Algorithms in the Victorian Curriculum: An essential entry point into Reasoning and Problem solving

Dr Max Stephens
Melbourne Graduate School of Education
m.stephens@unimelb.edu.au

Algorithms in the new Victorian curriculum

- Algorithmic processes are a natural part of Mathematics
- Computational or Algorithmic Thinking is a critical skill for the STEM jobs of the future
- Complements and is aligned with the digital technologies curriculum

Algorithms in the new Victorian curriculum

- Algorithmic processes are a natural part of Mathematics
- Computational or Algorithmic Thinking is a critical skill for the STEM jobs of the future
- Complements and is aligned with the digital technologies curriculum

Key points:

1. Focus is on helping students to develop their algorithmic thinking
2. Focus is <u>not</u> on having students memorise algorithms
3. Embedded in the mathematics curriculum
4. <u>Not</u> the same as Coding. That refers to some specific language that can be used

- Algorithmic processes are a natural part of Mathematics
- Computational or Algorithmic Thinking is a critical skill for the STEM jobs of the future
- Complements and is aligned with the digital technologies curriculum

Algorithmic thinking is sometimes described as Computational thinking

= The kind of thinking that a computer could execute

We’re talking about the thinking behind the language.

Some key terms

Algorithm:

Well defined set of instructions designed to perform a particular task or solve a problem.

Coding:

A set of worded instructions by which algorithms are represented and implemented

Sequence:

An ordered set of elements such as numbers, instructions or objects.

Decision:

A process by which a selection or choice is made from a set of alternatives

Iteration:

The repetition of a process a specified number of times, or until a condition is satisfied

- Strand: Number and Algebra, Sub-strand: Patterns and Algebra

Level	Content description (mandatory)
F	Follow a short sequence of instructions (VCMNA077)
1	Recognise the importance of repetition of a process in solving problems (VCMNA094)
2	Apply repetition in arithmetic operations, including multiplication as repeated addition and division as repeated subtraction (VCMNA114)
3	Use a function machine and the inverse machine as a model to apply mathematical rules to numbers or shapes (VCMNA139)
4	Define a simple class of problems and solve them using an effective algorithm that involves a short sequence of steps and decisions (VCMNA164)
5	Follow a mathematical algorithm involving branching and repetition (iteration) (VCMNA194)
6	Design algorithms involving branching and iteration to solve specific classes of mathematical problems (VCMNA221)

View these as suggested starting points. Not as end points.

THE UNIVERSITY OF MELBOURNE

Prior to Year 3: Follow a short sequence of instructions

- Important that children experience the need for systematic thinking (reasoning) to solve problems that are relevant and important for them.
- Links to STEM and Digital Technologies are important.
  E.g. From Australian curriculum: Digital technologies

Foundation to Year 2 subject achievement standard

View learning area achievement standard

The parts of the achievement standard targeted in the assessment task are highlighted.

By the end of Year 2, students identify how common digital systems (hardware and software) are used to meet specific purposes. They use digital systems to represent simple patterns in data in different ways.

Students design solutions to simple problems using a sequence of steps and decisions. They collect familiar data and display them to convey meaning. They create and organise ideas and information using information systems, and share information in safe online environments.

Prior to Year 3: Follow a short sequence of instructions

- Important that children experience the need for systematic thinking (reasoning) to solve problems that are relevant and important for them.
- Links to STEM and Digital Technologies are important:

Work Sample 3
Digital project: Bee-Bot activity

Work sample summary

Students explored the concept of sequencing steps and decisions using Bee-Bots. They worked in groups to solve progressively more complex routes using grids on paper. Students then programmed the Bee-Bots to navigate grids created on the floor using masking tape. The base level was a 4 x 4 grid. Some students progressed to the more complex 8 x 10 grid of a community including a range of obstacles....

Prior to Year 3

- Important that children experience the need for systematic thinking (reasoning) to solve problems that are relevant and important for them.
- Links to STEM and Digital Technologies are important:

Work Sample 3
Digital project: Bee-Bot activity

Work sample summary

Students explored the concept of sequencing steps and decisions using Bee-Bots. They worked in groups to solve progressively more complex routes using grids on paper. Students then programmed the Bee-Bots to navigate grids created on the floor using masking tape. The base level was a 4 x 4 grid. Some students progressed to the more complex 8 x 10 grid of a community including a range of obstacles....

Codable Robots for the Primary School Classroom

Meet Bee-Bot!

Bee-Bot is an exciting new robot designed for use by young children. This colorful, easy-to-operate, and friendly little robot is a perfect tool for teaching sequencing, estimation, problem-solving, and just having fun!

Tell Bee-Bot to move forward and back and turn left and right by pressing the arrow keys. Then press GO to send Bee-Bot on its way. Bee-Bot stops, beeps, and blinks after each step so you can follow along as Bee-Bot executes your program. Bee-Bot can remember up to 40 commands and so encourages increasingly sophisticated plans and programs. Children love to teach Bee-Bot how to go from place to place and learn to plan, count, and tell left from right as they do.

Meet Sphero!

The SPRK Lightning Lab app is your hub to create, contribute and learn with Sphero Robots. The visual block-based building interface makes learning the basic principles of programming approachable and fun. Browse through activities, keep track of a class, and collaborate with users around the world. When creativity strikes... Make. Code. Share.

Google Play | App Store | amazon

Other Codable robots cont.

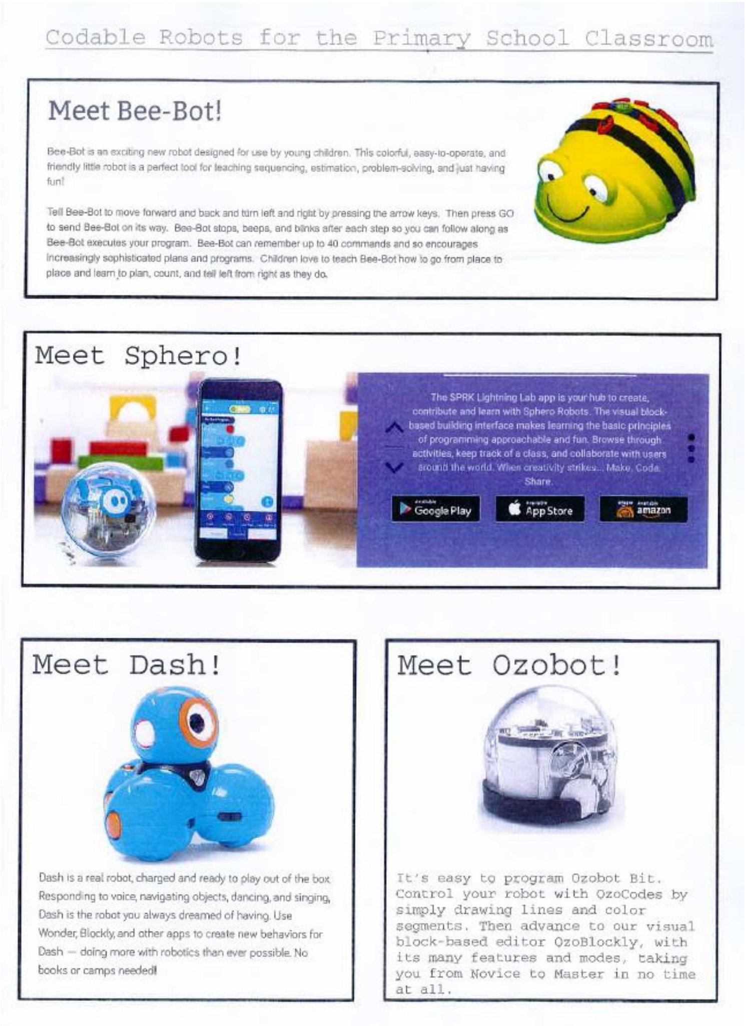
Meet Dash!

Dash is a real robot, charged and ready to play out of the box. Responding to voice, navigating objects, dancing, and singing, Dash is the robot you always dreamed of having. Use Wonder, Blockly, and other apps to create new behaviors for Dash — doing more with robotics than ever possible. No books or camps needed!

Meet Ozobot!

It's easy to program Ozobot Bit. Control your robot with OzoCodes by simply drawing lines and color segments. Then advance to our visual block-based editor OzoBlockly, with its many features and modes, taking you from Novice to Master in no time at all.

Resources: Mathspace and Codable robots



- Strand: Number and Algebra, Sub-strand: Patterns and Algebra

Level	Content description (mandatory)
F	Follow a short sequence of instructions (VCMNA077)
1	Recognise the importance of repetition of a process in solving problems (VCMNA094)
2	Apply repetition in arithmetic operations, including multiplication as repeated addition and division as repeated subtraction (VCMNA114)
3	Use a function machine and the inverse machine as a model to apply mathematical rules to numbers or shapes (VCMNA139)
4	Define a simple class of problems and solve them using an effective algorithm that involves a short sequence of steps and decisions (VCMNA164)
5	Follow a mathematical algorithm involving branching and repetition (iteration) (VCMNA194)
6	Design algorithms involving branching and iteration to solve specific classes of mathematical problems (VCMNA221)

View these as suggested starting points. Not as end points.

