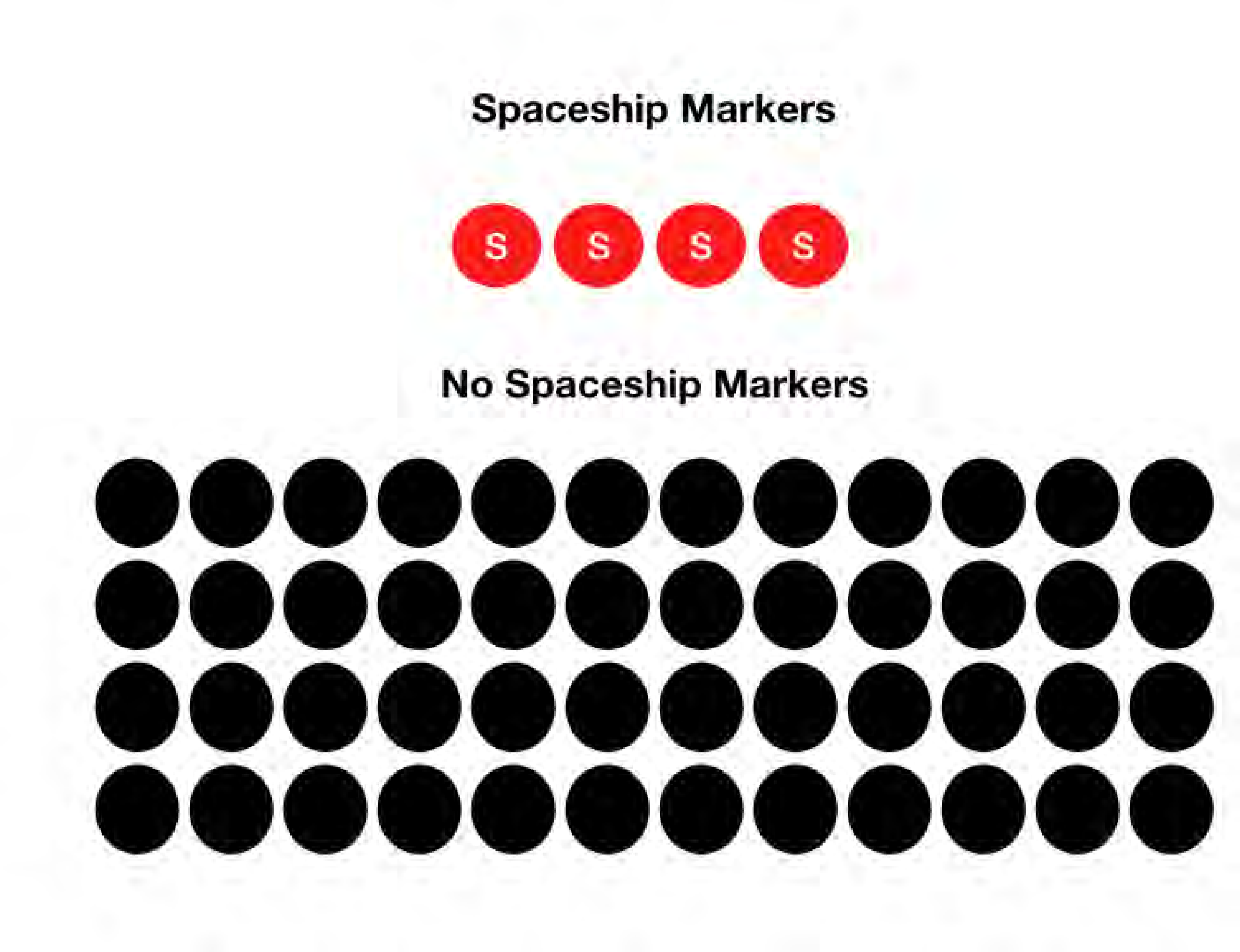


Image: Credit Australian Computing Academy, University of Sydney

This page is for

Students

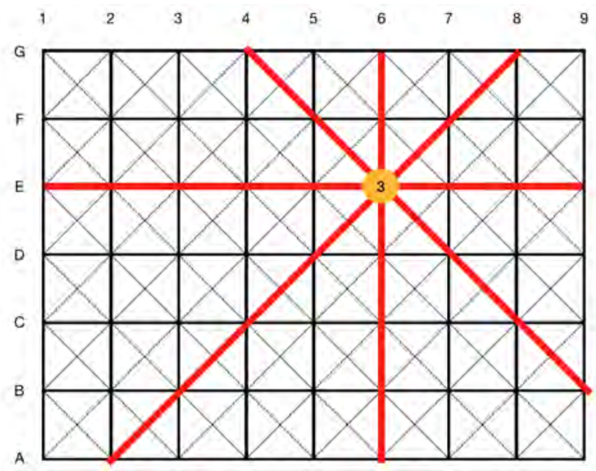
Preparation:
Cut out the red and black markers



Step 1

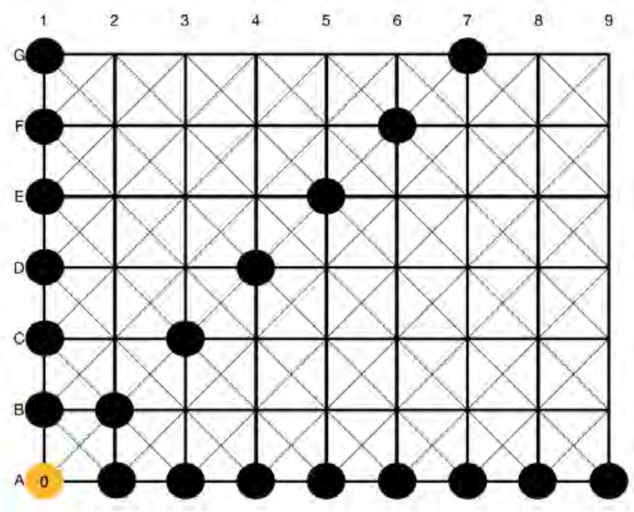
The yellow circles represent the space probes. Each one has a number that shows how many spaceships the probe sees horizontally, vertically and diagonally. So if the number is 3, then the probe sees exactly three spaceships along the lines that go in any direction from the probe to the end of the grid.

Here is an example. The probe on E6 sees three spaceships. The spaceships are somewhere on the thick red lines.



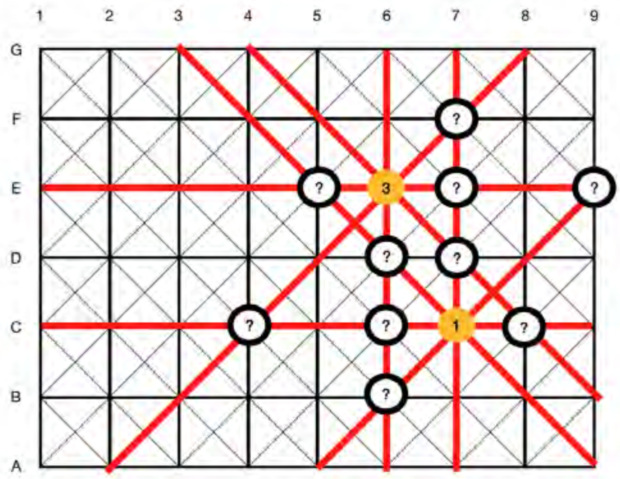
Step 2

If a space probe displays a zero, then it doesn't see any spaceship along its connected lines. Mark the places that are definitely free of spaceships with the black markers. This step can significantly reduce the search space, as shown in the figure below.



Step 3

The intersections of lines from multiple probes on the main grid are places where the spaceships could be.



Here is an example of potential spaceship locations using data from two probes. You can see that there are ten possible locations, marked with a '?' where one spaceship might be hiding. Why just one spaceship? That's because the probe at C7 sees only one spaceship. Remember each probe can only see 4 positions in any direction.

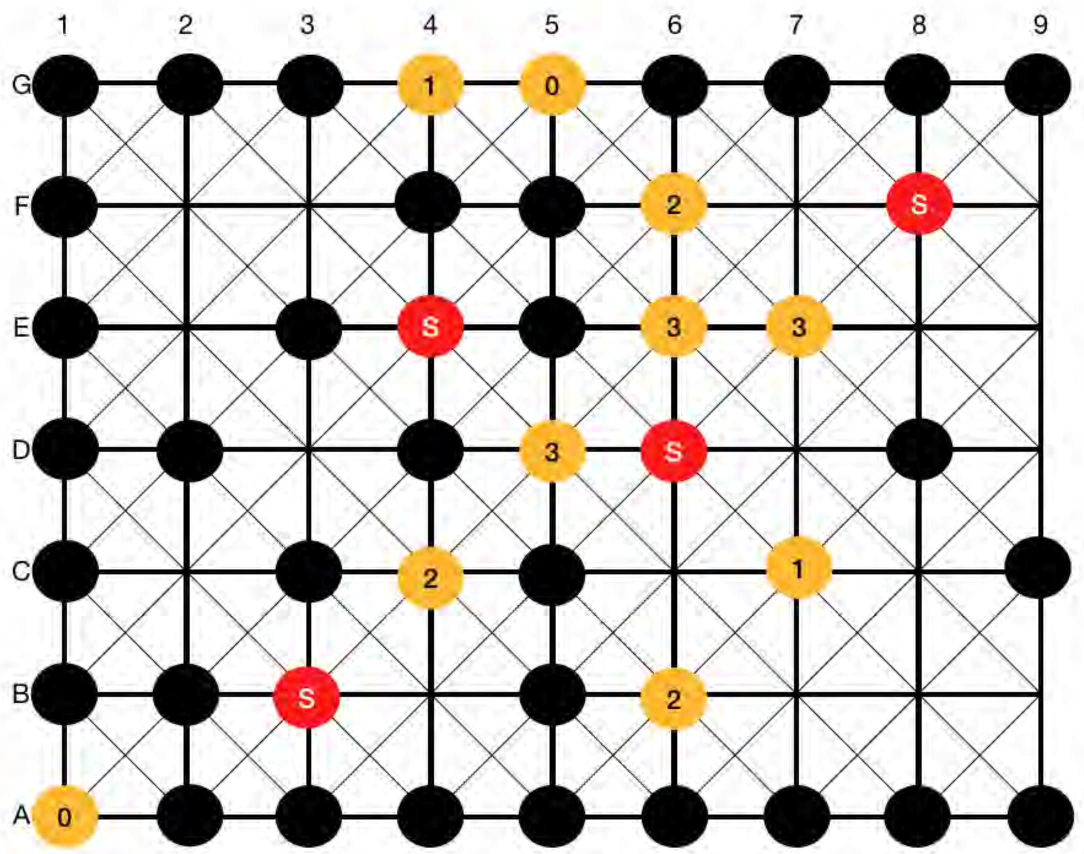
You need to follow the lines from multiple probes and determine if an intersection point is a sure location of a spaceship. Remember, each probe shows the total number of spaceships it can see, not more or less. Mark the location with a red spaceship marker.

Step 4

When you have placed all 4 spaceship markers, double-check that the numbers on the connected probes match with the number of spaceships a probe can see from its position. If you have made a mistake, place the spaceship marker somewhere better. When you think you have found all the spaceships, check your answer against the answer key (which your teacher or carer has.)

Answer key

This is a possible solution. It is entirely possible that your student may have found another valid solution!



More information for teachers

Here are some further activities, online resources, assessment ideas and curriculum references.

Adapting this activity

Once students understand how to complete this activity, ask them to prepare their own levels and share them with other students to solve them.
In a variation of the game rules, space probes cannot see spaceships that are hiding in the radio shadow of another spaceship. This makes the challenge more difficult to solve.

Keep the conversation going

- The activity is suitable for implementation as a computer game
- Students can work on designing an algorithm that randomly places four spaceships on a 9x7 grid.
- Another algorithm can inform the user about the number of spaceships that are visible from any user-selected grid position.
- Yet another algorithm can find the spaceships without user intervention.
- In a final step, students can design a game in which two human players play against each other or one human player against the computer.

Keep learning

For High School students interested in learning more about programming there are many courses to choose from at aca.edu.au/resources: you might like to try the space invaders game: cmp.ac/javaspaceinvaders

For teachers creating a portfolio of learning or considering this task for assessment

Ask students to describe the algorithm they followed to find the spaceships.

Linking it back to the Australian Curriculum: Digital Technologies

Algorithms
The precise sequence of steps and decisions needed to solve a problem. They often involve iterative (repeated) processes. (ACTDIP029 / ACTDIP040 - see cmp.ac/algorithms).

Refer to aca.edu.au/curriculum for more curriculum information.

Digital Technologies @ Home
Unplugged activities for students

This activity is for: Years 7-10

Cellular Automoji

This cellular automoji activity is an adaptation of *cellular automata*, but using emoji! 😉 Cellular automata are grids of cells that are filled depending on the number and type of neighbours surrounding them. The most famous cellular automata is Conway's Game of Life, a game created in the 70s; easy to get started with, but complex enough that it can simulate a computer!

This activity teaches algorithms

Algorithms are procedures that we follow to reach a goal. They can be as simple as following a recipe, or a complex set of rules like those for a game of sudoku. This activity is similar to sudoku, in that the algorithm is expressed using a set of rules, rather than an explicit sequence of steps.

By sharing the rules, we share the algorithm.

In this activity we start with a very simple set of rules and starting conditions, but by the end you see quite complex behaviour emerging from the system.

This behaviour is quite common in the natural world. For example, air behaves according to very simple rules about air pressure, temperature, and humidity, but when allowed to work across the entire Earth it creates very complex systems, like Monsoons, El Nino or the East Indian Dipole.

This activity is targeted towards secondary students, and should take about **45 minutes**.

See a demonstration

cmp.ac/automojivid

This page is for

Teachers

Parents and carers

Getting started (read this with your child):

We're going to make cellular automojis.

In cellular automoji we start with a single row of emoji at the top of a grid.

G1	🕛	🕛	🕛	🍕	🕛	🕛	🕛	🕛
G2								
G3								

Each row of emoji is called a generation.

Each generation is created from the one before, and each cell is filled depending on the three cells above it.

For the *pizza time* cellular emoji, the rules are:

Rules:

🍕	🕛	🕛
	🍕	

🕛	🕛	🍕
	🍕	

If no rule applies, we put in a base emoji: 🕛

Following these rules, the generations will evolve like this!

G1	🕛	🕛	🕛	🍕	🕛	🕛	🕛	🕛
G2	🕛	🕛	🍕	🕛	🍕	🕛	🕛	🕛
G3	🕛	🍕	🕛	🕛	🕛	🍕	🕛	🕛
G4	🍕	🕛	🍕	🕛	🍕	🕛	🍕	🕛
G5								

Can you fill in the fifth generation row together?

Tips for edge cases!

Since there's no rule for this pattern...

G1	🕛	🍕	🕛

The next generation is a clock!

G1	🕛	🍕	🕛
G2		🕛	

At the edges of the rows, where there is no emoji, we pretend there is a base emoji there..

So if a row starts like this:

G1	🕛	🍕

It produces a pizza on the edge of generation 2, because we pretend there is a clock where G1 is!

G1	🕛	🍕
G2	🍕	

Step by step

1. Learn how to fill in generations of cellular automoji
2. Draw the patterns out yourself.
3. Make your own!

Hint!

You don't need to do a perfect drawing of the pizza emoji when you fill in the generations. A triangle represents pizza just fine:

▽ = 🍕

Also, you don't need to fill in the base emoji if you don't want to. Sometimes leaving cells blank makes it easier to see the pattern!

This page is for

Students

Cellular Automoji

Follow the rules and fill in the cells

Question 1: New Shoe 💩 👟

Rules:

💩	👟	👟
	💩	

👟	👟	💩
	💩	

👟 base

Automoji:

G1	👟	💩	👟	👟	👟	👟	💩	👟
G2								
G3								
G4								

How many 💩 s in the 4th generation?

Question 2: Maze ⬜ 🌳

Rules:

⬜	⬜	⬜
	🌳	

🌳	🌳	⬜
	🌳	

🌳	⬜	🌳
	🌳	

⬜ base

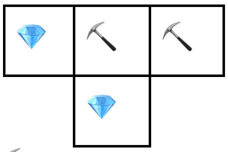
Automoji:

G1	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🌳
G2								
G3								
G4								
G5								
G6								

How many 🌳 s in the 6th generation?

Question 3: Diamond Miners ⛏💎

Rules:



⛏ base

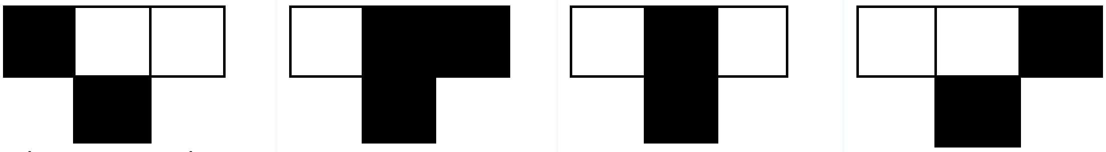
Automoji:

G1								
G2								
G3								
G4								
G5	⛏	⛏	⛏	⛏	💎	⛏	⛏	💎

Fill in the previous generations!