Example – Year 2, 3, 4

Apply repetition in arithmetic operations, including multiplication as repeated addition and division as repeated subtraction

Follow the algorithm below.

Step	Instruction	Result
1	Start with the number 54	☐
2	Add 6 to this number	☐
3	Repeat step 2 three more times	☐
		☐
		☐

From Mathspace

Example – Year 2, 3, 4

Apply repetition in arithmetic operations, including multiplication as repeated addition and division as repeated subtraction

Consider the algorithm below.

1. Start with the number 7.
2. Add 7 to this number.
3. Repeat step 2 3 more times.

a. How many 7's have been added together?

☐

b. Write this algorithm as a sum.

c. What is the final answer?

d. Calculate 7×5.

From Mathspace

Example - Year 4

Define a simple class of problems and solve them using an effective algorithm that involves a short sequence of steps and decisions

Manipulating sets of numbers using a given rule

- We teach a systematic way of doing this for whole numbers (lower primary) and for decimal numbers (upper primary). Students describe their thinking using a set of (plain language) instructions to achieve their purpose.

- Order these three digit numbers from largest to smallest:

  102, 191, 236, 499, 497, 407

- Example: Order these five decimal numbers from smallest to largest

  0.5, 0.45, 0.20, 0.03, 0.46

Manipulating sets of numbers using a given rule

- We teach a systematic way of doing this for whole numbers (lower primary) and for decimal numbers (upper primary). Students describe their thinking using a set of (plain language) instructions to achieve their purpose.

- Example: Order these five decimal numbers from smallest to largest

  0.5, 0.45, 0.20, 0.03, 0.46

Can you design and write down an effective algorithm that involves a short sequence of steps and decisions that will order these five decimal numbers from smallest to largest.

Manipulating sets of numbers using a given rule

- We teach a systematic way of doing this for whole numbers (lower primary) and for decimal numbers (upper primary). Students describe their thinking using a set of (plain language) instructions to achieve their purpose.

- Example: Order these five decimal numbers from smallest to largest

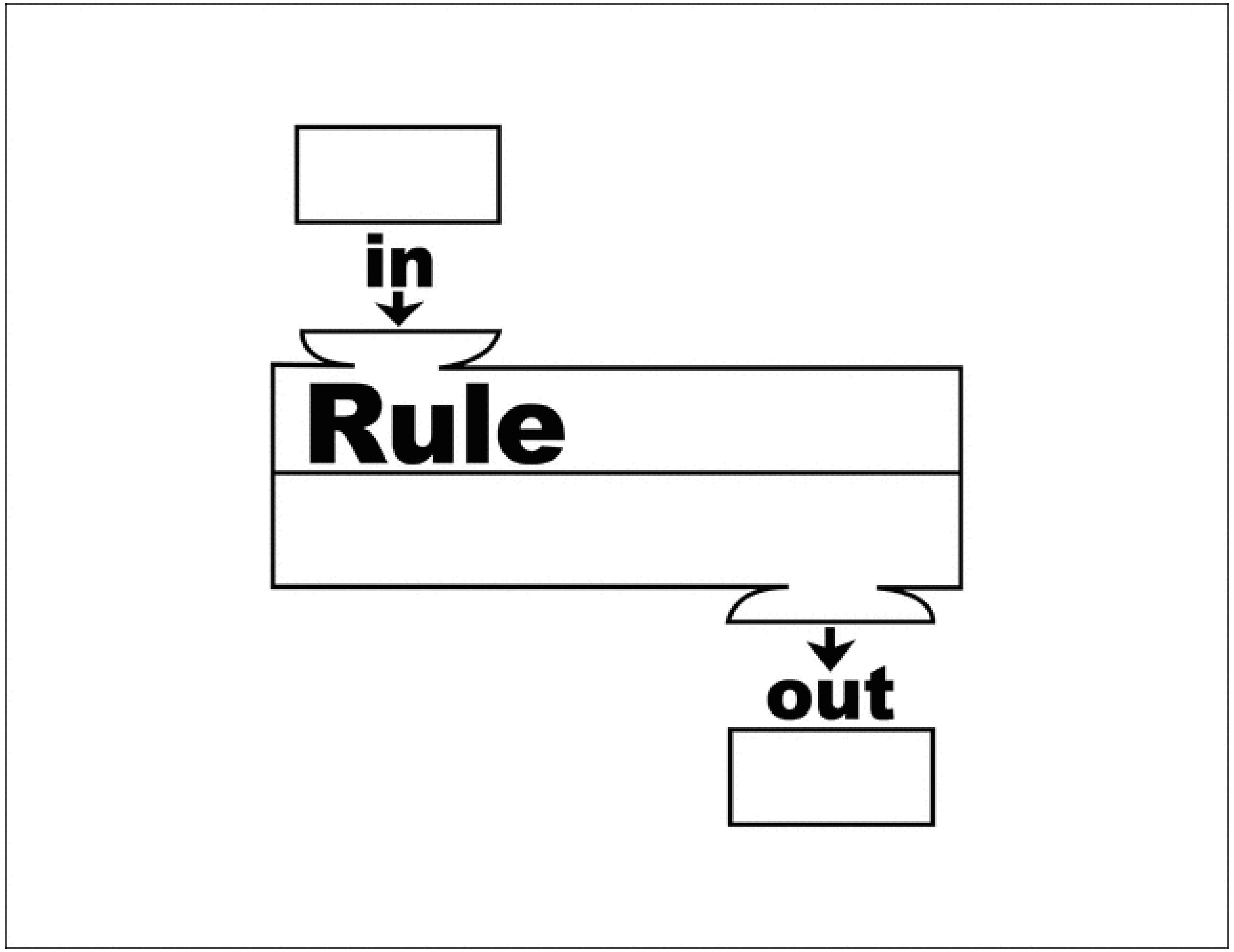
  0.5, 0.45, 0.20, 0.03, 0.46

  0.03, 0.20, 0.45, 0.46, 0.5

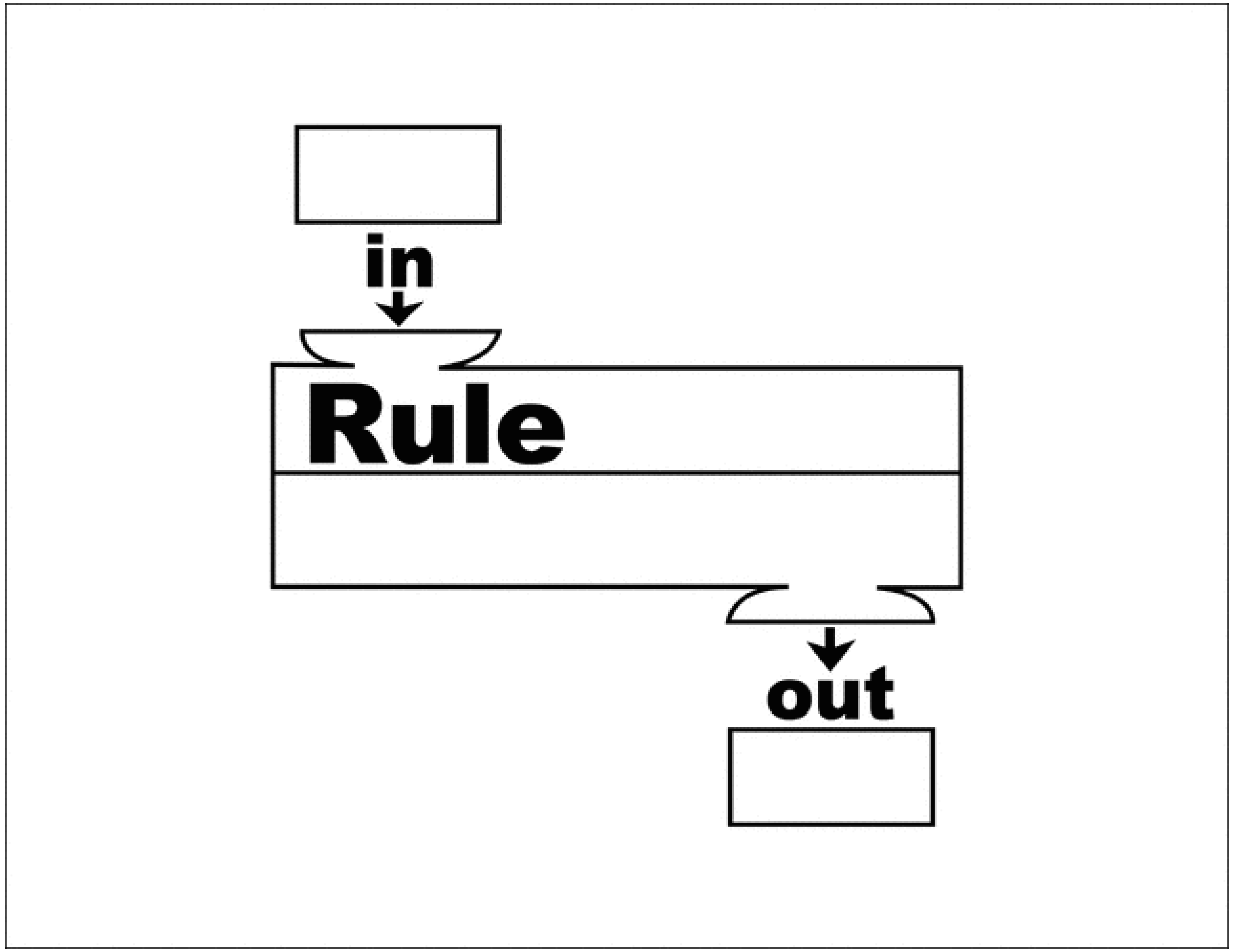
Share your written algorithm with the person next to you. Are your two algorithms the same?