Question 4: Rule 30

Rules:



White square = base

Automoji:

G1								■							
G2															
G3															
G4															
G5															
G6															
G7															
G8															

What's the eighth generation?

This page is for

Teachers

Parents and carers

Answer key

Choose if you want to print this for your students or keep it to yourself!

Example: Pizza Time

The 5th generation looks like this

G5. 🕛🕛🕛🕛🕛🕛🕛🍕

Answer 1: New Shoe

There are 2 💩 emoji's in the 4th gen

G1. 👟💩👟👟👟👟💩👟
G2. 💩👟💩👟👟💩👟💩
G3. 👟👟👟💩💩👟👟👟
G4. 👟👟💩👟👟💩👟👟

Answer 2: Maze

There are 5 🌳 emoji's in the 6th gen

G1. 🌫🌫🌫🌫🌫🌫🌫🌳
G2. 🌳🌳🌳🌳🌳🌳🌫🌫
G3. 🌫🌫🌫🌫🌫🌳🌫🌳
G4. 🌳🌳🌳🌳🌫🌫🌳🌫
G5. 🌫🌫🌫🌳🌫🌫🌫🌫
G6. 🌳🌳🌫🌫🌫🌳🌳🌳

Answer 3: Diamond Miners

G1. 💎⛏⛏💎⛏⛏⛏⛏
G2. ⛏💎⛏⛏💎⛏⛏⛏
G3. ⛏⛏💎⛏⛏💎⛏⛏
G4. ⛏⛏⛏💎⛏⛏💎⛏
G5. ⛏⛏⛏⛏💎⛏ ⛏💎

Answer 4: Rule 30

G1								■							
G2							■	■	■						
G3						■	■			■					
G4					■	■		■	■	■	■				
G5				■	■			■				■			
G6			■	■		■	■	■	■		■	■	■		
G7		■	■			■					■			■	
G8	■	■		■	■	■	■			■	■	■	■	■	■

Want more?

Here are some further activities, online resources, assessment ideas and curriculum references.

Adapting this activity

Once students understand how to complete this activity, they could create their own rules, symbols and starting patterns and see what evolves!

Older students could create more complex rules that depend on two rows of cells, or five cells in the previous generation, rather than three.

Keep the conversation going

- Do you think some of these patterns will go on forever?
- Do they repeat themselves?
- What happens if you change the starting pattern?
- What happens if you change the rules?
- Rule 30 creates a really complex pattern from very simple rules. We call this **emergent behaviour**. Can you make another set of rules that causes behaviour like this?

Keep learning

For students interested in learning more about cellular automata look up **Elementary Cellular Automaton** or **Conway's Game of Life**.

To learn more about implementing algorithms, you can try the courses at the ACA website.

We recommend Javascript Space Invaders!
cmp.ac/spaceinvaders

For teachers creating a portfolio of learning or considering this task for assessment

For younger students:
Ask students to create their own set of rules for cells on a grid, and see how it develops.

Do some rules make for more interesting patterns than others?

For older students:
Ask students to write a program that follows the rules to print new generations.

Linking it back to the Australian Curriculum: Digital Technologies

Algorithms
Design algorithms represented diagrammatically and in English, and trace algorithms to predict output for a given input and to identify errors. (ACTDIP019 / ACTDIP029 - see cmp.ac/algorithms)

Refer to aca.edu.au/curriculum for more curriculum information.

Digital Technologies @ Home
Unplugged activities for students

This activity is for years 7-8 and 9-10

Card Switches

Thanks to Girls Programming Network for providing this content

This activity teaches…

Computers need to sort things into order all the time, for example, if you use the 'sort by' filter on a website the computer has to read through everything on the page and re-sort it depending on your selection. This activity gets you to think like a computer, using specific steps to compare and sort cards.

As well as sorting into order, you are also going to scramble the cards up as best you can! The more switches it takes you to sort them back into order, the better job you did scrambling. For example, a set requiring only one switch to be sorted has not been scrambled very well.

It is targeted towards students in years 7-8 and 9-10, with different discussion points separating the band levels. It can be expanded to include Insertion Sort and Quicksort if required.

It is expected to take **one hour** of class time. Print pages 2-4 and 8 for students (instructions, example walkthrough and card cut-outs).

You will need…

Pen, paper and scissors

Getting started (read this with your child):

We're going to try and scramble number cards as much as we can. After the cards are scrambled, we are going to count how many 'pairwise switches' it takes to sort them back into ascending order. What is a pairwise switch? It's when you switch the places of a pair of cards that are next to each other. The more pairwise switches it takes to sort the cards out, the more scrambled the cards were.

First we will practice with some random scrambles, then we will try and make the most scrambled set possible.

Step by step

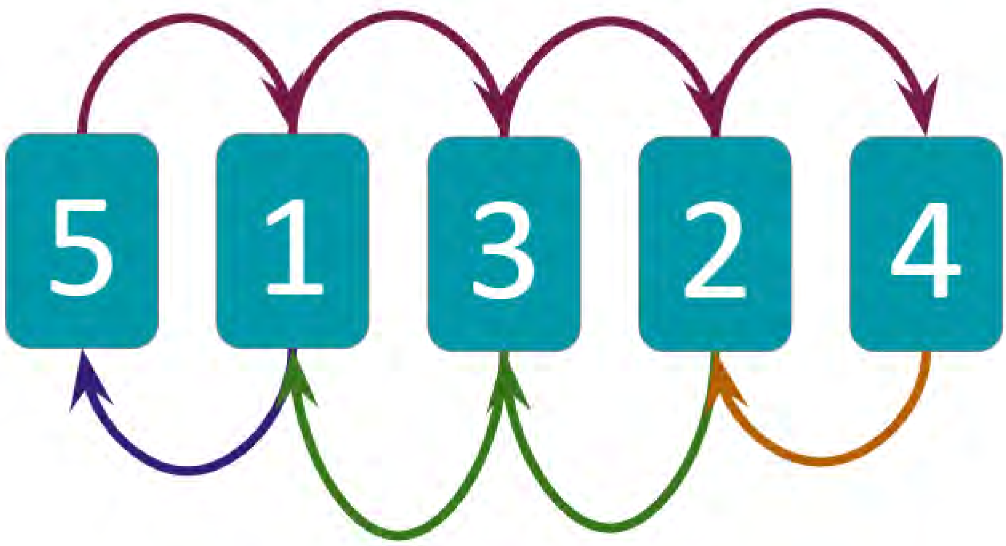
1. Have the student construct their cards and do their first unscrambling using the switches. If you would like to demonstrate try a scramble of 4, 1, 2, 3. It is a simple change (swap 4 and 1) but demonstrates the way the bubble sort 'floats' the highest number to the top after the first round of sorting.
2. Randomly scramble and sort a set of cards as a group, making sure you count the number of switches you make. The higher the number, the more scrambled the cards were.
3. Ask students to try and find the most scrambled sequence of cards. You are aiming to have the most number of pairwise switches you can.
4. When time is up, come back together and demonstrate the scramble and number of switches required.
5. Discuss which one is the ultimate Scramble and why.
6. Have the students turn the Pairwise Switch method into a pseudocode algorithm. (Consider specifying if the algorithms should be for an unknown number of cards and if it should count switches)
7. Have them validate their pseudocode by sorting a partner's scrambled cards with the partners pseudocode.

See a demonstration

This video demonstrates the example below.
cmp.ac/switchesvid

Card Switches

Learn how to use pairwise switches to sort cards into order and count how scrambled they are.



How to sort your cards using pairwise switches

Step 1

Cut an A4 sheet of paper into 12 cards. Write the numbers 1 to 6 (one number per card) onto one side of each. This should give you 2 sets of cards that go from 1 to 6 (or print and cut out the last page of this activity).

Step 2

Scramble your cards and lay them out.

Step 3

It's time to unscramble using Pairwise Switches. To do a Pairwise Switch, start at the leftmost pair of cards. **Do a switch when the card on the left is bigger than the card on the right.** Record how many times you make a switch.

Step 4

Work your way across the cards. When you reach the end, check the order of the cards. If they aren't in order yet, go back to the start of the cards and work through again. Keep going back to start until they're in the right order!

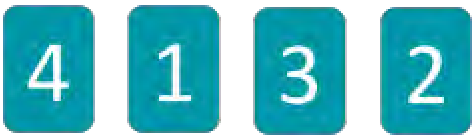
Step 5

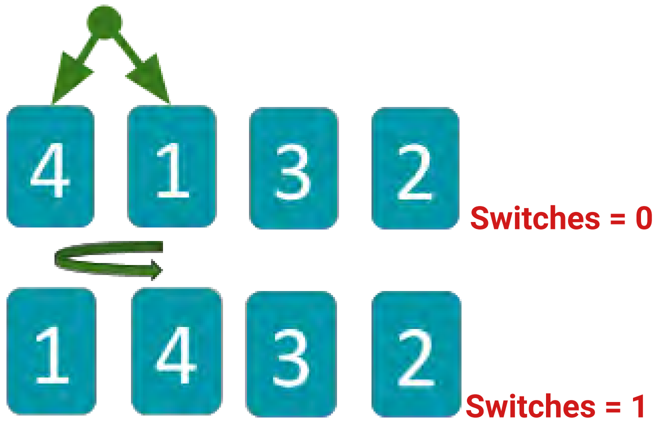
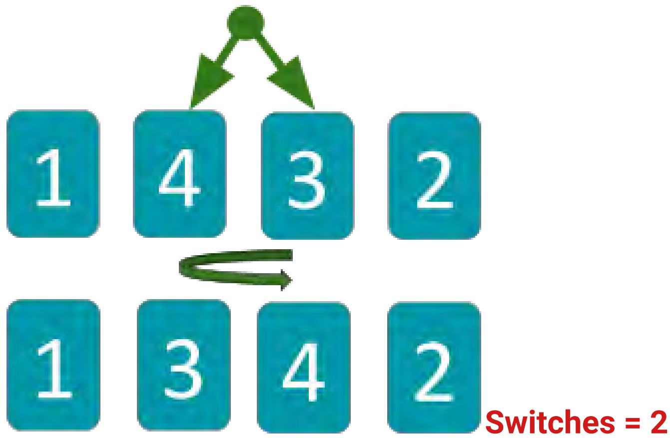
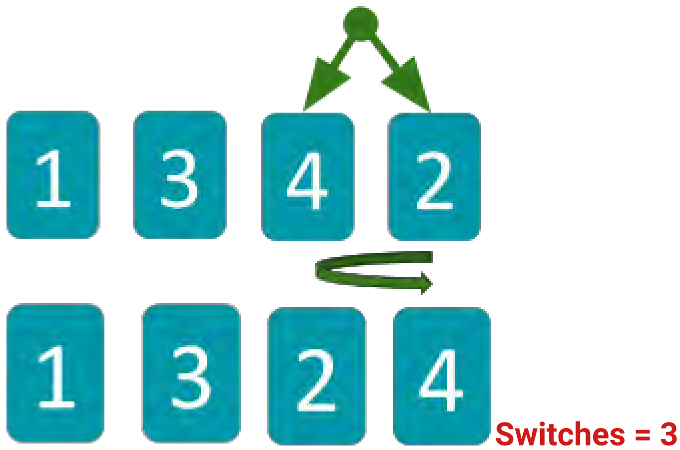
Now that you know how to sort them, try and find what combination of cards takes the most switched to unscramble. Try different combinations and see how many switches they take to solve. Make sure you keep a record of which combinations you try.

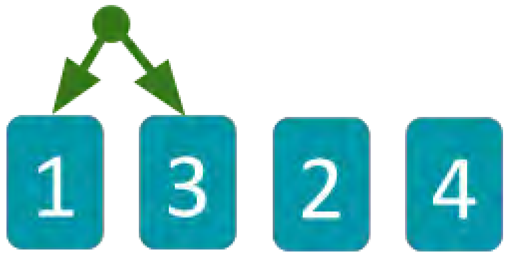
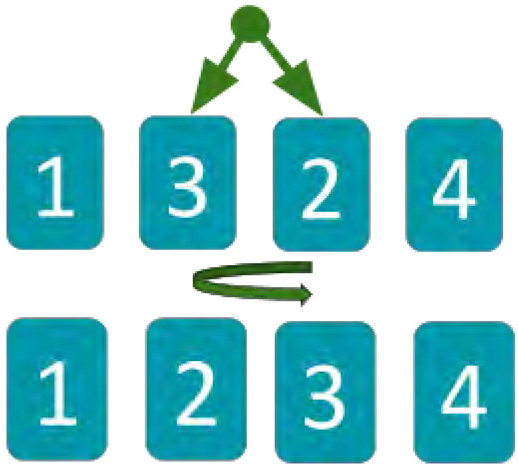
Step 6

Now it's time to turn your pairwise switch method into pseudocode. See if you can write the pseudocode to sort your set of cards. Can you write the pseudocode to sort a set of cards that you don't know the length of?

Let's unscramble one set of cards together.



Look at the first 2 cards, is 4 bigger than 1? Yes! Let's switch them.	
Look at the next 2 cards. Is 4 bigger than 3? Yes! Let's switch!	
Look at the next 2 cards. Is 4 bigger than 2? Yes! Let's switch!	

We are at the end of the row. Are the cards sorted? No. We need to go back to the start.	
Look at the first 2 cards again. Is 1 bigger than 3? No! So we leave the cards where they are.	
Next 2 cards. Is 3 bigger than 2? Yes! Let's switch.	 **Switches = 4**
Is 3 bigger than 4? No leave them	
Are the cards sorted? Yes! We are done! And it took us **4** pairwise switches to sort the cards!	

This page is for

Teachers

Parents and carers

Answer key

Choose if you want to print this for your kids or keep it to yourself!

The combination that will require the most pairwise switches is a reversed order of the original.

6 5 4 3 2 1

This will take **15 switches** to sort into ascending order.

To understand why, you should know that Bubble Sort is so named because it 'floats' the highest number to the top with each pass over the cards (like a bubble).

With that in mind, the way to make it take the longest is to move the highest numbers the furthest from the top of the list as you can.

Once the highest number is the furthest from the top, the second highest should be second furthest from the top etc etc.

There may be disagreement from students that reverse order is not exactly 'scrambled', it's a good opportunity to ask them why they think that. It could be because the linguistic meaning of 'scrambled' to them means 'without any order'. You can explain that in some cases, that is what scrambled means. But in this case, scrambled is being defined as "as many pairwise swaps away from the original order as possible."

Keep the conversation going - answers (pg 7)

(9-10) Can you write your sort as an algorithm?

This could be done using pseudocode or in a programming language students are familiar with.

A Python example:

```
#Define the Bubble sort function, which takes an array of objects
def bubbleSort(arr):
    n = len(arr)
    # Traverse through all array elements
    for i in range(n):
        # Last i elements are already in place
        for j in range(0, n-i-1):
            # traverse the array from 0 to n-i-1
            # Swap if the element found is greater
            # than the next element
            if arr[j] > arr[j+1] :
                arr[j], arr[j+1] = arr[j+1], arr[j]
```

(9-10) How could you optimise this bubble sort to be faster?

By adding a check to stop you from checking the already sorted cards. For example, a variable called 'unsorted_length' that gets shorted by 1 every time you go over the cards

(9-10) Is this a fast way to sort these cards? Can you think of a faster way?

This is designed to make the students think about alternatives to bubble sort (which is actually an inefficient sort, taking a lot of time)

(9-10) starting from a sorted set, what's the most scrambled you can make the cards using only 3 changes? Why?

This refers to the solution which states that the best way to scramble for a bubble sort is to move the highest number to the position farthest from its sorted position, the same for second highest , etc.

(9-10) Research Insertion Sort and Quicksort. Can you demonstrate those sorts with cards? What is the highest scramble combination you can make for these sorts? Watch the video link and read the explanation resources cited above.

- Insertion Sort - The worst case for Insertion Sort is actually the same as the Bubble sort. The set of cards sorted into reverse order will take the longest amount of time to sort. Each time the insertion sort collects the next card, it has to swap it with every preceding card, so while it does not cover the entire set of cards over and over, it still has to touch and swap each e=card in the dec.
- Quicksort - Given the random (sometimes, depends on implementation) assignment of the pivot position, it is not just a matter of the cards being in a certain order, it is also a matter of the pivot position. In a smaller set like the cards it doesn't make a huge amount of difference. In a larger set it can make a relatively unshuffled set take much longer to sort through.

  The worst case scenario for a quicksort would be a reverse order sorted set of cards with a pivot that sits at the highest or lowest point each iteration.

Want more?

Here are some further activities, online resources, assessment ideas and curriculum references.

Keep the conversation going

- What strategy did you use to scramble your cards?
- (7-8) If these cards were letters instead of numbers, how would you sort them?
- (7-8) If these cards were animals instead of numbers, how would you sort them?
- (7-8) Did you know that you were using a Bubble Sort to sort these cards?
- (7-8) Why do you think it is called bubble sort?
- (9-10) Can you write your sort as an algorithm in a programming language?
- (9-10) How could you optimise this bubble sort to be faster?
- (9-10) Is this a fast way to sort these cards? Can you think of a faster way?
- (9-10) starting from a sorted set, what's the most scrambled you can make the cards using only 3 changes? Why?
- (9-10) Research Insertion Sort and Quicksort. (Links below, or you can find your own) Can you demonstrate those sorts with cards? What is the highest scramble combination you can make for these sorts?

Keep learning

For High School students interested in different sorting algorithms

- An text based introduction to quicksort https://medium.com/karuna-sehgal/a-quick-explanation-of-quick-sort-7d8e2563629b
- An text based introduction to Insertion sort https://medium.com/karuna-sehgal/an-introduction-to-insertion-sort-16b97619697
- A good overview video of the sorts , but with an emphasis on time taken to do them. https://www.youtube.com/watch?v=WaNLJf8xzC4

Some more resources for students interested in watching sorts in action.

- http://sorting.at/
- https://www.youtube.com/user/AlgoRythmics
- https://visualgo.net/bn/sorting

For teachers creating a portfolio of learning or considering this task for assessment

Present students with a sequence of objects that need sorting into an order. Have the student sort the objects and determine how many swaps are needed to complete the sort.

Note: It is not recommended to use numbers or letters as sorting objects as there are a number of tools available online that will sort and count the swaps for you. Choose a random set of images, provide the students with the scrambled order to work from and the unscrambled one for reference.

Linking it back to the Australian Curriculum: Digital Technologies

Algorithms

(7-8) Design algorithms represented diagrammatically and in English, and trace algorithms to predict output for a given input and to identify errors (ACTDIP029 - see cmp.ac/algorithms)

(9-10) Design algorithms represented diagrammatically and in structured English and validate algorithms and programs through tracing and test cases (ACTDIP040 - see cmp.ac/algorithms)

Refer to aca.edu.au/curriculum for more curriculum information.

1	1
2	2
3	3
4	4
5	5
6	6

This page is for

Digital Technologies @ Home
Unplugged activities for students

Teachers
Parents and carers

This activity is for: Years 7-8, 9-10

Hamiltonian and Eulerian Paths and Circuits

With thanks to Girls Programming Network for providing this content.

This activity teaches…

This is an introduction to Graph Theory problems and their real world applications by looking at Hamiltonian and Eulerian (pronounced "oil-air-ian") Paths and Circuits.

It helps to teach problem specification by showing scenarios broken down to 2 main components - the places to visit and the paths between them - as a diagram. Being presented with the visual map of the problem, students learn how to use it as a tool to find solutions to different scenarios that present themselves. It requires 'trial and error' solutions, introducing algorithmic filters to solutions later on.

It is targeted towards secondary students in years 7-10 and is expected to take 1.5 to 2 hours.

You will need…

Paper, scissors, pencil and eraser (Optional: a plastic sleeve and whiteboard marker. Place the printed graph question in the sleeve and use the whiteboard marker on the plastic so that it can be erased easily.)

Getting started (read this with your child):

Imagine you are trick-or-treating around your neighbourhood. You have to get to each house without stopping at any of the houses twice and you need to start and finish at your own house. This is an example of a Graph Theory problem that needs solving! What you need is called **a Hamiltonian circuit**: it's a path around the suburb that stops at each house once and gets you back home.

Now imagine you are in charge of re-paving the roads in your neighbourhood. You need to cover all of the roads, but you can't go over a road twice because the road will still be wet and your paving machine will get stuck. This is also a graph theory problem, and you're going to learn to find the solution.

Step by step

Work through the explanation sheets, discussing the rules of different paths and circuits. Work through the first map together, finding the different paths and circuits. (If you want to draw on the maps use pencil and eraser or the plastic sleeve option detailed above). There are multiple answers to many of these graphs.

Make sure your child understands the difference between the different routes they are expected to find. Then give your child time to work through the later options by themselves.

This page is for Students

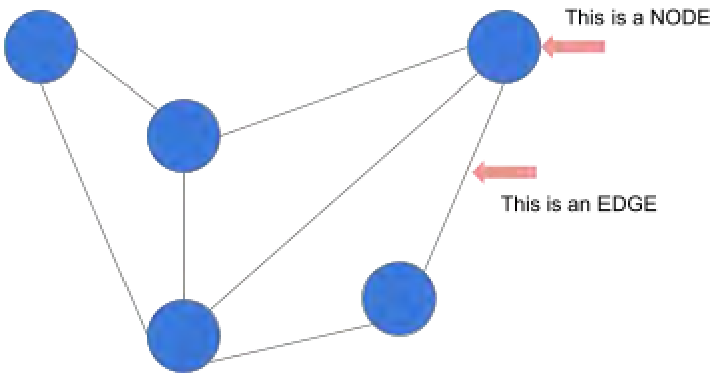
Graph Theory Circuits

Learn how to find Hamiltonian and Eulerian Circuits for different maps!

Image designed by rawpixel.com, hosted on Freepik

Graph theory problems are all around us. They look at the different ways you can move around places and objects, depending on what you need to do. The first thing to learn is how to read the graphs you will find in the puzzles.

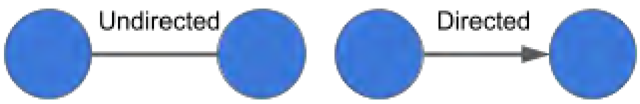
These graphs don't have an x and y axis, they are made up of **Nodes** and **Edges.** The **nodes represent places to visit** and the **edges represent the different ways to get to the nodes**. You could think of it like houses and the streets that connect them.



Seems simple right? It's designed to be! Using these graphs we can break down what look like complex problems into easier to manage graphs.

We are going to look for ways to move around these graphs that solve problems. Specifically, Hamiltonian and Eulerian paths and circuits. The different types solve different problems, one is about visiting each node (Hamiltonain) the other is about visiting each edge (Eulerian).

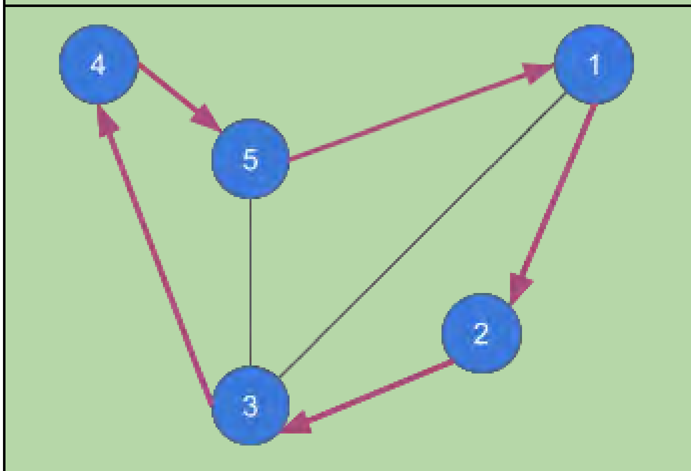
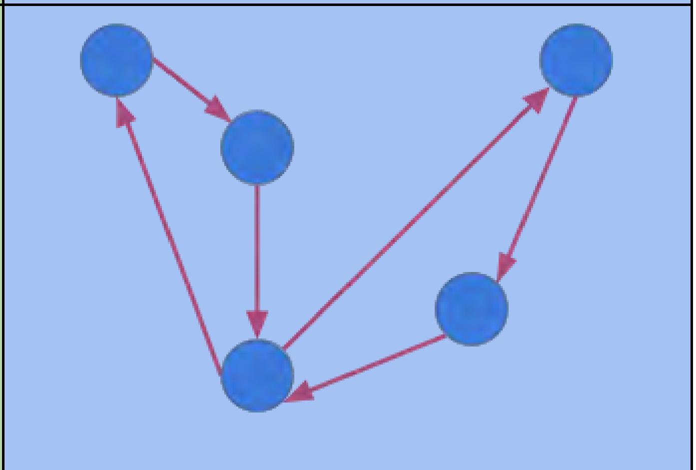
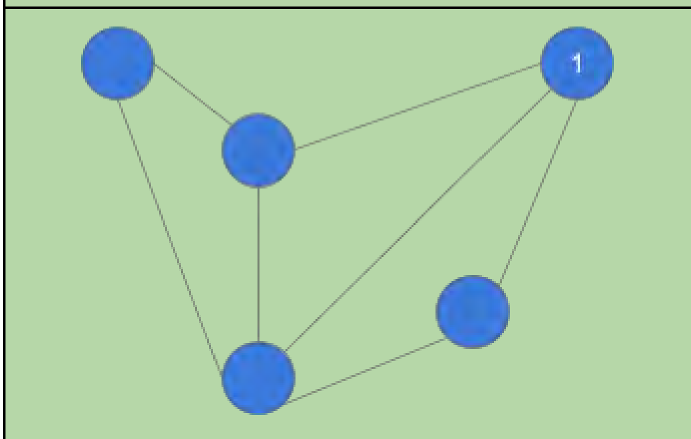
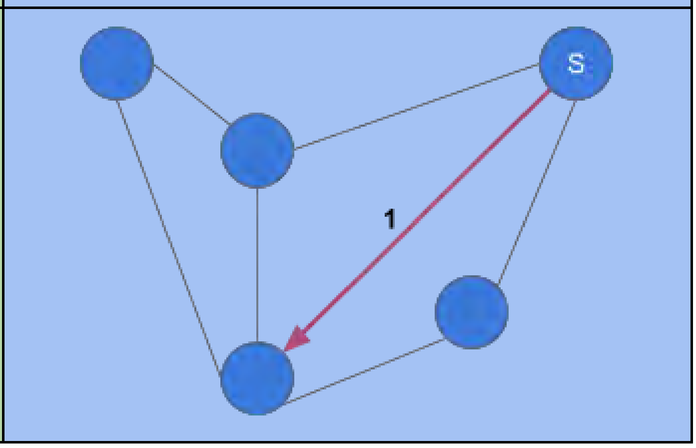
The last thing to learn is that we are working with **undirected graphs**. The means you can go either way up or down the edges. A directed graph would have arrows on the edges that you would have to follow.

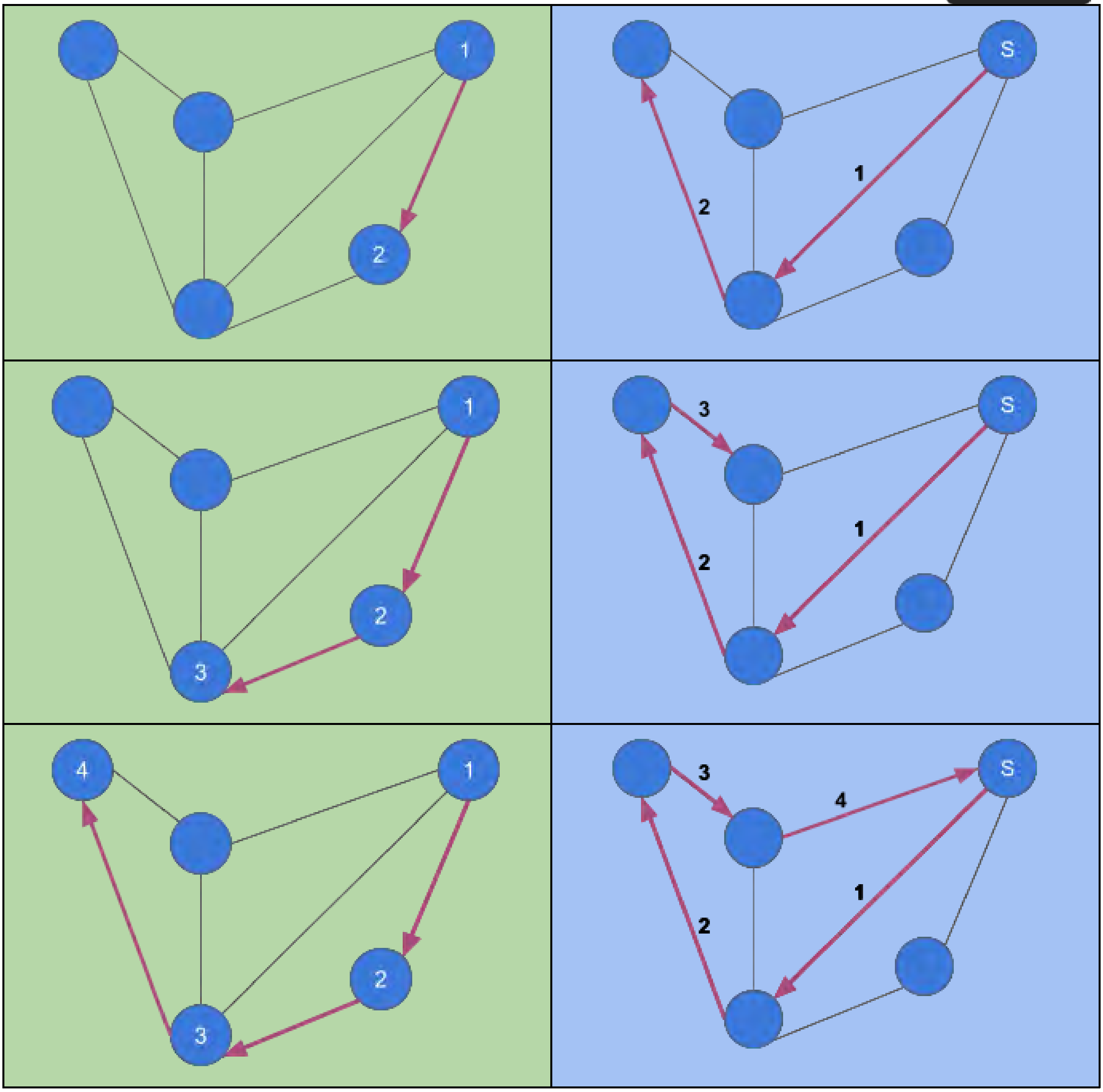


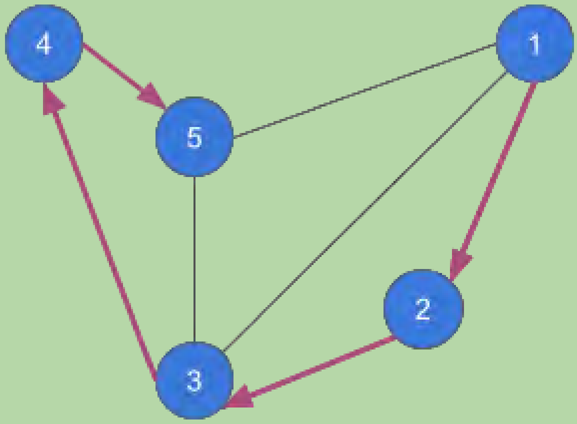
For each undirected graph question, there are 4 potential puzzles to find:

Hamiltonian Path - Visit every each **node** exactly once - Finish anywhere	**Eulerian Path** - Visit each **edge** exactly once - Finish anywhere
Hamiltonian Circuit - Visit every each **node** exactly once - Finish back where you start	**Eulerian Circuit** - Visit each **edge** exactly once - Finish back where you start

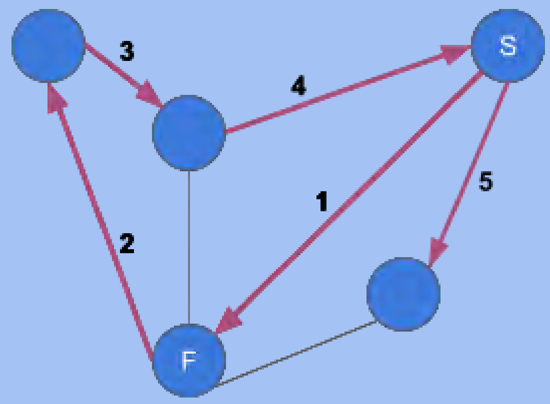
Not every graph has a Hamiltonian or Eurlerian property!
You might not be able to find all 4 puzzles in all the graphs.