Using simple function machines to represent and apply a process or the inverse process

Function machines: Year 3 and above

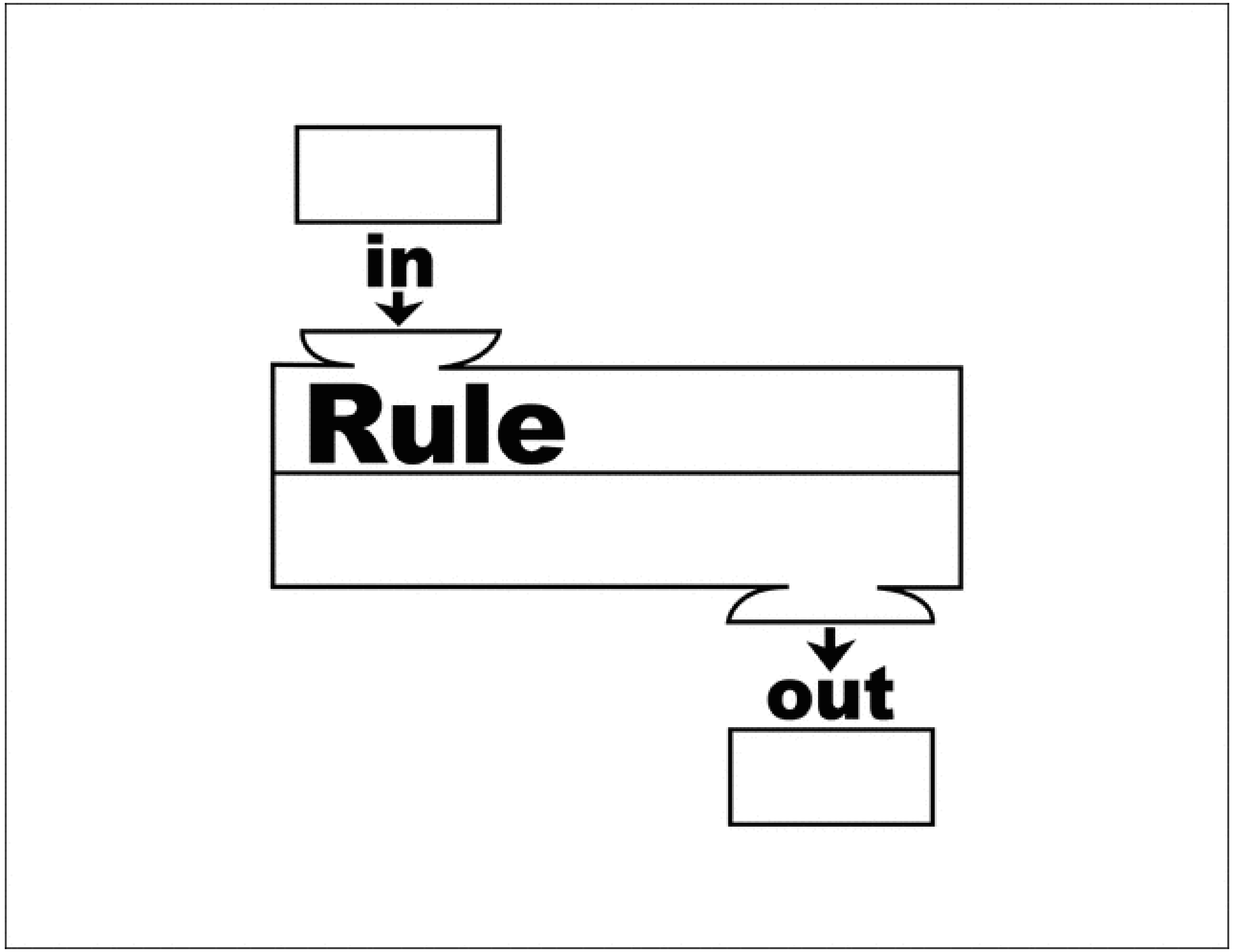


Function machines



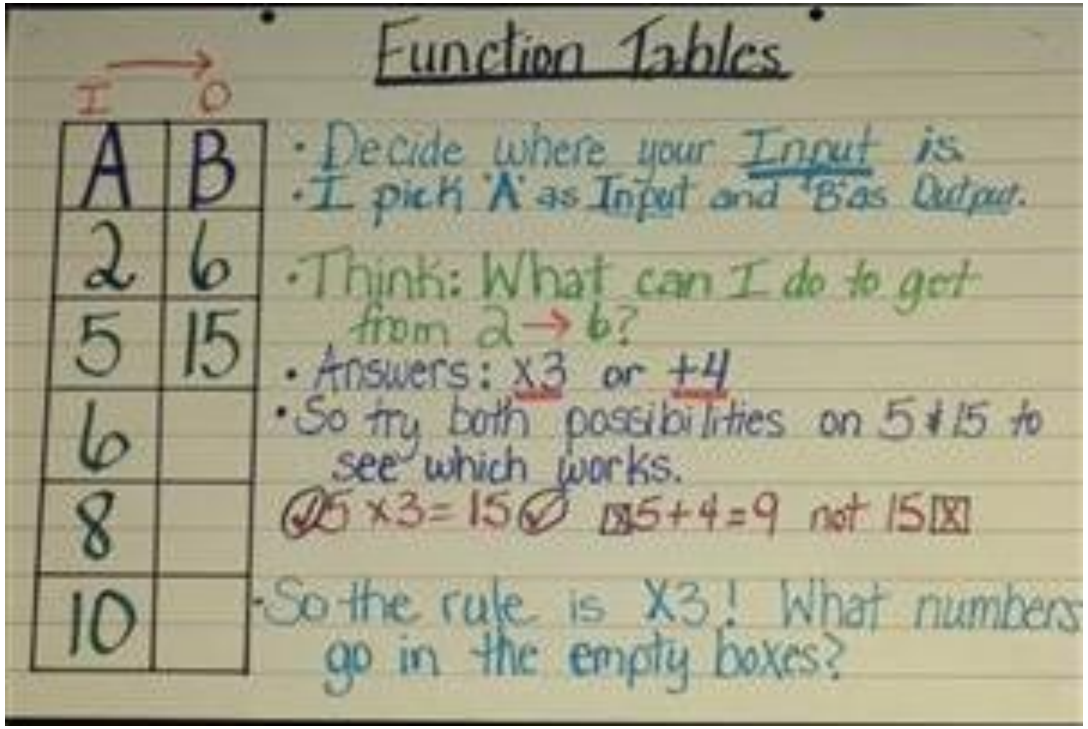
Start with an Input and the Rule. Vary the inputs and make a list of the outputs.