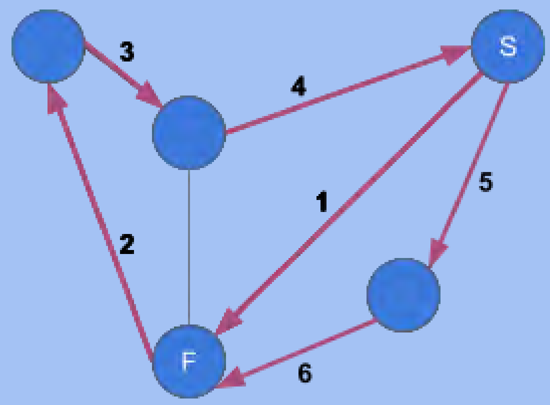
Hamiltonian Circuit Example	**Eulerian Circuit Example**
	
A Hamiltonian circuit solution will visit each node and **must finish at the start node.** If it was a path, it could finish at any node.	An Eulerlian circuit solution will visit each edge and **must finish at the start node.** If it was a path, it could finish at any node.
Hamiltonian Path Example	**Eulerian Path Example**
	

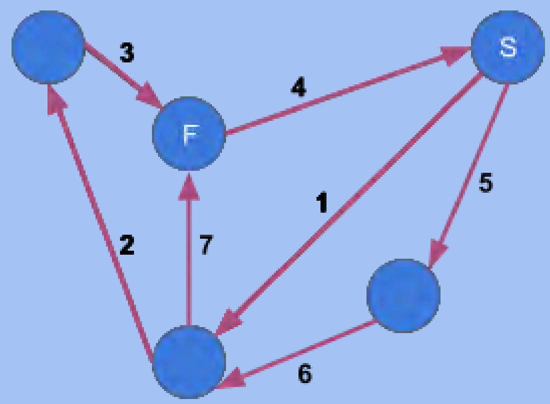

1
2
S
1
2
1
2
3
S
3
1
2
4
1
2
3
S
3
4
1
2



- All nodes visited
- No need to get to start node







- All paths visited
- No need to get back to start node

You can choose to start at any node, unless the question tells you specifically where to start.

Something interesting to remember, **if a graph has an Eulerian circuit, you can start at any node and complete the circuit.**

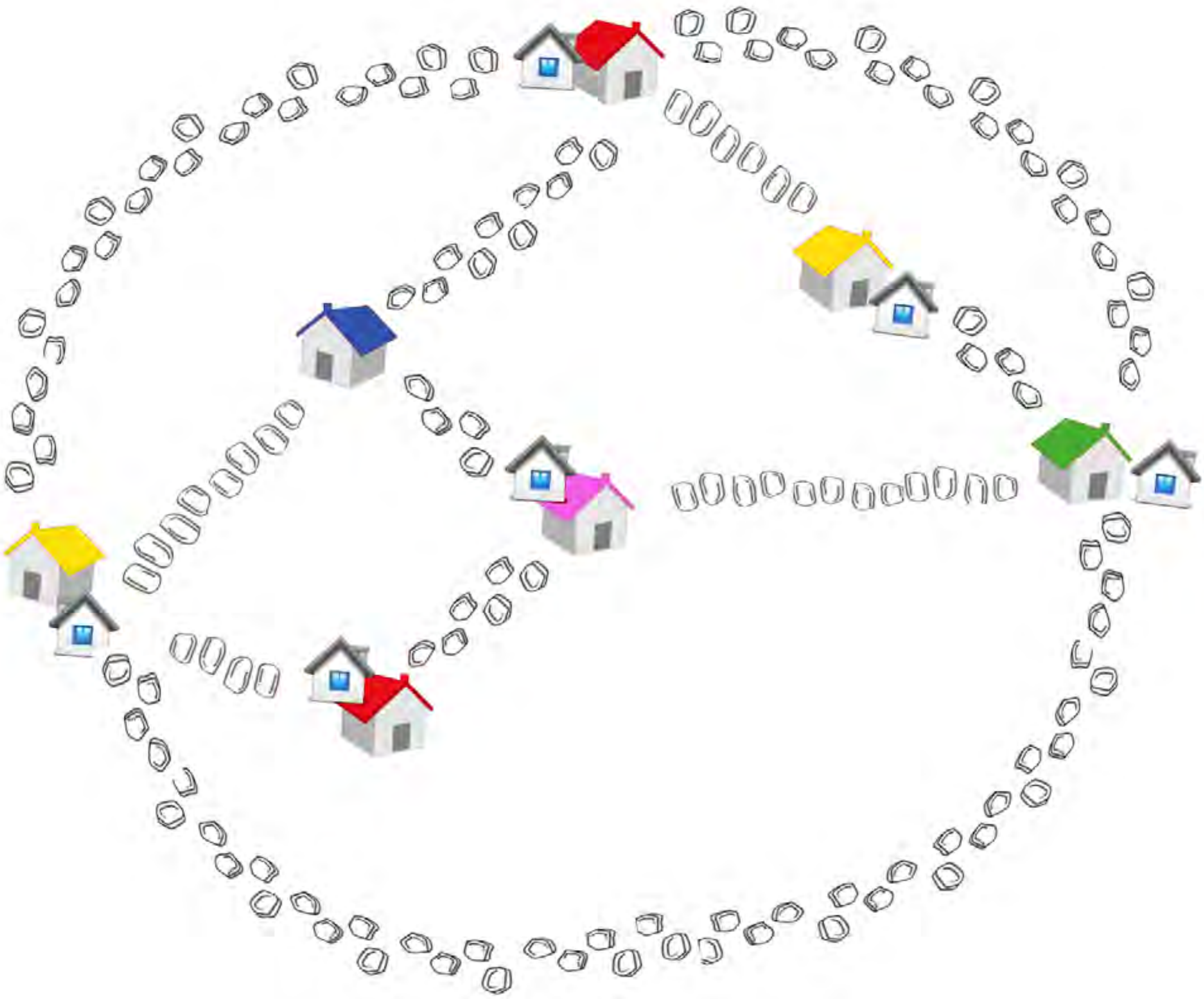
Puzzle 1!

You are the bus driver of the school bus, you drive around the neighbourhood and pick up students and take them to school each morning. Starting from the Green School House, find a route that visits each house node exactly once. You can finish anywhere.

1. Which kind of path do you need to find?
2. Find a path
3. Find a circuit that starts and finishes at the green school house.

Now imagine you are in charge of drawing maps for the suburb. You need to go down every street in the suburb and write their names down for the map. You are trying to be as efficient as possible, so you don't want to go over any streets twice.

4. Which kind of path do you need to find?
5. Find a path that goes over every street exactly once. You should start from the blue house but you can finish at any house. (This one can be tricky).



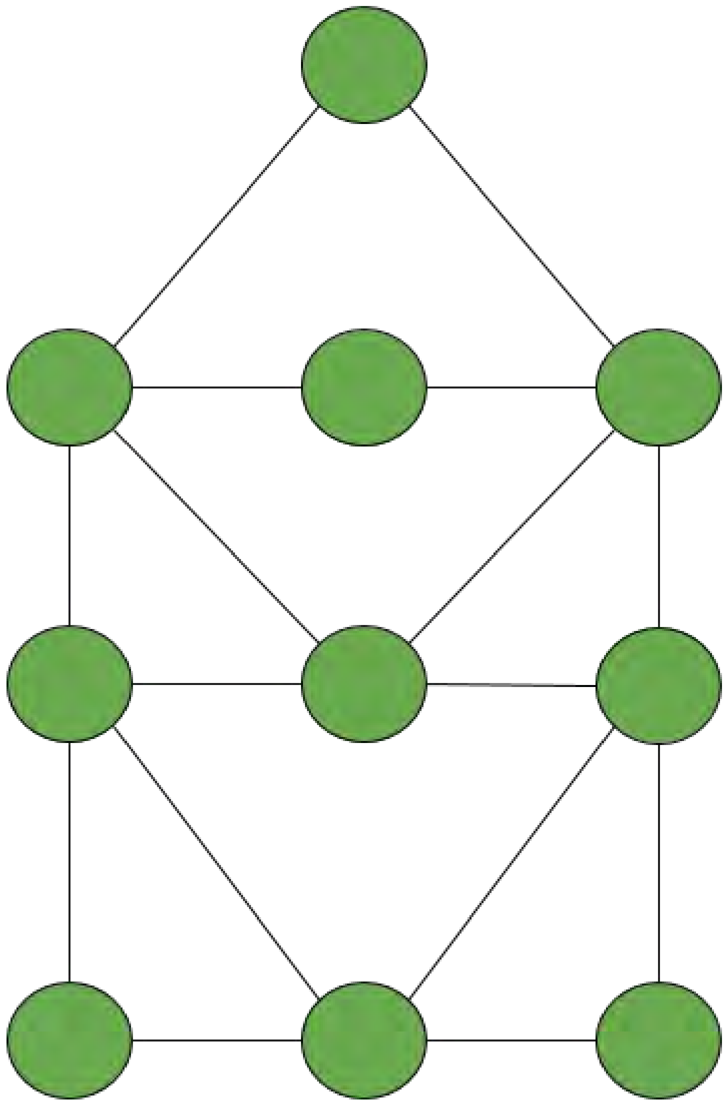
Puzzle 2!

It's time to paint the fences in the garden! This graph represents the fences (edges) and plants (nodes) in a garden. You have to find a path to paint every fence, you only have enough paint for 1 coat each, so you can visit each edge only once.

1. Which kind of path through the graph are you looking for?
2. Find a path that paints each edge exactly once, you can start anywhere.
3. Can you find a circuit? (It might not have one)

Now that all the fences are freshly painted, you should make sure the plants all have fertiliser. You only have enough fertiliser to visit each plant once.

4. What kind of path are you looking for?
5. Find a path that gets to every plant exactly once, starting anywhere.
6. Can you find a circuit? (It might not have one)



Puzzle 3!

This graph represents the classrooms (nodes) at school and the paths (edges) that connect them together.

At school your teacher asks you to go along all the paths to make sure they are clean and tidy, but you only have a few minutes before the start of class so you can't go over any twice.

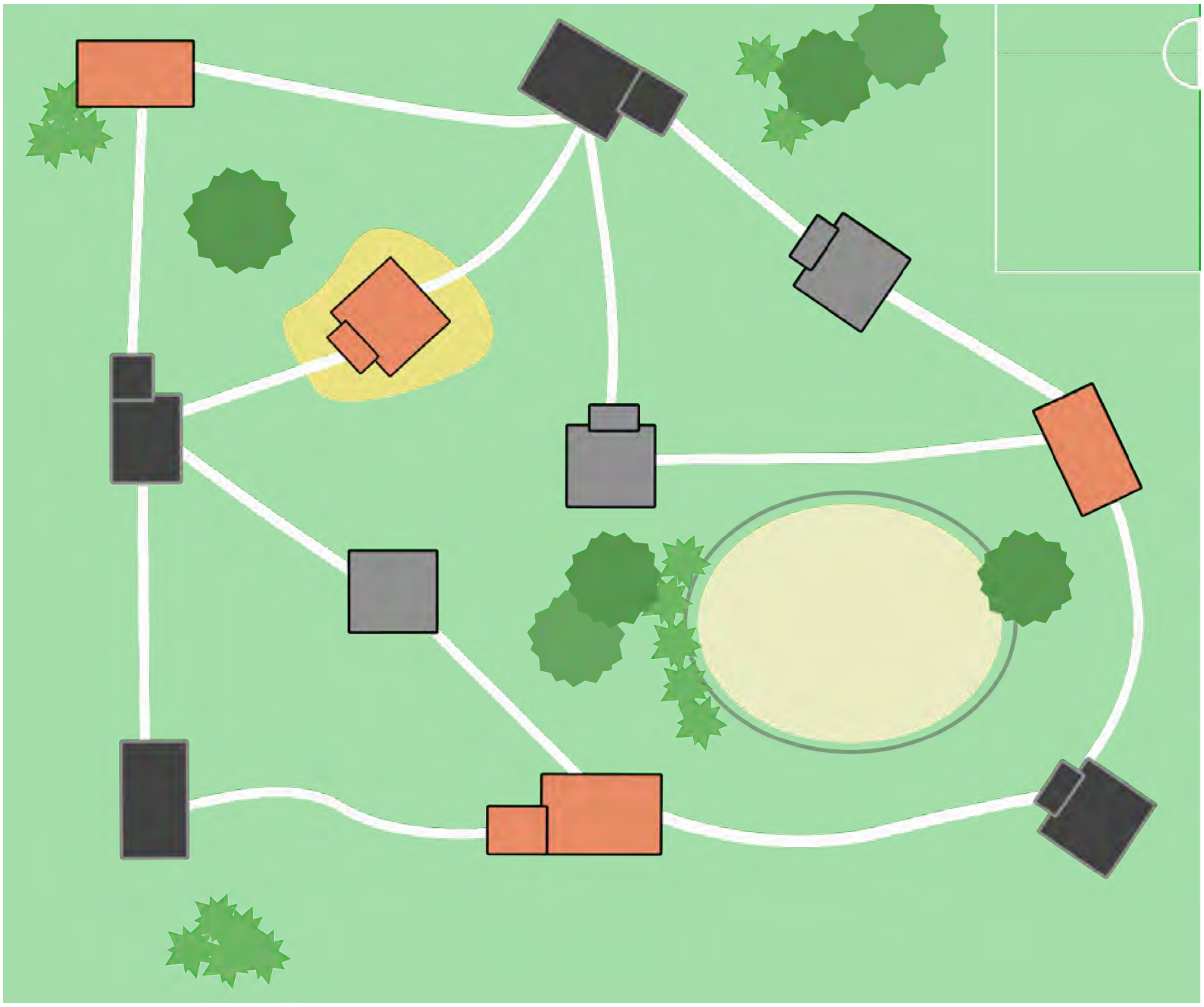
1. What kind of path are you looking for?
2. Can you find a path through the graph?
3. Can you find a circuit? (It might not have one)

You are in charge of collecting the recycling bins in all of the classrooms, you need to visit each room but only once!

4. What kind of path are you looking for?
5. Can you find a Path? (It might not have one)
6. Can you find a circuit? (It might not have one)

Hint question!

7. Adding one extra edge will make a circuit possible. Can you find which one? (Have a look at the next question for some guidance)



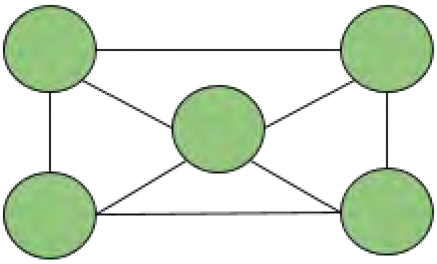
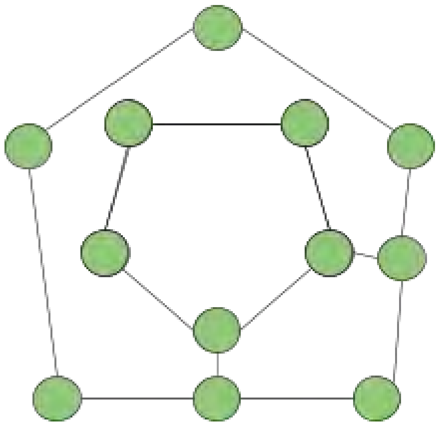
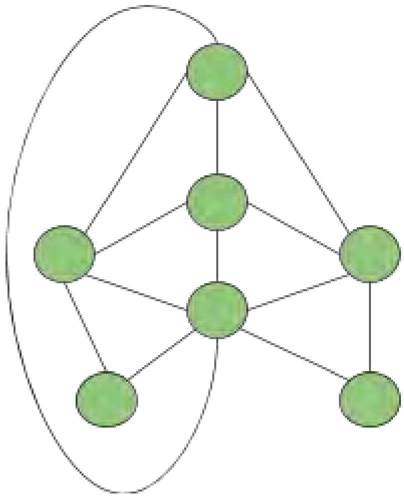
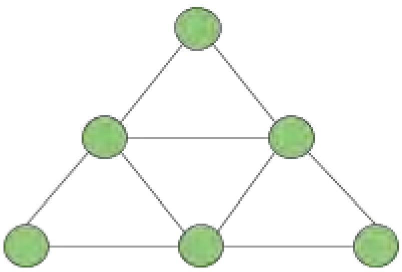
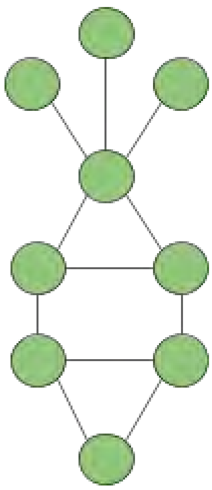
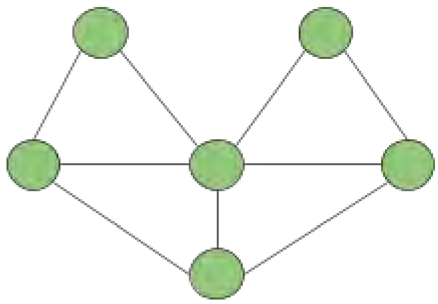
Bonus Puzzle! Eulerian quickfire round.

Looking for the paths and circuits can take a while, especially if there isn't one! Luckily, there are a few rules we can use to quickly check for Eulerian paths/circuits in a graph without having to trace any routes along the graphs.

For each undirected graph:

- An Eulerian Circuit is only possible if **every node has an even number of edges.**
- An eulerian Path is only possible if **every node has an even number of edges** or **exactly 2 nodes have an odd number of edges.**

Using those rules, can you check for Eulerian paths and circuits in the graphs below using just the number of edges?