Function machines

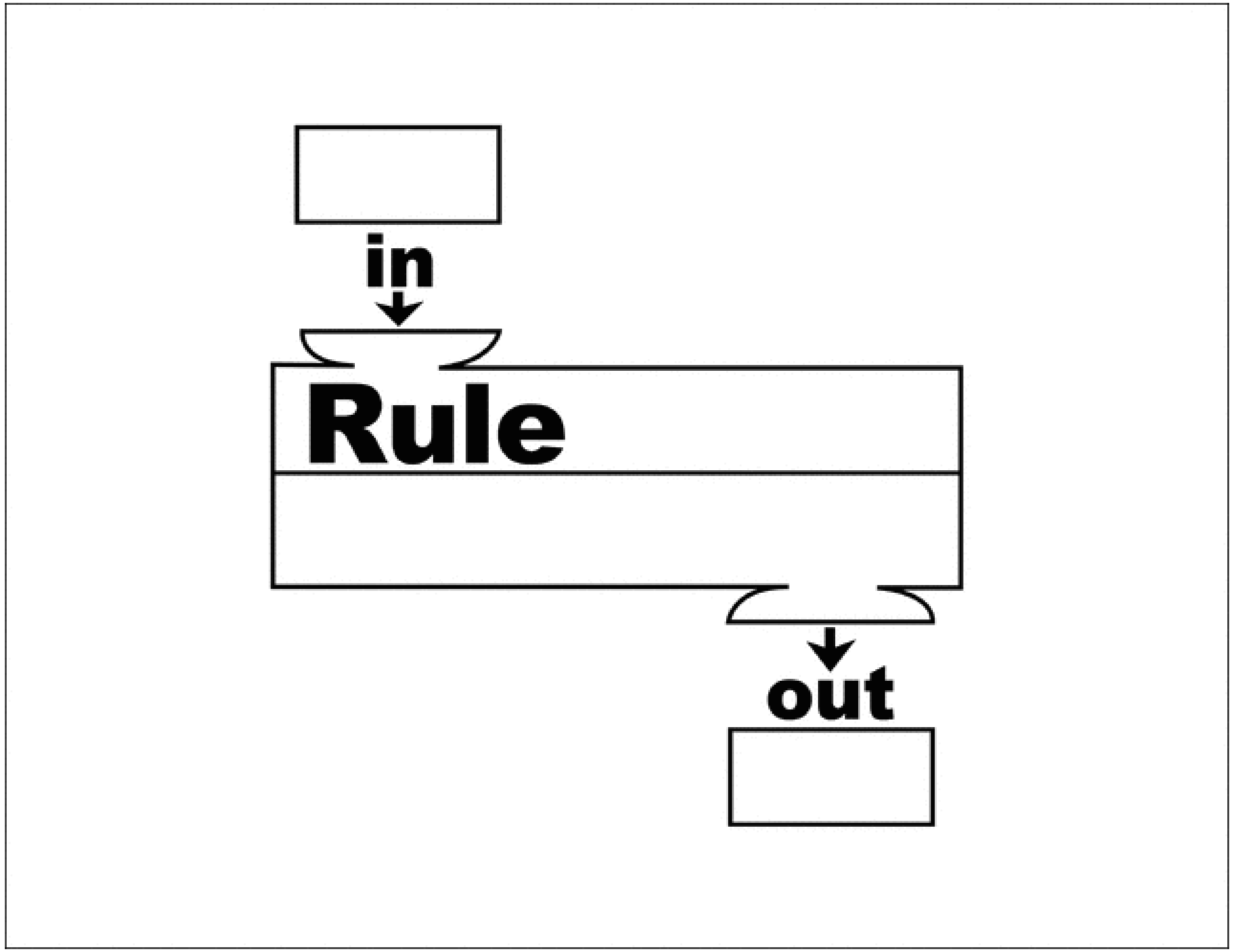


But we can also start with some inputs and outputs and have students determine the rule

Function Machine Tables: What's the rule?



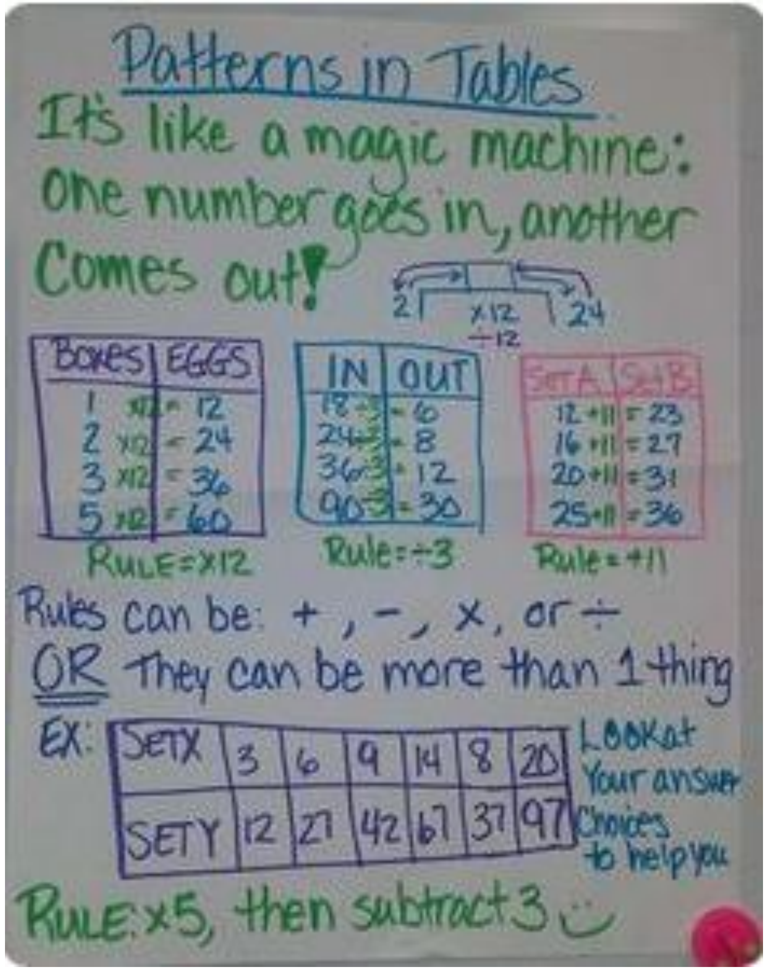
Function machines: working backwards



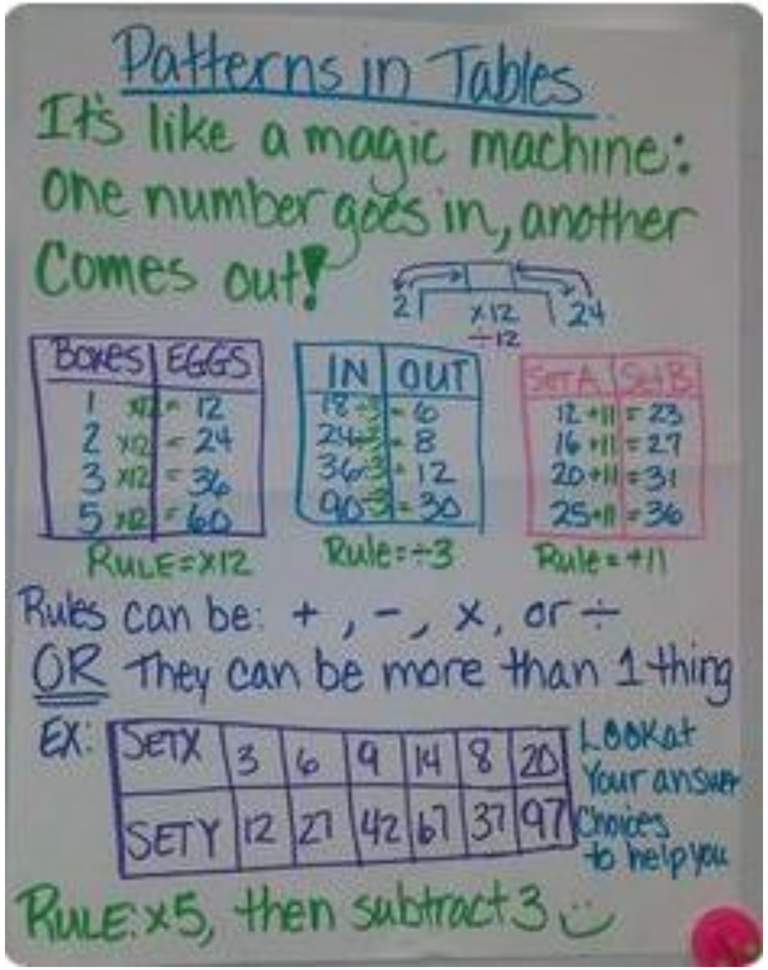
But we can also start with some outputs and the rule and ask students to work backwards. What would be the input?

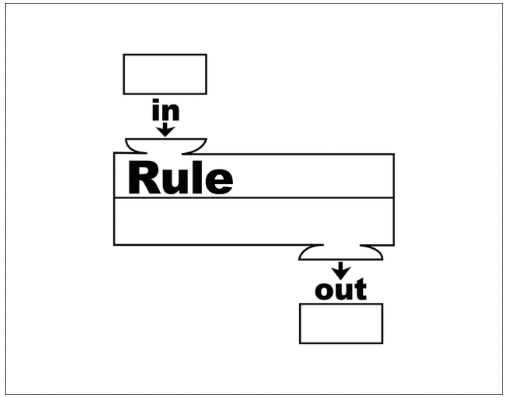
Starting in the early grades with simple one-step rules like Adding 5 and working backwards.
In later grades, the one step rules include multiplication and division.
But we can also begin to include two-step rules.

Patterns in Tables: Year 5

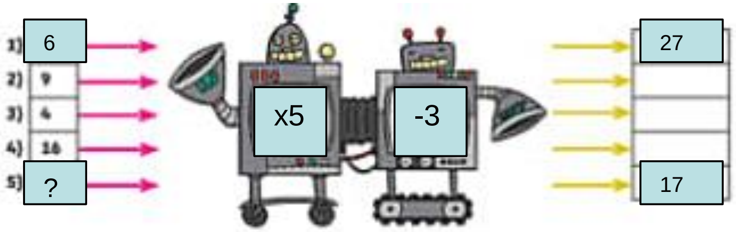


Patterns in Tables: Year 5 - Function Machines





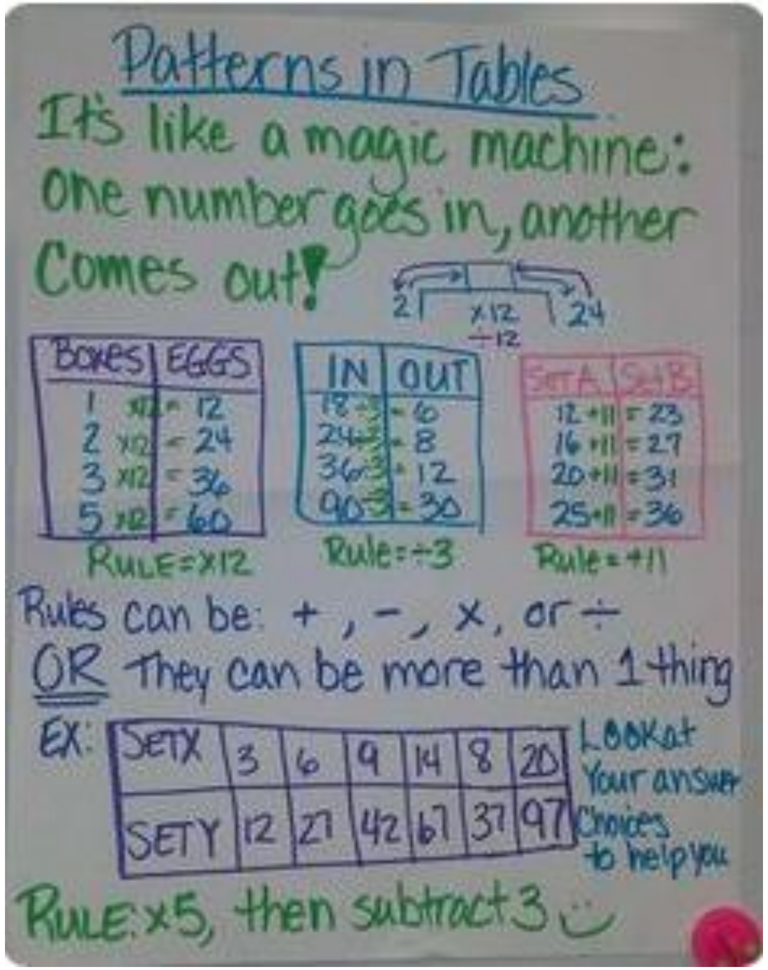
One-step function machine with a single rule:
× 12, or ÷3, or + 11

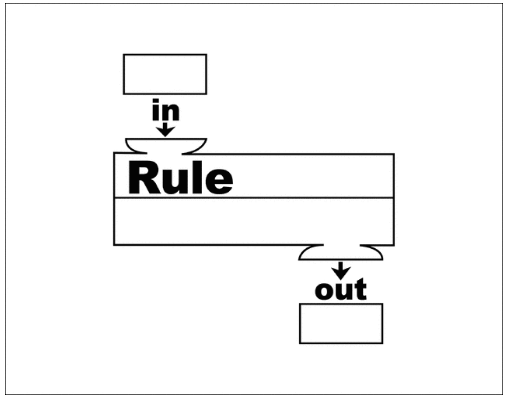


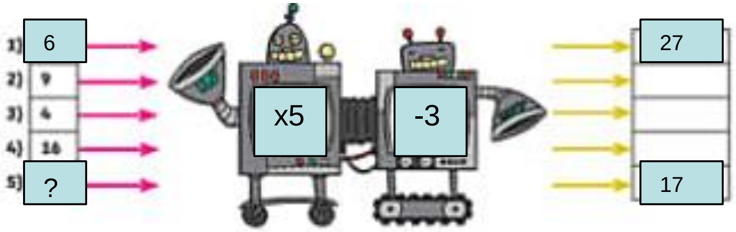
Two-step function machines where the output of the first becomes the input for the second

THE UNIVERSITY OF MELBOURNE

Patterns in Tables: Year 5 - Function Machines



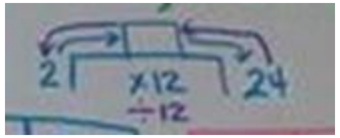




One-step function machine with a single rule:
× 12, or ÷ 3, or + 11

Two-step function machines where the output of the first becomes the input for the second

Working backwards is important



Example Year 5 and above

Follow a mathematical algorithm involving branching and repetition (iteration)

- Can start in Year 5 (or earlier) but will continue in subsequent years
- Here is an example of a branching for sorting shapes:
  - Suppose we start with a whole lot of quadrilaterals, and we want to find only the rectangles and squares
    - Can you think of a simple sorting algorithm for finding only the rectangles and squares and then sorting out these two shapes
    - What are the key decisions you need to make?

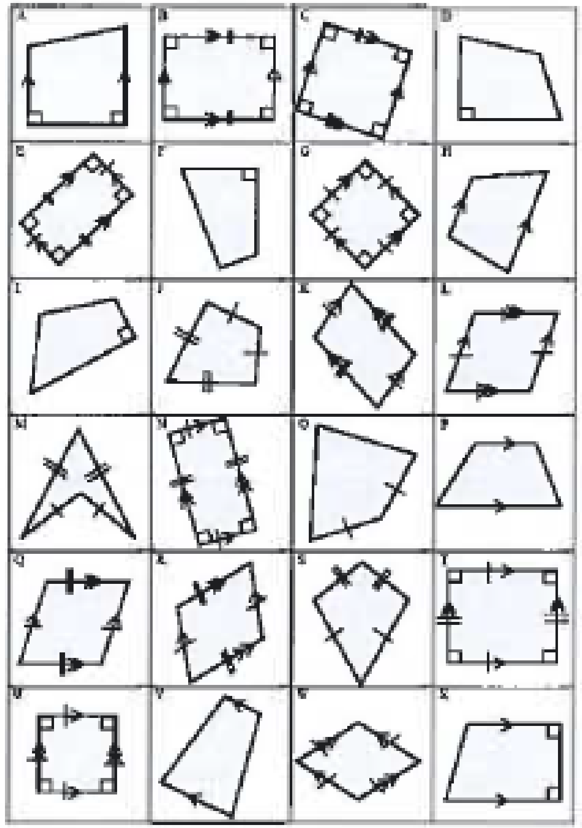
Flowcharts and designing an algorithm (upper primary, junior secondary)

You are provided with a set of shapes that are all examples of quadrilaterals.

Use the shapes to help you design an algorithm that categorises a quadrilateral into one of the following categories:

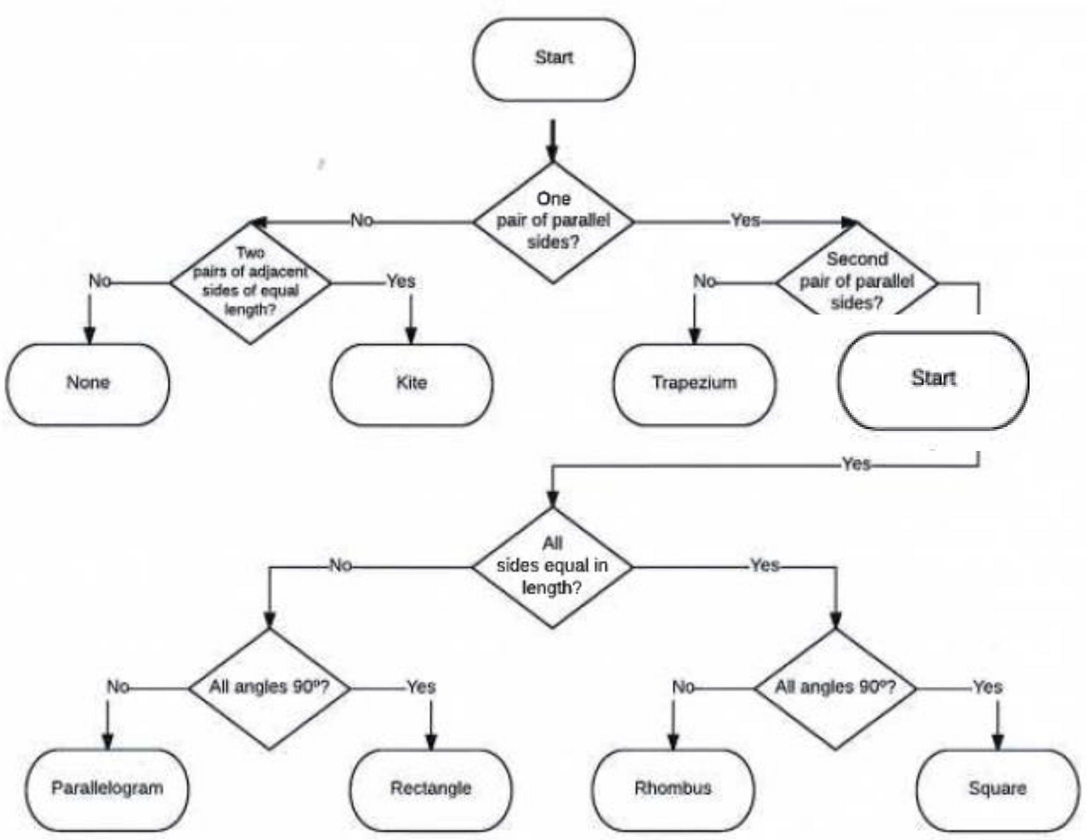
- rectangle, square, parallelogram, rhombus, kite, trapezium, none.