 **Path?: Y / N** **Circuit? Y / N**	 **Path?: Y / N** **Circuit? Y / N**	 **Path?: Y / N** **Circuit? Y / N**
 **Path?: Y / N** **Circuit? Y / N**	 **Path?: Y / N** **Circuit? Y / N**	 **Path?: Y / N** **Circuit? Y / N**

This page is for

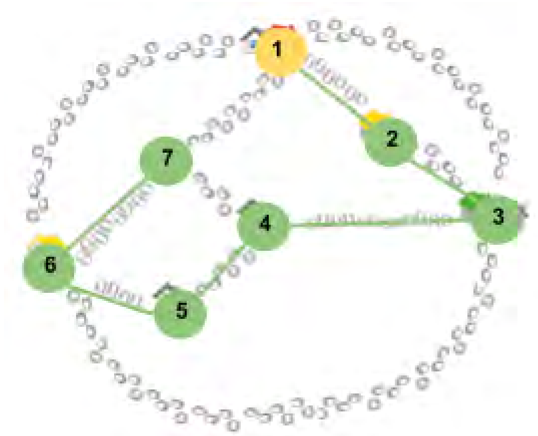
Parents and carers

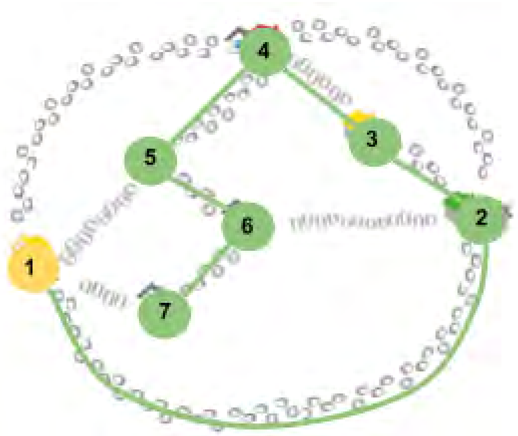
Answer key

Choose if you want to print this for your kids or keep it to yourself!

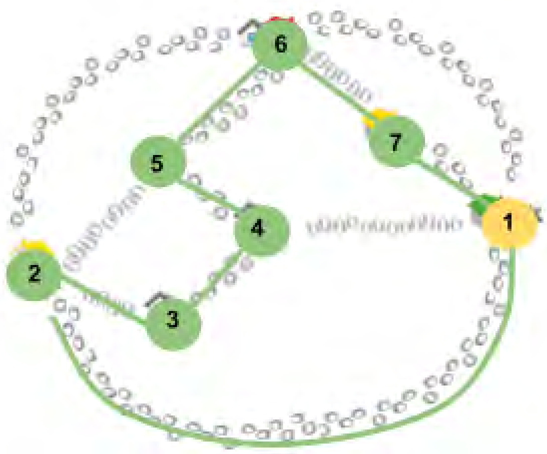
Puzzle 1

1. **A Hamiltonian Path will reach all the houses exactly once.**
2. Find a **Hamiltonian** path. **There are multiple answers, 2 examples are below.**

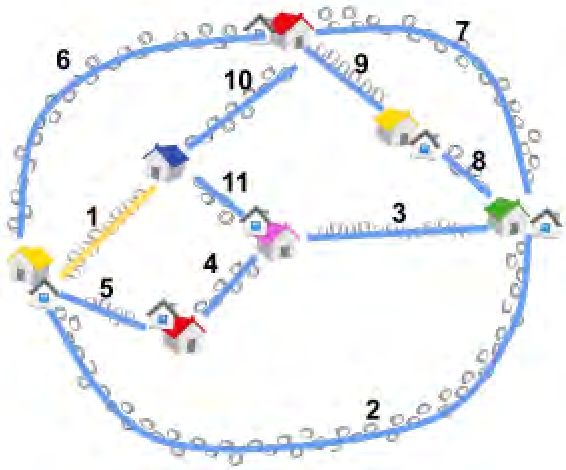




3. Find a **Hamiltonian** circuit (start and finish at the green house) **There are multiple answers, this is just one example.**

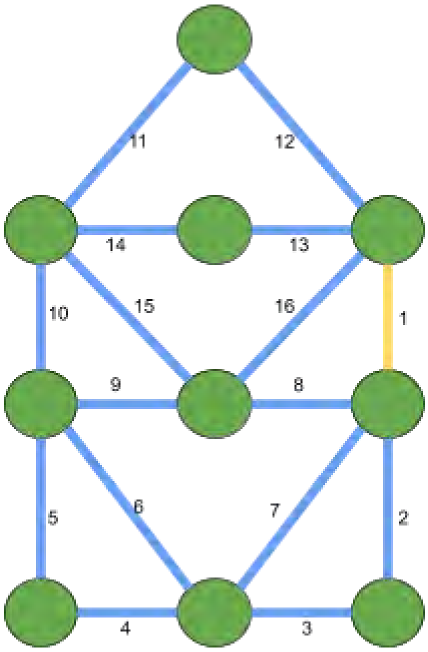


4. To find the street names you need an **Eulerian Path.**
5. Find an Eulerian circuit. **There are a few answers, this is just one.**

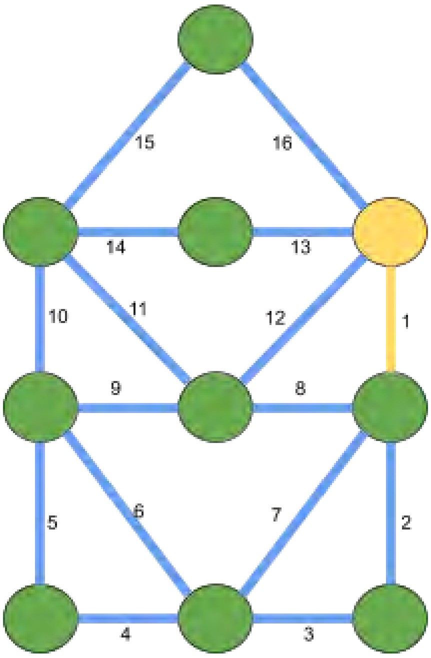


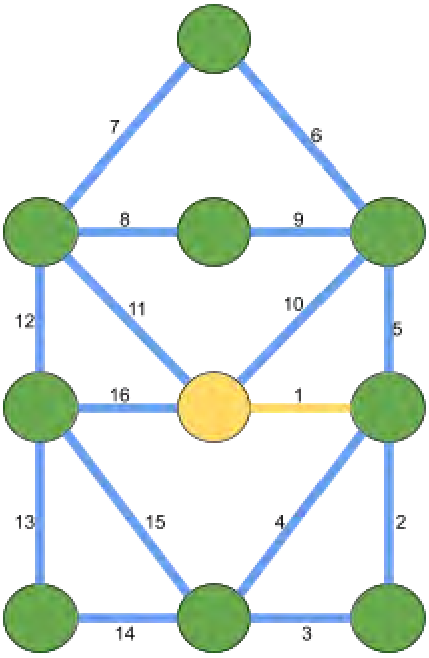
Puzzle 2!

1. You need to find an **Eulerian Path.**
2. Find An Eulerian Path. **There are multiple solutions, this is just one example**

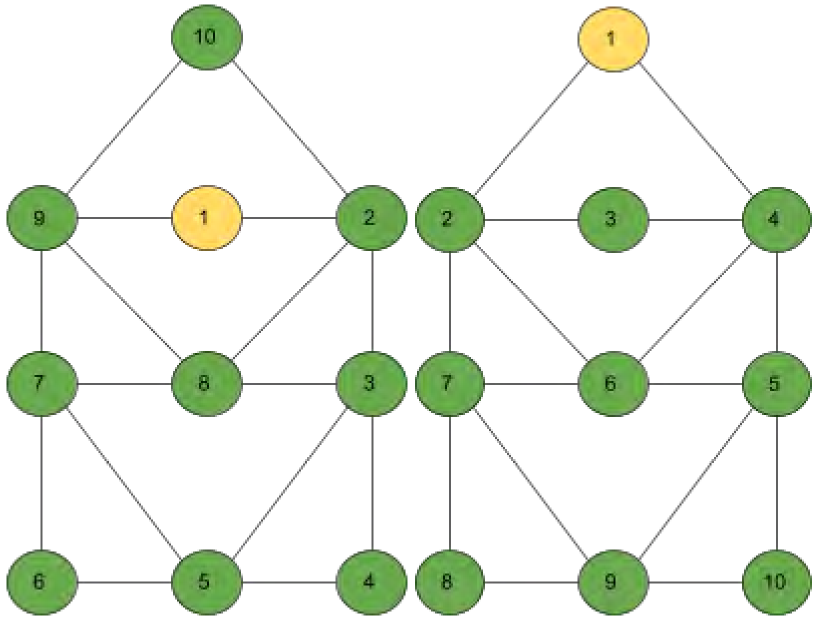


3. Find an Eulerian Circuit. **There are multiple answers, these are 2 examples.**





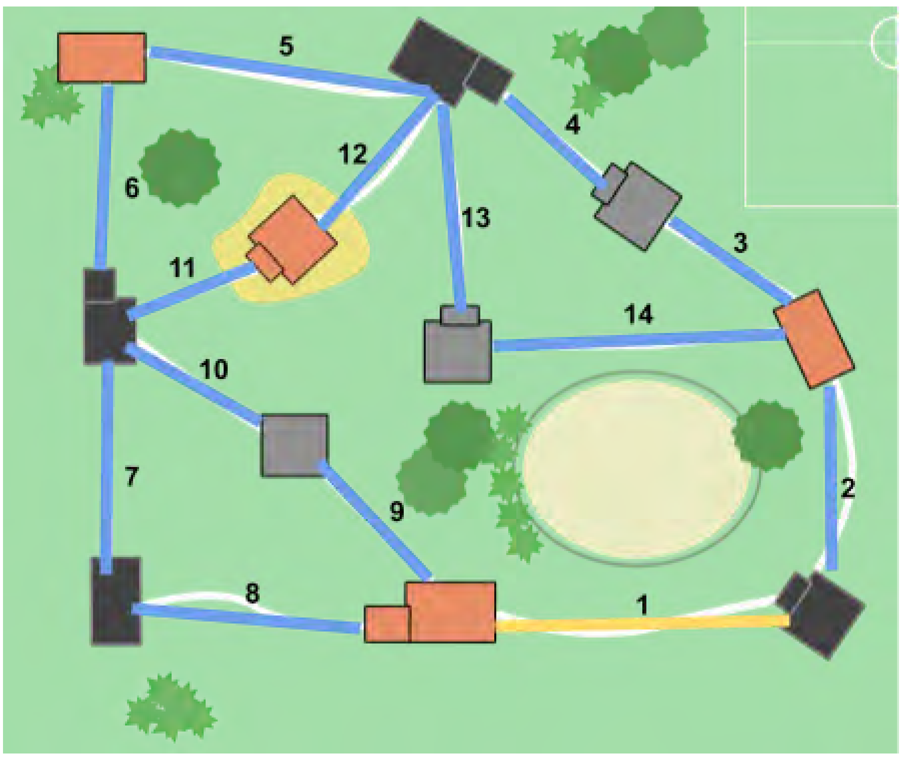
4. **A Hamiltonian Path** reaches each of the nodes.
5. Find a Hamiltonian Path. **There are multiple answers, this is just two examples.**



6. Find a Hamiltonian Circuit.
   This graph does not have one.
   There is no way to go to each node exactly once and get back to where you started.

Puzzle 3!

1. An Eulerian Path
2. Find a path. **There are multiple answers, this is just one example.**

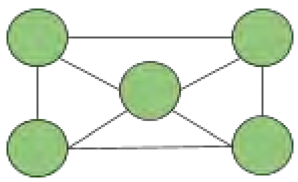
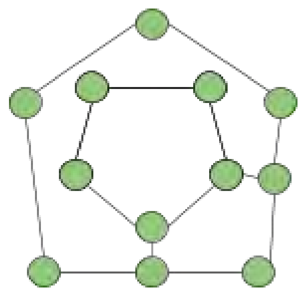
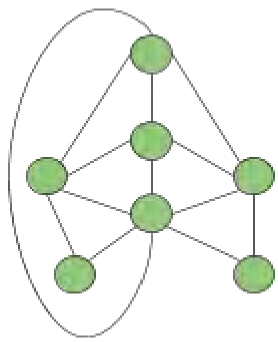
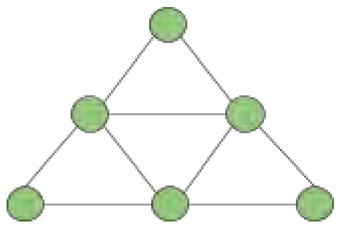
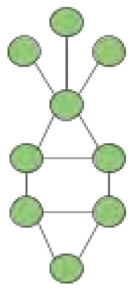
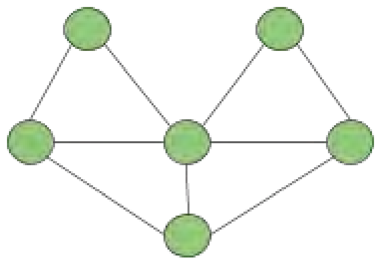


3. Find a circuit
   This graph does not have an Eulerian Circuit. There is no way to go to each edge exactly once and get back to where you started.

4. What kind of Path are you looking for? **A Hamiltonian Path.**
5. Can you find a path?
   This graph does not have an Hamiltonian Path.
   There is no way to go to each node exactly once and get back to where you started.
   This means it cannot have a Hamiltonian Circuit either.

Bonus Eulerian Puzzle.

- An Eulerian Circuit is only possible if **every node has an even number of edges.**
- An eulerian Path is only possible if **every node has an even number of edges or exactly 2 nodes have an odd number of edges.**

 **Path?: No** **Circuit? No**	 **Path?: No** **Circuit? No**	 **Path?: Yes** **Circuit? Yes**
 **Path?: Yes** **Circuit? Yes**	 **Path?: No** **Circuit? No**	 **Path?: No** **Circuit? No**

Want more?

Here are some further activities, online resources, assessment ideas and curriculum references.

Keep the conversation going

- We only worked with undirected graphs in this activity. Can you think of a real world example of a Directed graph?
- Try drawing your own directed graph and solving as many paths and circuits as you can. If you can't find any paths at all, adjust your directions and try again.

Keep learning

For students who would like to try designing their own paths online, there is a graph construction tool that will also help you find paths and circuits:
https://graphonline.ru/en/

Students interested in exploring a theorem for determining if a graph has a Hamiltonian Path can look into Ore's Theorem.
A good introduction can be found at:
https://study.com/academy/answer/how-to-determine-if-the-graph-is-hamiltonian.html

ACA Resources

(7-8) Students use physical characteristics of different animals to develop an algorithm that allows you to easily group and identify each animal based on a series of simple questions:
cmp.ac/classifying

(7-8, 9-10) Students learn about Voroni algorithms and how they are used to determine with certainty the shortest distance to key locations on a map:
cmp.ac/stores

For teachers creating a portfolio of learning or considering this task for assessment

Understanding that the graphs represent real world situations is important. Have students come up with their own scenarios and create graphs that correspond.

Or, present students with maps containing different nodes (houses, hospitals, bus stops, train stations) and ask them to construct graphs of their connections. For example, the graph of all the bus stop locations in a suburb and the roads between them.

To test understanding of the Eulerian circuit and path rule from puzzle 3, present students with graphs that do not have circuits or paths and ask them to adjust the graph so that the circuit becomes possible. Ask them to explain the change they made in relation to the rule.

Digital Technologies @ Home
Unplugged activities for students

This activity is for: Years 7-10

Convenient Stores

With thanks to the Girl's Programming Network for providing this content.

This activity teaches…

Technology allows us to innovate and explore ideas without having to physically build them. In this activity we explore the placement of grocery stores and how an algorithm known as *Voronoi Diagrams* can help us determine which grocery stores are the most convenient to travel to.

We'll learn how to create Voronoi Diagrams and test that they are correct. We will also explore how these diagrams can address people's needs for grocery stores and other physical locations, and how the diagrams can fall short.

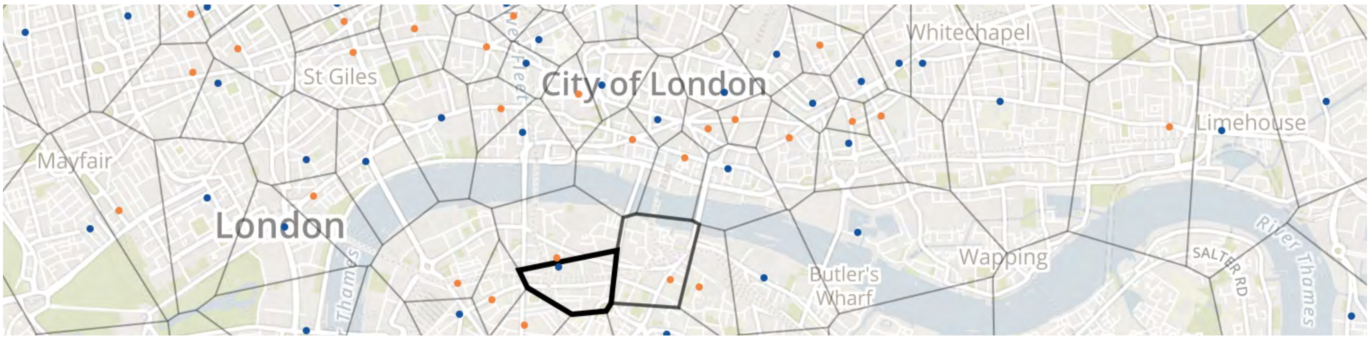
This activity is targeted towards secondary students and is expected to take 1.5 hours, including some discussion time.

You will need…

A pencil, a pen, a ruler

Voronoi Groceries

Which is the closest grocery store?



Voronoi diagram by Keir Clarke for http://googlemapsmania.blogspot.com/2014/06/the-supermarket-voronoi-map.html Map by Mapbox and OpenStreetMap

Voronoi Diagrams

The picture above shows grocery stores as dots on a map. Each grocery store is surrounded by a boundary (known as a *cell*). The cell shows which households are closest to that grocery store. That is, for any house inside a cell, no other grocery store outside the cell is closer.

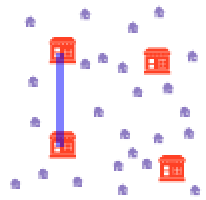
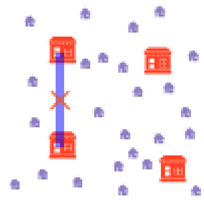
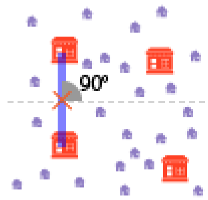
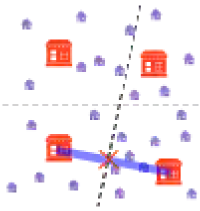
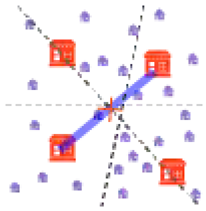
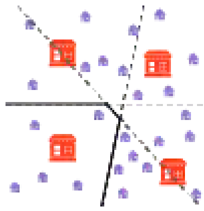
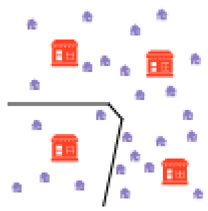
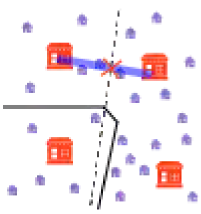
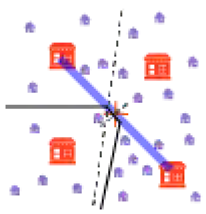
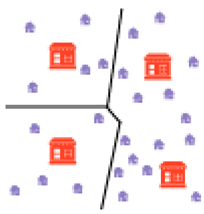
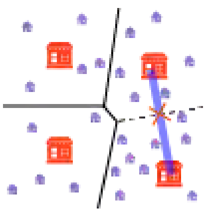
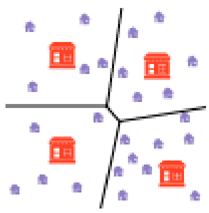
We can make our own Voronoi diagrams using a simple algorithm.

```
For each grocery store
      For each other grocery store
            Draw the boundary that is halfway
              Between the grocery store and the other grocery store
      For each boundary
            Trim the sections of the boundary
              that fall on the side of another boundary
              that is further from grocery store

      (This is known as a cell)
```

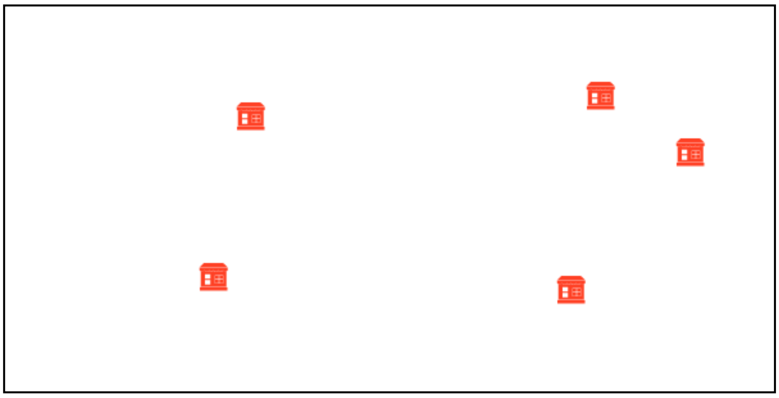
There is an example of the algorithm in action on the next page.

Here's an example of the algorithm in action:

 1) Start with a map	 2) Place a ruler between two grocery stores	 3) Find the midpoint between the two stores	 4) Draw a guideline perpendicular to the ruler
 5) Repeat for one of the previous grocery stores plus a new store	 6) Repeat with the same store until you have a line for all of the other stores	 7) Trim the lines by drawing the sections of the guidelines closest to the store	 8) This **cell** is complete!
 9) Repeat the algorithm for another grocery store	 10) Some lines fall outside of the cell boundaries You don't need to draw them	 11) Another cell done!	*Tip: Three edges will always meet at a single point* *Tip: Cells will always be convex (no indents in the cell's boundaries)*
 12) Repeat for the final cell	 13) Complete!	*You can now see which houses are closest to each grocery store!*	

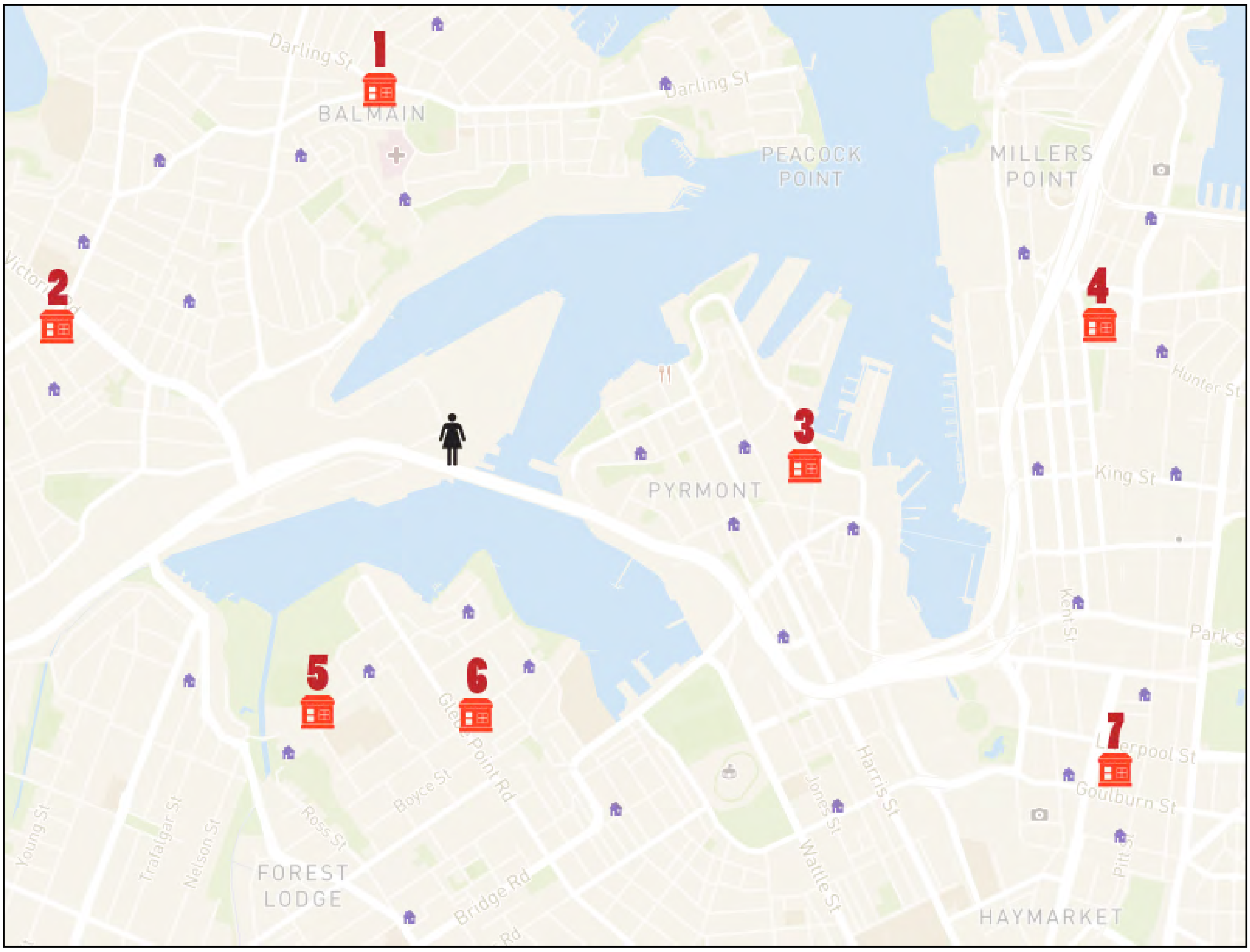
Draw your first Voronoi diagram

Using the steps on the previous page, draw the voronoi cells for this abstract map.



Now use a real map

Now draw the voronoi cells for this real world map.



Questions

How many houses are in each cell?

Cell number	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Number of houses							

Use a ruler to measure the distance directly between the person and each grocery store.

Grocery store	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Distance from the person							

Which grocery store is closest to the person? Is the person inside of the cell for that grocery store?

________________ _________________

Can you find any points on the map which are closer (straight line distance) to a grocery store outside of their cell?

____

Can you think of any other tests to make sure you executed the Voronoi algorithm correctly?

If you had to drive from the person to the grocery store which grocery store would be the closest?

____

Impact

How could you use a voronoi diagram to decide where to place a new grocery store to minimize the number of households per store?

The voronoi cells show us the closest grocery stores “as the crow flies”, but people drive, walk or take public transport to stores. How are these different modes of travel affected by the difference between the voronoi cells and reality?

What are some other reasons you might not want to go to the closest grocery store?

Custom map

Choose a place and create a new voronoi diagram for it.
It can be your local area or somewhere completely different.
You can place the paper over a screen and trace the map, print a map out or draw a rough map from memory.

Draw the locations for the Voronoi diagram on the map.
You can choose grocery stores, or a new kind of location, such as parks, fast food restaurants or hospitals.

Predict the closest locations for a few different points of interest (e.g., your house) on the map.

Draw the voronoi diagram for your points of interest.

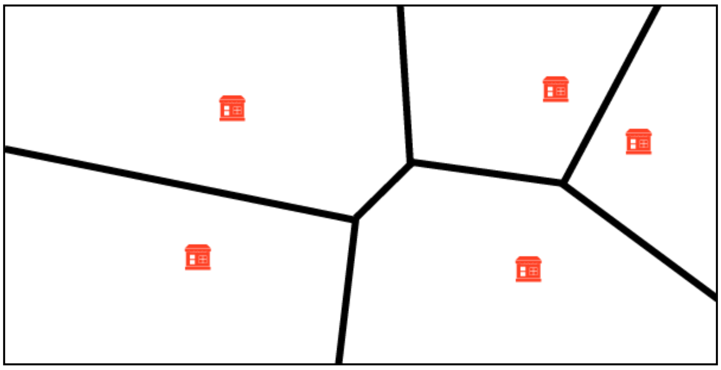
Questions

- Are the boundaries for the closest locations where you expected?
- Have you ever been in a point of interest on the map and gone to a location that wasn't the closest? Why?
- If you were going to add another location on the map (e.g., another grocery store or park) where would you put it? Would a Voronoi diagram help you make your decision?
- Could you use the voronoi diagram to choose a layout of locations which uses less resources?

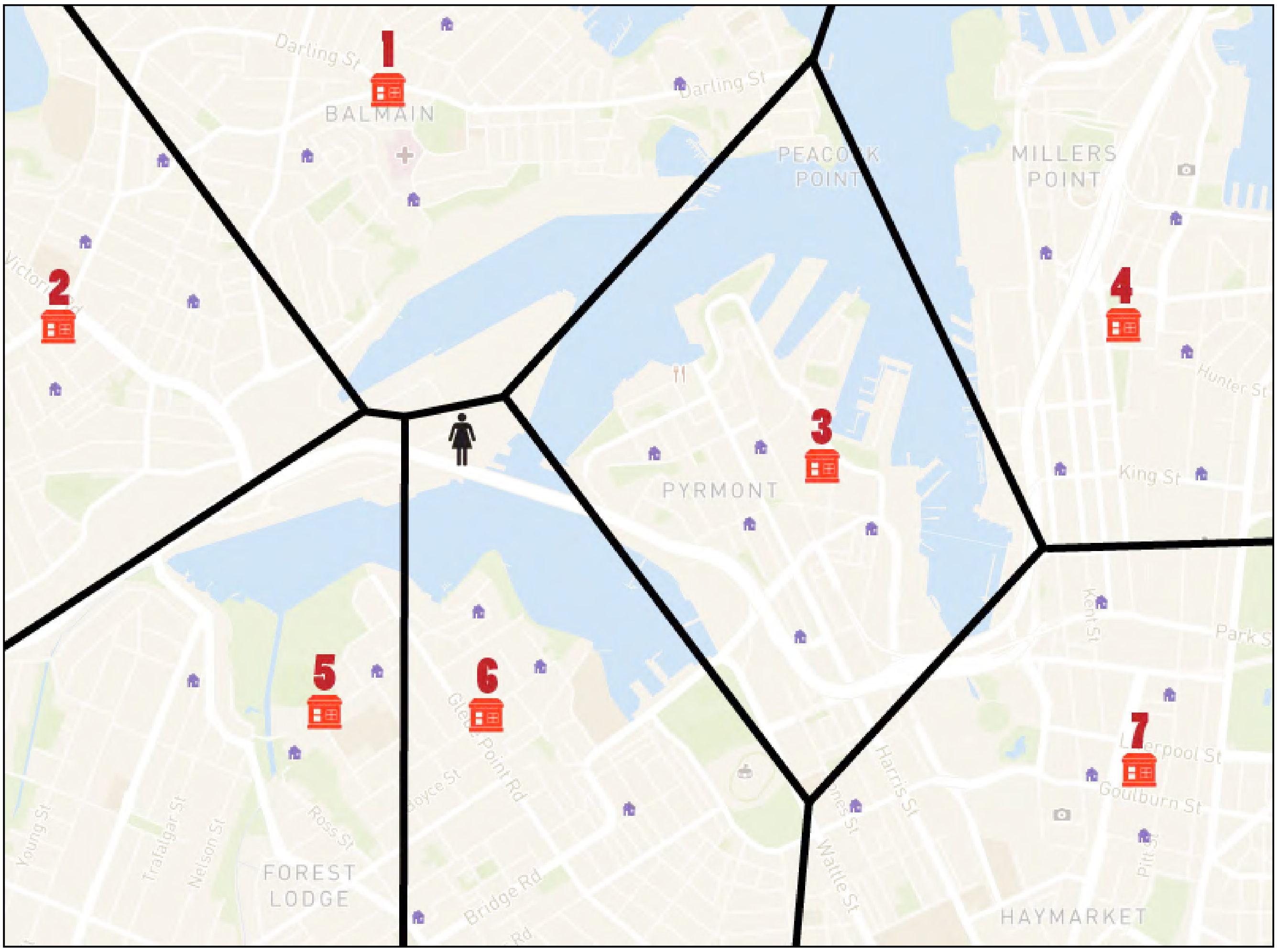
Answer key

Print this for your students or keep it to yourself!

Draw your first Voronoi diagram



Now use a real map



Answers to questions

How many houses are in each cell?

Cell number	1	2	3	4	5	6	7
Number of people	4	4	5	5	3	4	5

Use a ruler to measure the distance directly between the person and each grocery store.

Grocery store	1	2	3	4	5	6	7
Distance from the person	4 cm	5 cm	4 cm	8 cm	3.5 cm	3 cm	8.5 cm

Which grocery store is the closest to the person? Is the person inside of the cell for that grocery store?

6 Yes

Can you find any points on the map which are closer to a grocery store outside of their cell?

No

Can you think of any other tests to make sure you executed the Voronoi algorithm correctly?

Check every point of interest on the map! (This isn't 100% proof we've executed the algorithm correctly because there are infinite possible points)

Between three locations the lines should all meet at a single point. This is because there is always a point equidistant between three locations (unless the 3 locations are in a line).

The shape of the cells should be convex (no sharp angles). (Think about bubbles expanding and forming boundaries between each other to get an intuition for why this occurs)

Every point on a line should be equidistant to two points.

If you had to drive from the person to the grocery store which grocery store would be the closest?

2 or 3

Impact answers

The answers below are indicative of the kinds of answers students may give.
There may be other, equally valid ideas the students explore!

How could you use a voronoi diagram to decide where to place a new grocery store?

You could place a new point on the map where stores seem to have too many people. Then you can generate a new Voronoi diagram to see how many people each grocery store has.Then if some stores still have too many people, you could move the new grocery stores closer to the over-assigned grocery stores.