Where multiple categories would be relevant then the most specific category should be provided.

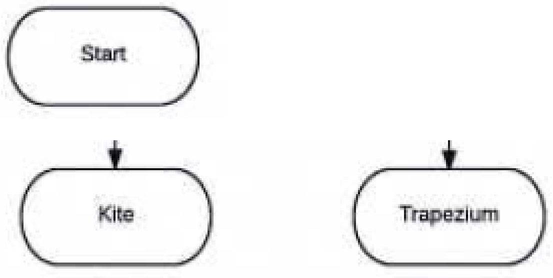


A sorting algorithm and its different elements

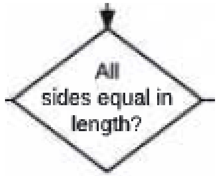
An algorithm described using a flowchart



First element: Starts and ends (running track shapes)



Second element: decision box (diamond shapes)

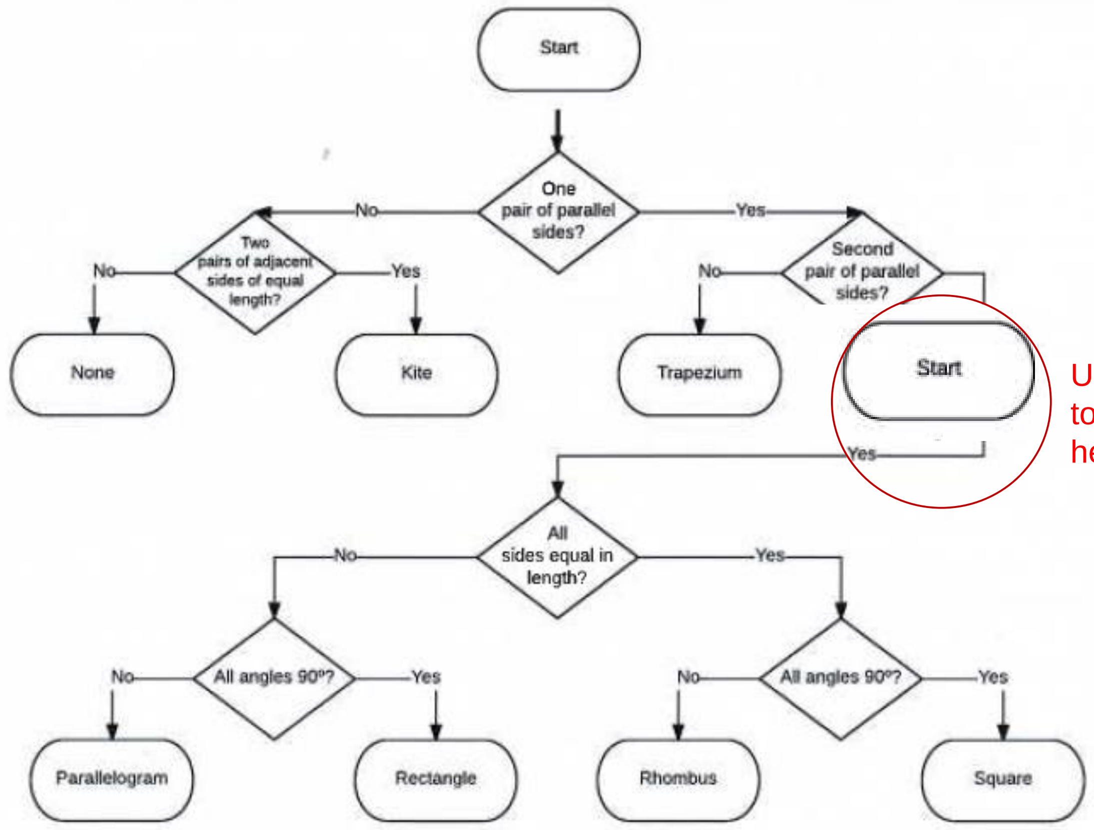


Third element: arrows Yes/No

THE UNIVERSITY OF MELBOURNE

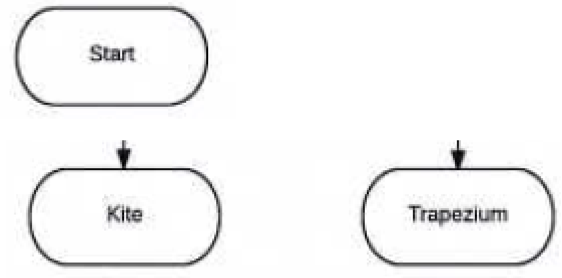
A sorting algorithm and its different elements

An algorithm described using a flowchart

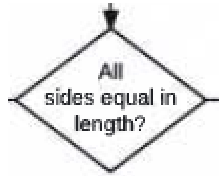


Unnecessary to have Start here

First element: Starts and End points (running track shapes)

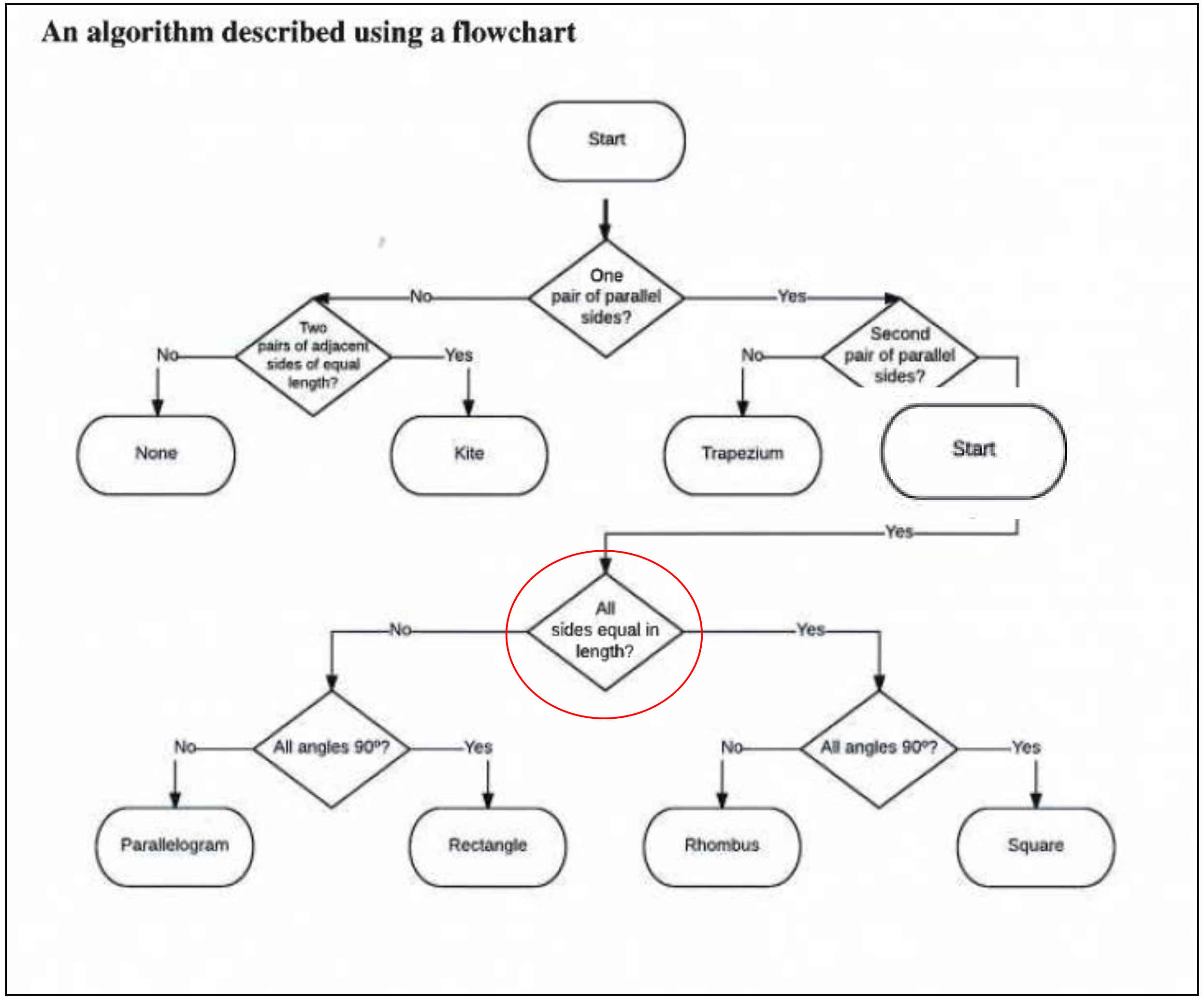


Second element: decision box (diamond shapes)



Third element: arrows Yes/No

A sorting algorithm and its different elements



To get to the decision box “All sides equal in length” what decisions have already been made?