The voronoi cells show us the closest grocery stores "as the crow flies", but people drive, walk or take public transport to stores. How are these different modes of travel affected by the difference between the voronoi cells and reality?

People who drive are affected by high traffic areas. Certain short distances may actually take a long time in traffic..

People who walk cannot walk on highways or across water.

People who take public transport can only go to certain locations, so some physically close stores may take a longer travel time.

What are some other reasons you might not want to go to the closest grocery store?
The closest grocery store may be too expensive or not have the products you desire.

There may be a grocery store close to where you work or go to school.

You may order your groceries online.

Want more?

Here are some further activities, online resources, assessment ideas and curriculum references.

Adapting this activity

Older students in Years 11 and 12 may want to try and implement this algorithm using a programming language. They will need to have the requisite geometry knowledge to calculate the midpoint and intersection of lines.

Keep the conversation going

- The Voronoi diagram algorithm assumes that all grocery stores are equally desirable. How could you adapt the algorithm to account for some grocery stores being worth extra travel time?
- What are the costs (financial, environmental, logistical) associated with travelling a greater/smaller distance to the grocery store?
- What is an innovative product you could create using Voronoi diagrams?
    - Who is the audience?
    - Do they need to see the diagrams or just know the results?

Keep learning

For year 7-8 students who would like to learn basic geometry using algorithms, try this DT Challenge: cmp.ac/geometry

For students interested in learning about Voronoi diagrams in an artistic context you can watch Khan Academy's video about their use in Pixar's movies: https://www.youtube.com/watch?v=Q804hv73L6U

You can use the interactive Voronoi Diagram tool from the Statistics Online Computational Resource: https://socr.umich.edu/HTML5/others/Voronoi_App/

For teachers creating a portfolio of learning or considering this task for assessment

Ask the students to use the custom map question for their portfolios. Use the associated questions or create your own to add as explanation to their portfolios.

Linking it back to the Australian Curriculum: Digital Technologies

The content descriptors below are for years 7-10. This activity covers a single example of tracing algorithms to predict errors, using test cases, and evaluating information systems to take account of future risks (see ***bold text***).
x
In order to meet curriculum requirements, students should apply the investigations used in this activity to their own algorithms and hypothetical solutions and other, additional algorithms and real world scenarios.

Algorithms
Design algorithms represented diagrammatically and in English, and ***trace algorithms to predict output for a given input and to identify errors*** (ACTDIP040 - see cmp.ac/algorithms)

Impact
Evaluate how student solutions and ***existing information systems meet needs***, are innovative, and take account of future risks and sustainability (ACTDIP031 - see cmp.ac/impact)

Refer to aca.edu.au/curriculum for more curriculum information.

Digital Technologies @ Home
Unplugged activities for students

This activity is for: Years 7-10

TV Torment

This activity teaches…

How we interact with technology has a major impact on both our likelihood to continue using it and how widespread its adoption becomes. In this activity, we ask students to compare two different remote control devices, and ask them to reflect on which aspects of the designs are appealing.

We'll learn a little about what makes an interface intuitive and enjoyable to use, and how poor design decisions can turn users away from a product or application.

This activity is targeted towards secondary students and is expected to take 1 - 1.5 hours.

You will need…

A pencil and some paper.

TV Torment

Elements of User Experience (UX)

When we talk about the user experience of a system or product, we're interested in three main things:

- how easy it is to use;
- how it looks; and
- how it makes you feel, or "the joy of use".

Sometimes it can be hard to make all of these things work together to create a "perfect" experience for every user. This could be because there are some typical conventions that impact the design, the devices you're using require compromises to be made, or it may be because different types of users have different needs. It turns out designing interfaces to digital systems is harder that you think!

To introduce these ideas and get you thinking about what UX means, let's consider a device you interact with every day - the humble remote control!

Example 1: Your typical TV remote control

The image shown is an example of a remote control that might come with your television. Although each brand differs slightly, every remote has a similar combination of buttons in some variation of this layout.

There are a lot of buttons on this remote control, but most of them are rarely used. This can be confusing for some people, especially those that aren't very comfortable with technology.

The inclusion of lots of buttons also means that the buttons themselves are quite small, which can make them hard to press for people with limited motor function or large fingers.

Some of the buttons also have mysterious labels. What do the A, B, C and D mean on the coloured buttons? And do you know what all of the other symbols mean?

Example 2: Apple's Siri Remote

A lot of people regard Apple as a company that creates products that look great and have excellent usability. There's no doubt the Siri remote is simpler than the TV remote - it only has 6 buttons on it! It's also much smaller, making it easier to hold than the larger TV remote control. But let's think about some problems with this design.

Did you know the top section of the remote around the Menu and Home buttons is actually a touchpad? It's a convenient way to navigate menus, but an accidental slide of the finger can easily rewind, fast forward or otherwise interrupt your viewing.

You may also notice that with the buttons in the middle, both ends of the remote look similar. One of the common complaints people have of this remote is that it's easy to pick up the wrong way around, and that means the orientation of the buttons isn't what you expect, and the touchpad ends up in your palm and not at your fingertips.

Another thing that people criticise is the lack of colour - an all black remote means the buttons blend together, and it's harder to find the remote in the dark when you're watching your TV at night.

Addressing design problems

These problems aren't new - as the power of our TVs has improved and new features are added, the complexity of our remote controls has increased. Because many people find a lot of the new buttons unnecessary, some of them have made simple, home-made alterations to their remote controls to help their less technically-savvy relatives understand the main features.

You can see in this example that by using paper to cover up the buttons that aren't needed and writing a few simple instructions to help the user, it becomes clear just how few features are actually needed for this person's normal use case. Wouldn't it be great if, instead of this home-made change, the remote control just did what it needed to do out-of-the-box?

Designing for users

Different users will have different needs, and that makes designing a single product for everyone a difficult problem. One way to help you come up with a universal solution is to come up with lots of different designs that target a specific group of users.

In the boxes below, come up with your own design for a TV remote control that would be ideal for the identified group of users. Think carefully about how each group users their TV, and the devices they have connected to them.

Group 1: People like you

Group 2: People who only watch free-to-air

Group 3: People with limited fine-motor skills

Group 3: People with limited tech knowledge

Bringing your ideas together

Now that you've had a chance to think of the ways different people might use a remote control, the next step is to consider how you can take the most important aspects of each of your designs and combine them to create a design that works for as many people as possible. Answer each of the questions below, and then use these answers and the designs you did on the previous page to come up with your final design for a remote control.

List all of the common features in your designs that are important for ***all*** users.

____

____

____

____

____

____

Are there things necessary for the TV to work that aren't used often but must be included? What could you do to keep them separate from the main features?

____

____

____

____

To keep the remote control as useful as possible to most people, which features are you leaving out? How could those people access those features in a different way? Does this require you to add something new?

____

____

____

____

Final Design:

Want more?

Here are some further activities, online resources, assessment ideas and curriculum references.

Adapting this activity

Once students understand how to complete this activity, ask them to consider similar issues for a more complicated system or application. Consider a website or app they are familiar with, or how different devices change the user experience (e.g. phone vs computer).

For younger students, you can ask them to generate a single design for a specific audience e.g. thinking about how Grandma uses her TV and what buttons she needs on her remote control.

Keep the conversation going

- How did you determine which features were the most important?
- What compromises were made?
- Can you think of a different audience that would further alter your design (e.g. blind people)?
- Most TVs also have physical buttons - how does the UX of these compare to that of the remote control?
- Some remote controls have accelerometers and can be used like a wand. If you've used one of these, what do you think of that user experience? Who is it good for? What does it allow that buttons don't?
- Is there anything you do regularly when interacting with your TV that isn't suited to a remote control (e.g. text entry)? What would be a better alternative?

Keep learning

For High School students interested in learning more about user experience design, try some of these websites:

- https://www.uxmatters.com/mt/archives/2017/03/3-kinds-of-simplicity.php
- https://www.springboard.com/blog/ux-design-principles/

For teachers creating a portfolio of learning or considering this task for assessment

This task can be used as is, or you can require students to undertake a more thorough evaluation of their original four designs, and write a solution brief explaining how their final design satisfied the needs of the majority of users. They should also acknowledge any of the compromises they were forced to make.

Linking it back to the Australian Curriculum: Digital Technologies

The content descriptors below are for years 7-10. This activity covers user experience design, specifically the generation, evaluation and communication of alternative designs.

In Years 9-10, students should determine more formal criteria for evaluating the effectiveness of their designs. The criteria for evaluation should be developed with external stakeholders the represent the breadth of the users targeted by their solution.

Interactions
Design the user experience of a digital system, generating, evaluating and communicating alternative designs (ACTDIP028 - see cmp.ac/intcomp)

Refer to aca.edu.au/curriculum for more curriculum information.

This page is for

Teachers

Parents and carers

Digital Technologies @ Home
Unplugged activities for students

This activity is for: Years 9-10

LAN Party

This activity teaches...

- Subnet masks
- Basic local network configuration

The activity teaches students how IP addresses - the addresses that identify computers - determine the network a computer belongs to. It explains how a subnet mask is used to identify which part of the address belongs to the network, and which part identifies the computer itself. It also describes how a CIDR number can be used as a shorthand for writing the subnet of a network.

It is targeted towards secondary students in years 9 and 10 and addresses the expectations of the Australian Curriculum at that level. It assumes students have learned the content in the Year 7 and 8 band of the Digital Technologies curriculum, and understand both what an IP address is, and basic binary operations.

It should take about 1 hour.

You will need...

Pen and paper

Getting started (read this with your child):

Understanding how local networks are configured is important, from setting up a home network, to running an enterprise system. The fundamental knowledge remains the same, and central to that is understanding who can communicate with others in a local network.

Step by step

Read the notes and answer the questions below.

LAN Party

Understanding subnet masks - who is visible on your network

LAN and subnet masks

Computers, smartphones, tablets and many other devices communicate in networks. The network you have at home is called a local area network (LAN), and it will look something like this:

So we can identify each device uniquely, it has an address assigned to it, just like different houses have addresses on the same street. These addresses are called IP addresses, because they use the Internet Protocol (IP). Because home networks are quite small, they use the IPv4 (version 4) protocol, and addresses using IPv4 look something like this:

`192.168.0.1`

The IP address is a series of 4 numbers between 0 and 255, and each can be represented as an 8 bit number.

Although it may not be obvious this address is made up of two parts - one part describes the network, and the other identifies the host. A host is a device on a network, like your router, laptop, phone, wireless printer or even a smart LED like the Philips Hue.

The subnet mask on a typical LAN is usually `255.255.255.0` If we convert the subnet mask into binary, we see something interesting:

Subnet mask	`255.255.255.0`
Subnet mask (binary)	`1111 1111 . 1111 1111 . 1111 1111 . 0000 0000`
Network/hosts	`|------------network------------|   |-hosts-|`

When we convert to binary, the network part is always at the front and is represented as a series of ones. The remaining part - the zeros, is set aside for the hosts. ***Once a zero appears in the subnet mask, no ones can be added after it***! The subnet mask dictates which devices are *visible* to other devices. It also dictates what addresses a ***broadcast*** message will go to (all hosts in the network).

Reserved addresses - broadcasting and the network address

*In this scheme, a host can have addresses from **1 to 254**. The number 255 is used for broadcasting a message to everyone, and 0 is reserved for describing the network without specifying a host.*

When we combine this info with the IP address (**192.168.0.1**) , it tells us the network address and the possible range of IP addresses that can be given out to devices on the same network.

How does a device know what other hosts are visible

A host knows which other hosts it can communicate with by performing the AND operation between the destination IP address and the subnet mask to find the network address..

<table>
<tr><td>Subnet mask (binary)</td><td>1111 1111 . 1111 1111 . 1111 1111 . 0000 0000</td></tr>
<tr><td>Network/hosts</td><td>|------------network------------| |-hosts-|</td></tr>
<tr><td>IP address (binary)</td><td>1100 0000 . 1010 1000 . 0000 0000 . 0000 0001</td></tr>
<tr><td>Network address (binary)</td><td>1100 0000 . 1010 1000 . 0000 0000 . 0000 0000</td></tr>
<tr><td>Network address</td><td>192.168.0.0</td></tr>
<tr><td>Host range</td><td>192.168.0.1 - 192.168.0.254</td></tr>
</table>

All computers that can communicate will have the same network address. Different devices will have a different *host* number.

Communicating outside the network

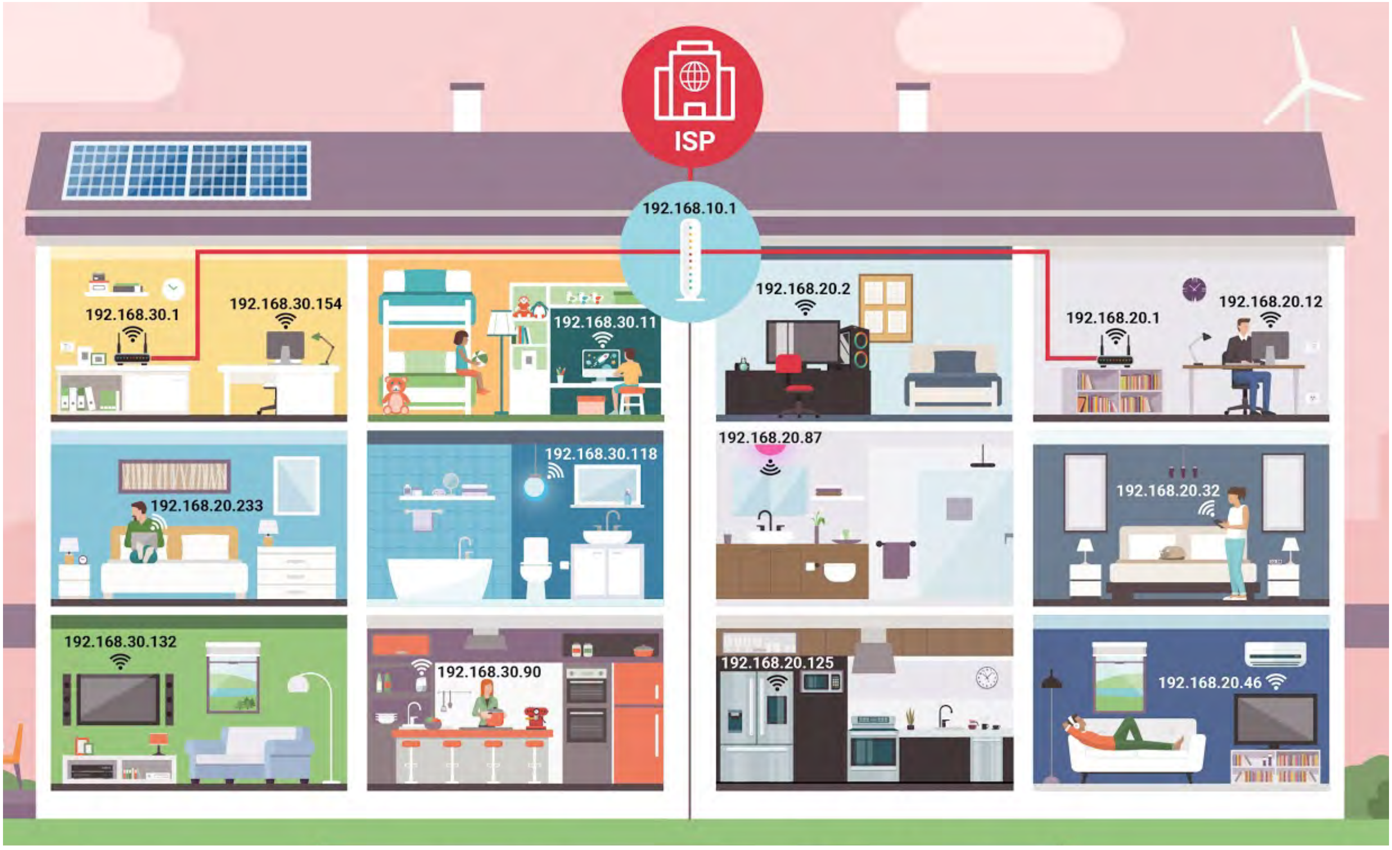
*Communicating outside the network requires a **router** to forward the packets (chunks of data) to the correct location. A host can specify the **default gateway** to be the IP address of the router, and if the destination is outside the network the packets are forwarded by the router.*

What does this have to do with subnets?

IP addresses take the general form of `<network><host>`. Sometimes you want to separate some devices on your network from others, like you might in a guest network where visitors shouldn't get access to your network drive. In this situation, you might consider breaking your network into smaller networks - or ***subnets***. When we do this, we use the last part of the `<network>` part of the address to define separate subnets.