Example for Year 6 also below and above

Design algorithms involving branching and iteration to solve specific classes of mathematical problems

Monkeys & Bananas

Task 212 ... Years 2 - 10

Summary

Three monkeys work all day to collect a big pile of bananas and then flop into bed too tired to eat them. The card then goes on to explain how each monkey got up in the night, ate a banana and then hid one third of the bananas remaining at that time. Despite these night hungries - and displays of selfishness - the monkeys were still able to have equal shares of bananas at breakfast. How many bananas might have been in the original pile.

This cameo includes an Investigation Guide which offers a similar problem as an extension. It also has a From The Classroom section which takes up an idea in the whole class lesson notes below and converts the lesson into a Poster Problem Clinic. This is adapted further by Beth Bright into Minions & Bananas complete with mobile phone apps.

Materials

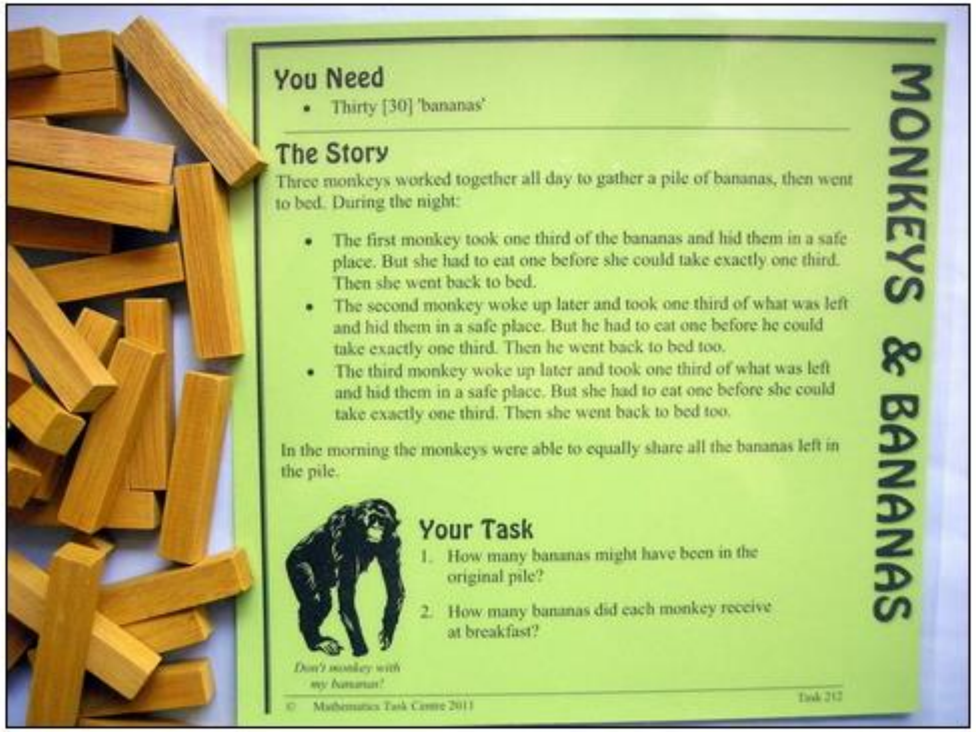
- 30 'bananas'

Content

- basic arithmetic skills
- fractions of a set of objects
- problem solving strategies
- multiplicative thinking
- spreadsheets as a problem solving tool
- linear algebra

Iceberg

A task is the tip of a learning iceberg. There is always more to a task than is recorded on the card.



Some students will begin this problem by guessing a pile number; others by guessing a number the monkeys might have shared at breakfast. Whichever way is chosen the guess will either lead to a fraction of a banana - which doesn't exist in the problem - or will lead to a solution. Along the way, students might discover some, as yet, unrealised information in the problem. For example:

- Suppose the original pile is 23. Then Monkey 1 eats one and divides the remainder into three equal shares. Hey that won't work because 22 doesn't divide equally by 3. Okay then, what about 24. No that won't work either because then there would be 23 to make into thirds. 25? Yep, that's okay!

  Monkey 1 eats one, which leaves 24, hides one third of that, which leaves 16 for Monkey 2 to find.

  Then Monkey 2 eats one...

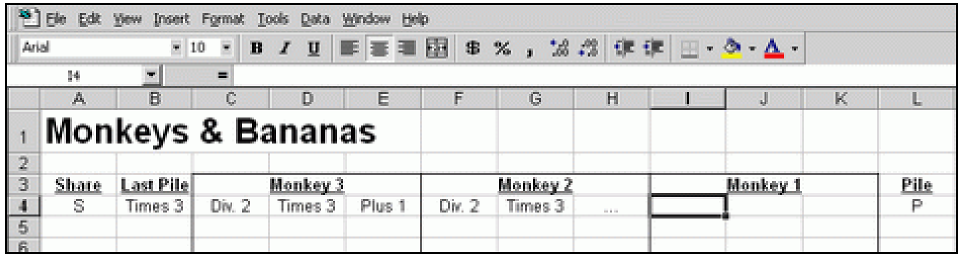
- Whatever is left at the end must be a multiple of three. It also has to come from a multiple of three because Monkey 3 had to hide one third and leave two thirds. Suppose there were 3 bananas for breakfast. What multiple of three number number could that come from by making thirds? It can't come from 3 because two thirds of that is 2. And it can't come from 6 because two thirds of that is 4. So 3 bananas to share at breakfast can't work. What about 6 bananas to share at breakfast? What multiple of three could that come from by making thirds? If it's there, it has to be bigger than 6. Try 9. Hey that works!

  Dividing 9 into three parts and keeping two of them gives 6 bananas for breakfast, so Monkey 3 must have found 10 bananas when he woke up.

  But this had to come from a multiple of three number...

Extensions

- The use of the word 'might' on the card suggests that there also might be another solution. Encourage the students to re-apply their reasoning to look for at least one more.
- This is a great problem for introducing, or reinforcing, how a spreadsheet works. Support students to design a spreadsheet which starts with the breakfast share and works backwards, column by column, to the original pile. Starting from the left, the question used each time to find the formula for the next cell, is *What had to happen to the number in the next cell to get the number in this cell?* Now reverse that.

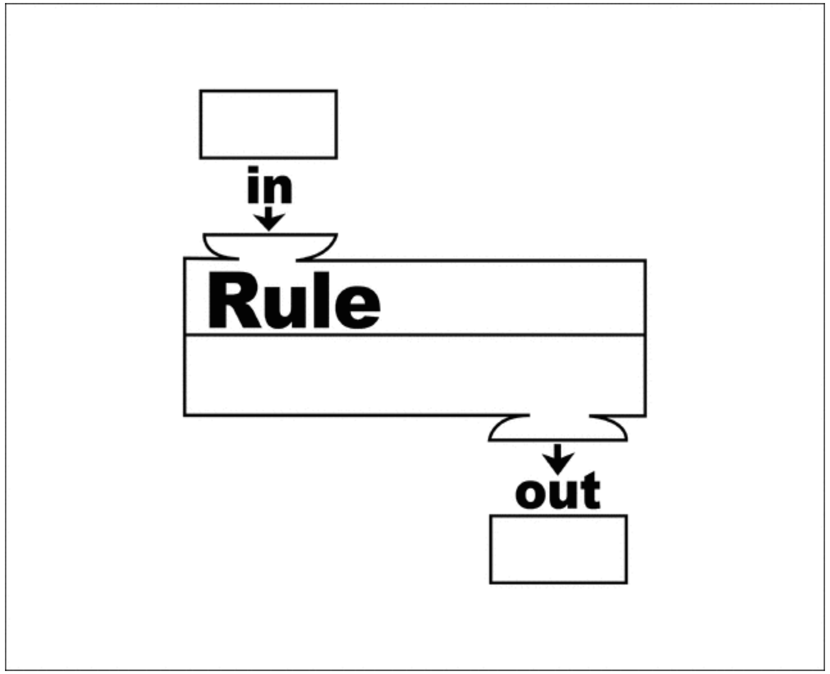


It seems to help students crystallise their thinking to break the problem into parts like this. The approach is essentially the same as that in Maths300 Lesson 19, *Backtracking*.