<table>
<tr><td>Subnet addresses</td><td>192.168.0.0, 192.168.1.0, 192.168.2.0, ..., 192.168.255.0</td></tr>
<tr><td>Subnet mask (binary)</td><td>1111 1111 . 1111 1111 . 1111 1111 . 0000 0000</td></tr>
<tr><td>Network/hosts</td><td>|------network-------| |subnet-| |-hosts-|</td></tr>
<tr><td>Host ranges</td><td>192.168.X.1 - 192.168.X.254 (where X is anything from 0 - 255)</td></tr>
</table>

Let's take a look at an example network, where two townhouses are sharing an internet connection on the same network. Each device in the picture (that isn't a router) has a subnet mask of `255.255.255.0`.



Let's look at some of these devices, all of these devices in the house have the same subnet, which means they can communicate to other devices in the same network. But the devices in the separate townhouses can't communicate with each other.

	Left house - IP addresses			**Right house - IP addresses**		
Device	**Network**	**Subnet**	**Host**	**Network**	**Subnet**	**Host**
Router	`192.168`	**30**	`1`	`192.168`	**20**	`1`
Computer	`192.168`	**30**	`154`	`192.168`	**20**	`12`
Smart TV	`192.168`	**30**	`132`	`192.168`	**20**	`46`
Smart light	`192.168`	**30**	`118`	`192.168`	**20**	`87`
Google home	`192.168`	**30**	`90`			
Phone				`192.168`	**20**	`32`
Smart fridge				`192.168`	**20**	`125`

A shorthand notation

*There is a shorthand way to specify an IP address and the associated subnet masks. For example: 192.168.1.33**/24**. The "/24" indicates there are 24 1s in the subnet mask. In other words, it's saying the subnet mask is*

```
1111 1111 . 1111 1111 . 1111 1111 . 0000 0000
```

*which is our old friend **255.255.255**.0.*

Since subnet masks always start with 1s, this shorthand notation just says how many 1s there are in the subnet mask - that's it! This is called Classless Inter-Domain Routing (or CIDR) https://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing.

Questions

Question 1

The network address is `172.10.45.22/16`, what is the subnet mask?

** Find the answer in the box titled* ***A shorthand notation***

Question 2

Take a look at the image in the ***What does this have to do with subnets*** section. A pair of townhouses are sharing an internet connection, but each have their own routers and networks.

a) Why is it useful for them to be on separate networks?

____

____

** Find the answer in the section* ***What does this have to do with subnets***

b) One of the devices has a problem. It still connects to the internet, but it can't connect to the smart TV in the same house. What is the problem, and which device is it?

Problem: ____

Problem device IP address: ______________________________

** Use the image in the section* ***What does this have to do with subnets***

Question 3

`255.255.115.0` - Is this a valid subnet mask?

** Find the answer in the section* ***LAN and subnet masks***

Question 4

A device has an address of `192.160.33.5/24.` What address would you use to broadcast a message to everyone on the same subnet?

** Find the answer in the box titled* ***Reserved addresses - broadcasting and the network address***

Question 5

The printer is at `10.0.160.5/20`. Which devices can print on the printer?

** Find the answer in the section* ***What does this have to do with subnets***

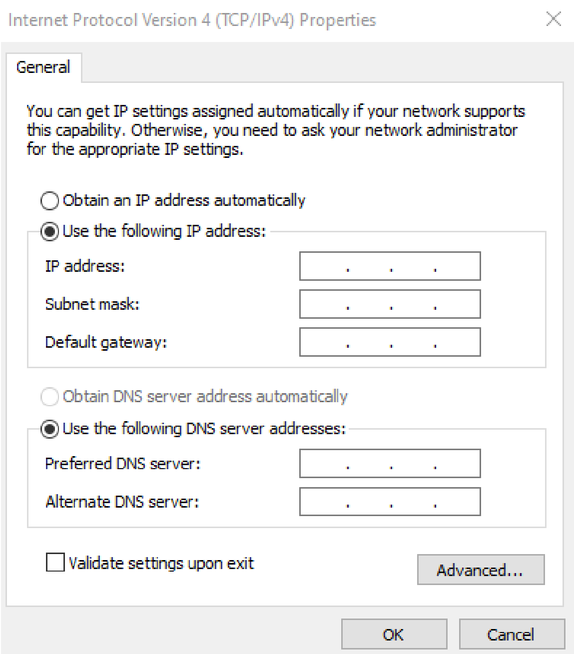
Laptop 1 - `10.0.162.6`
Laptop 2 - `10.0.172.5`
Laptop 3 - `10.10.3.165`
Laptop 4 - `10.0.159.240`
Laptop 5 - `10.0.160.25`

Question 6

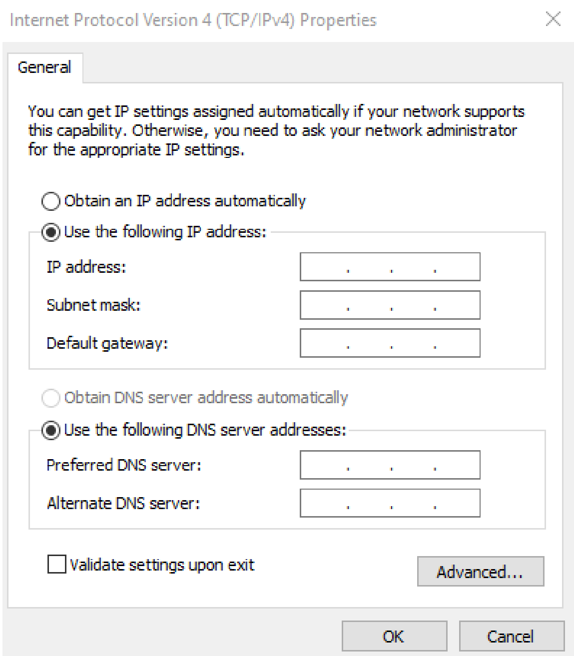
You have invited 2 other friends over for a LAN party to play games, and are configuring the 3 devices. Unfortunately you find the router you have is broken, but you still want to be able to play. Luckily, you have a network switch and can plug everything in. What do you put in each field so the devices can communicate? (You can specify your own network configuration)

Ignore the DNS settings, you don't need them.

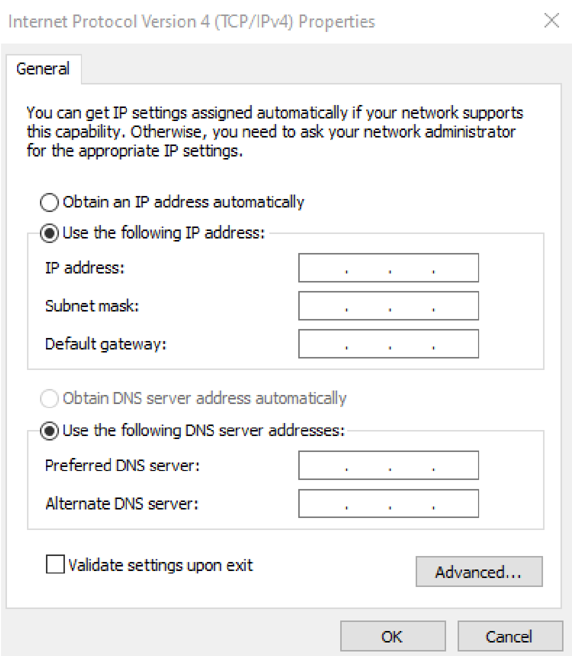
Computer 1



Computer 2



Computer 3



Answer key

Choose if you want to print this for your kids or keep it to yourself!

Answer 1

The network address is `172.10.45.22/16`
Subnet mask is:
`255.255.0.0`
Since the end is /16, we know there are 16 1s in the subnet mask.

Expanding the above into binary

```
1111 1111 . 1111 1111 . 0000 0000 . 0000 0000
   255    .    255    .     0     .     0
```

Answer 2

a) They should use separate subnets so that their devices are separated. If not, they could access their neighbors printer, smart TV, or Google home, which is undesirable.
b) The left house's laptop is configured wrong. It's on the neighbours network!
It's IP address is `192.168.20.233`

Answer 3

No - 255.255.115.0 is not a valid subnet mask. Converting it to binary it looks like this:
1111 1111 . 1111 1111 . **01**11 **001**1 . 0000 0000

Since there is a mix of 1s *after* 0s, it is not a valid subnet mask.

Answer 4

192.160.33.255 - since 255 is reserved for broadcast messages to everyone on that subnet.

Answer 5

The subnet mask becomes:
`255.255.240.0`
`11111111.11111111.11110000.00000000`
So we can see that any devices with the same first 20 digits of their IP address will be on the same network.

The printer is at:
`10.0.160.5`
`00001010.00000000.10100000.00000101`

So any device whose IP address starts with `00001010.00000000.1010` is on the same network.

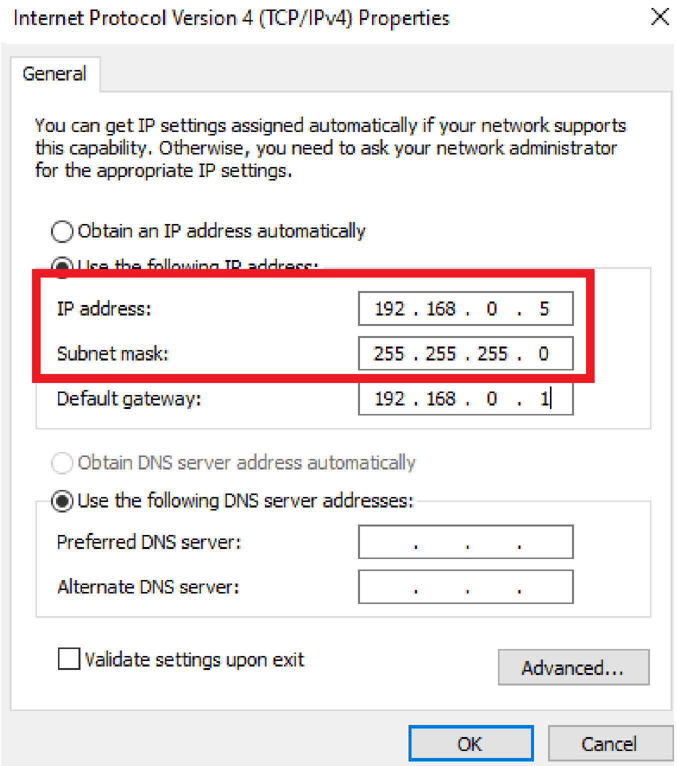
To check this, convert each address to binary, and compare their initial numbers to the printers'.

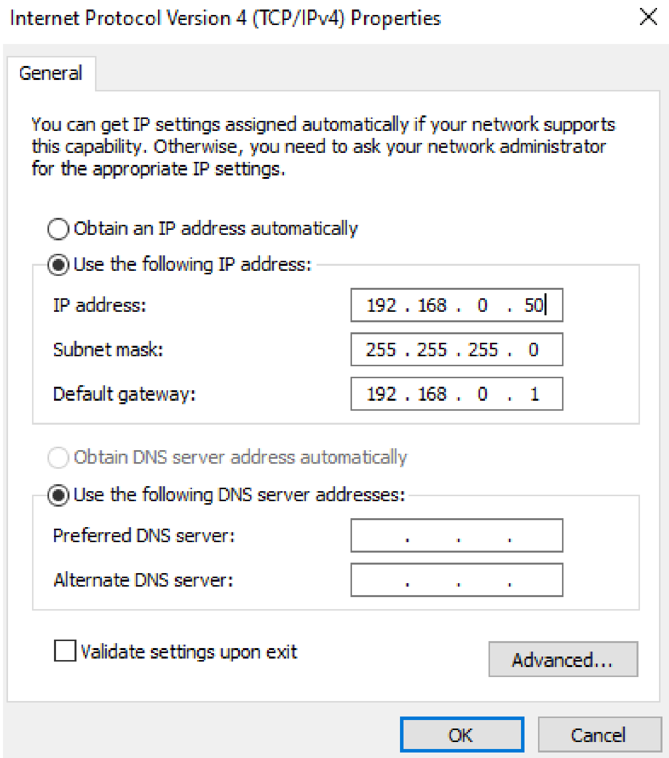
	IP Address	Binary IP Address	Matches printer?
Laptop 1	10.0.162.6	0000 1010 . 0000 0000 . 1010 0010 . 0000 0110	**Yes**
Laptop 2	10.0.172.5	0000 1010 . 0000 0000 . 1010 1100 . 0000 0101	**Yes**
Laptop 3	10.10.3.165	0000 1010 . 0000 1010 . 0000 0011 . 1010 0101	**No**
Laptop 4	10.0.159.240	0000 1010 . 0000 0000 . 1001 1111 . 1111 0000	**No**
Laptop 5	10.0.160.25	0000 1010 . 0000 0000 . 1010 0000 . 0001 1001	**Yes**

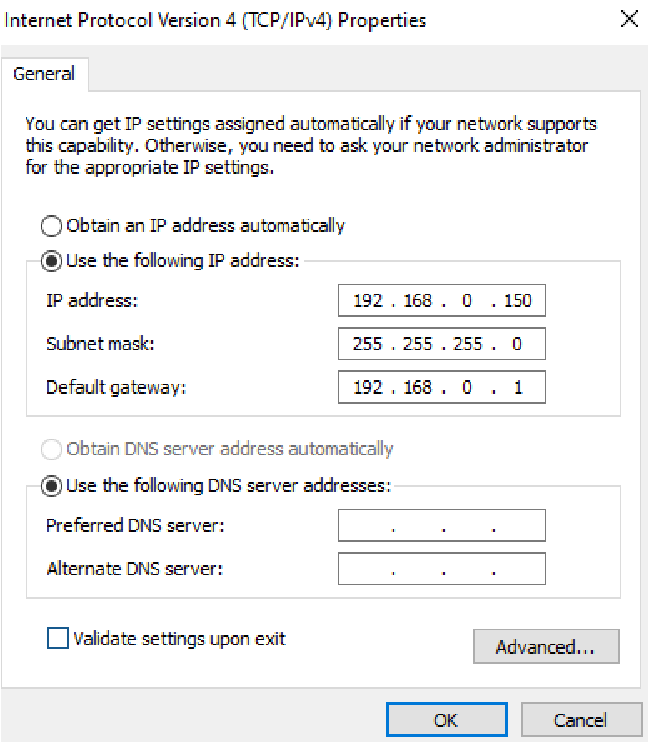
Answer 6

This is a screen that has often faced students who want to set up their LAN party. There are *millions* of different answers here. Since there are only 3 devices, we want to have a small subnet, 255.255.255.0 is a sensible default choice. Any host number between 1 and 254 will work, as long as there are *no conflicts*. You can add more bits to the mask, if you like. The IP addresses should be narrow accordingly.

The default gateway field is not used, since there is no external communication. **So you can leave that field empty**, or just put in a number that doesn't conflict like 192.168.0.1 as a default (even though it doesn't exist). Either will work.


Internet Protocol Version 4 (TCP/IPv4) Properties
General
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
Obtain an IP address automatically
Use the following IP address:
IP address: 192 . 168 . 0 . 5
Subnet mask: 255 . 255 . 255 . 0
Default gateway: 192 . 168 . 0 . 1
Obtain DNS server address automatically
Use the following DNS server addresses:
Preferred DNS server:
Alternate DNS server:
Validate settings upon exit
Advanced...
OK
Cancel


Internet Protocol Version 4 (TCP/IPv4) Properties
General
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
Obtain an IP address automatically
Use the following IP address:
IP address: 192 . 168 . 0 . 50
Subnet mask: 255 . 255 . 255 . 0
Default gateway: 192 . 168 . 0 . 1
Obtain DNS server address automatically
Use the following DNS server addresses:
Preferred DNS server:
Alternate DNS server:
Validate settings upon exit
Advanced...
OK
Cancel


Internet Protocol Version 4 (TCP/IPv4) Properties
General
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
Obtain an IP address automatically
Use the following IP address:
IP address: 192 . 168 . 0 . 150
Subnet mask: 255 . 255 . 255 . 0
Default gateway: 192 . 168 . 0 . 1
Obtain DNS server address automatically
Use the following DNS server addresses:
Preferred DNS server:
Alternate DNS server:
Validate settings upon exit
Advanced...
OK
Cancel

This page is for Teachers

Want more?

Here are some further activities, online resources, assessment ideas and curriculum references.

This is just the start of networking

Once students understand how local networks operate, you can extend into routing, how IP addresses are allocated, types of data packets and more!

There's a whole lot more to networking than what was covered here! A few topics to investigate:

1. DHCP - how devices get assigned IP addresses automatically
2. DNS - how words (like google.com) are converted into a destination IP addresses
3. TCP - what is put inside a data packet so information can be sent back and forth reliably

Keep the conversation going

- What's the purpose of subnet masks?
- What would happen if any host could communicate with a wider range of IP addresses?
- What happens if we run out of addresses?

Keep learning

For High School students interested in learning more about how computers communicate, try this course: cmp.ac/networksec

Linking it back to the Australian Curriculum: Digital Technologies

Digital Systems
Investigate the role of hardware and software in managing, controlling and securing **the movement of** and access to **data in networked digital systems**. (ACTDIK034 - see cmp.ac/systems)

Refer to aca.edu.au/curriculum for more curriculum information.