Setting up formulas for each column also creates a tool that can investigate what happens for any given morning share and thus help uncover other solutions. The introduction of the spreadsheet into this problem often revitalises student interest in it because the burden of the calculations is taken away. Once set up, it only takes milliseconds for the spreadsheet to find several answers and the mathematical focus shifts to looking for patterns in the answers and why the might be there.

A further extension would be to ask questions like *What happens if there are four monkeys in the team and during the night they each eat one and hide one fourth?*.

Represent the rule that the three Monkeys use



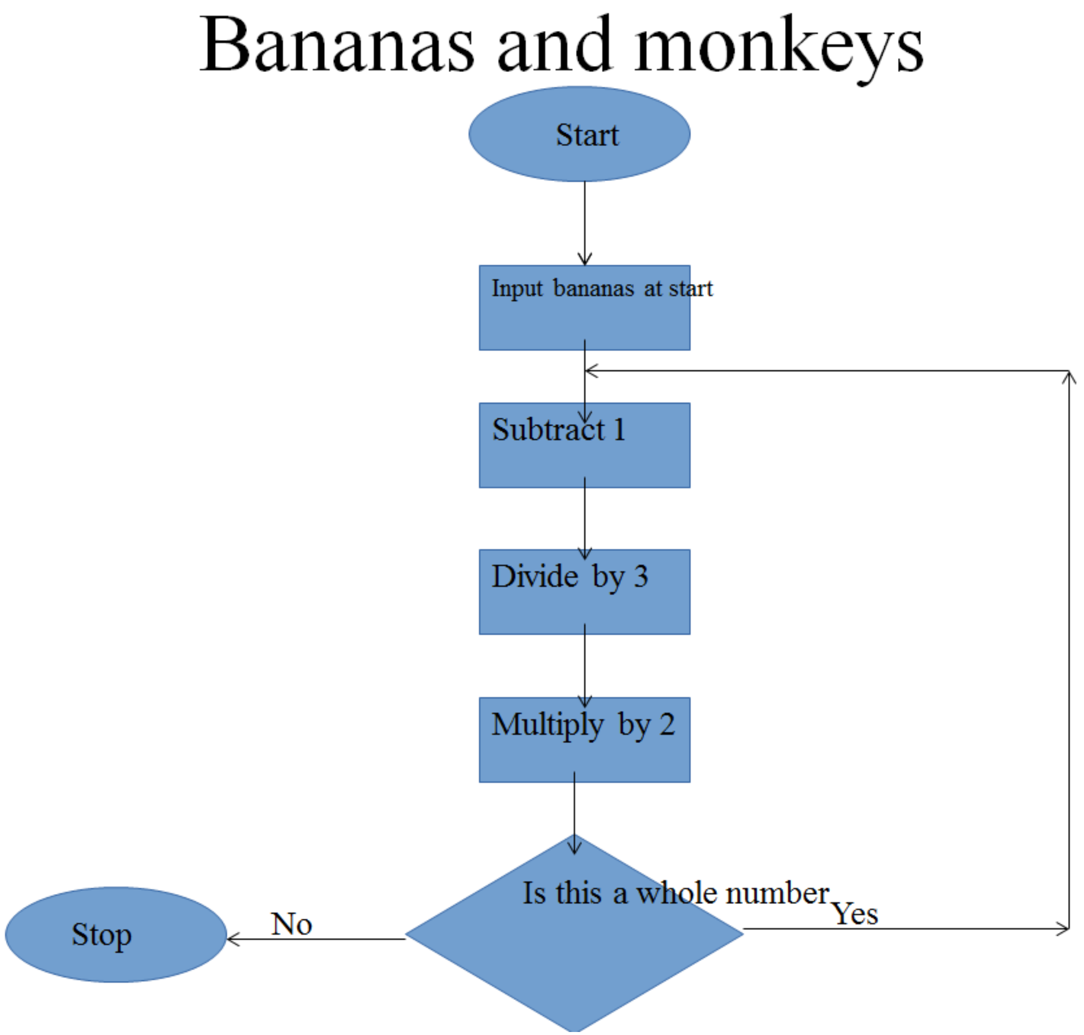
Begin at the Start:

Start with the amount that the monkeys collected during the day.
The first monkey takes one and keeps one third. What's left for Monkey2?

Write a Rule.

More than one way of writing this rule.

Flowchart representing the rule three Monkeys use



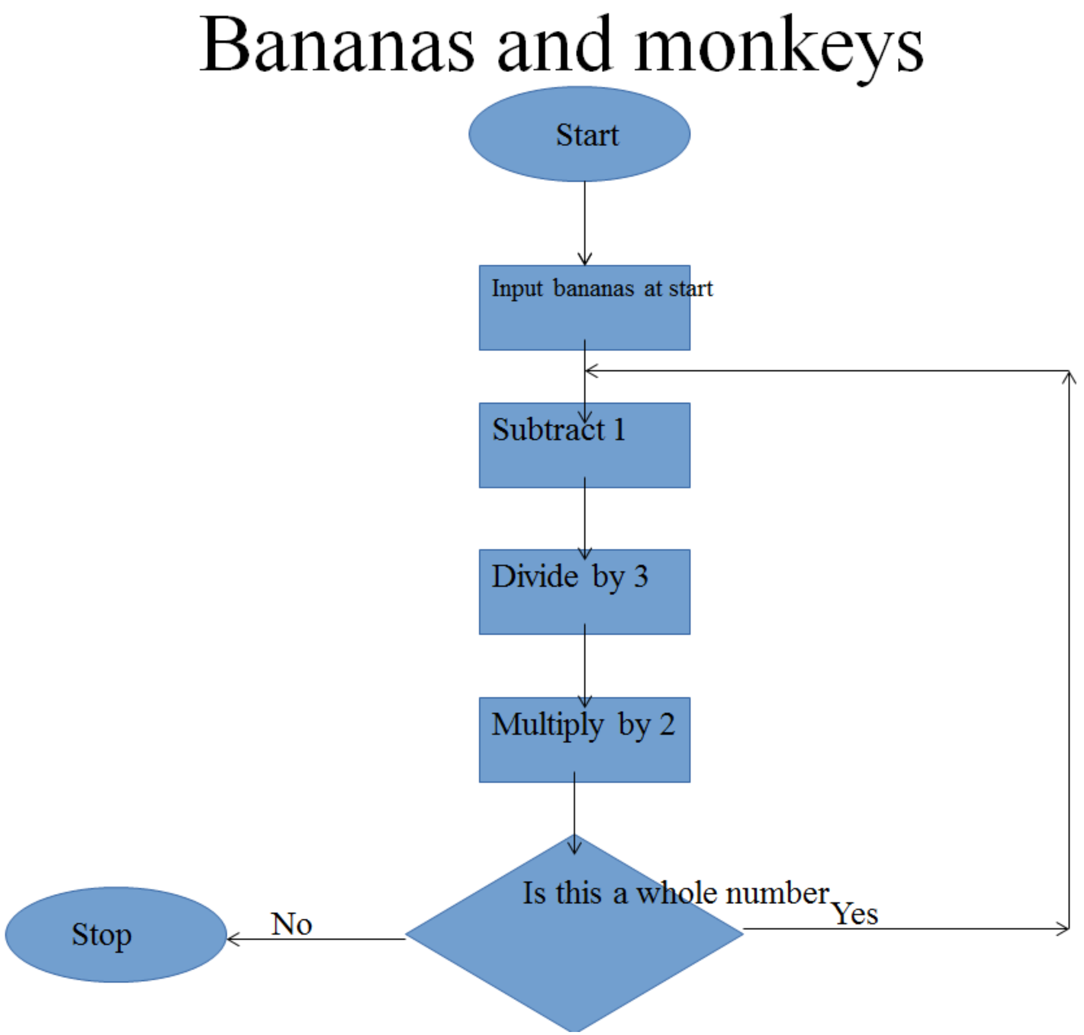
Begin at the Start:
Start with the amount that the monkeys collected during the day. The first monkey takes one and keeps one third. What's left for Monkey2?

The rule is Subtract 1 and take 2/3 of what is left.

Can this be expressed in another way?

Monkey 2 and Monkey 3 follow the same rule.

Flowchart representing the rule three Monkeys use



Begin at the Start:

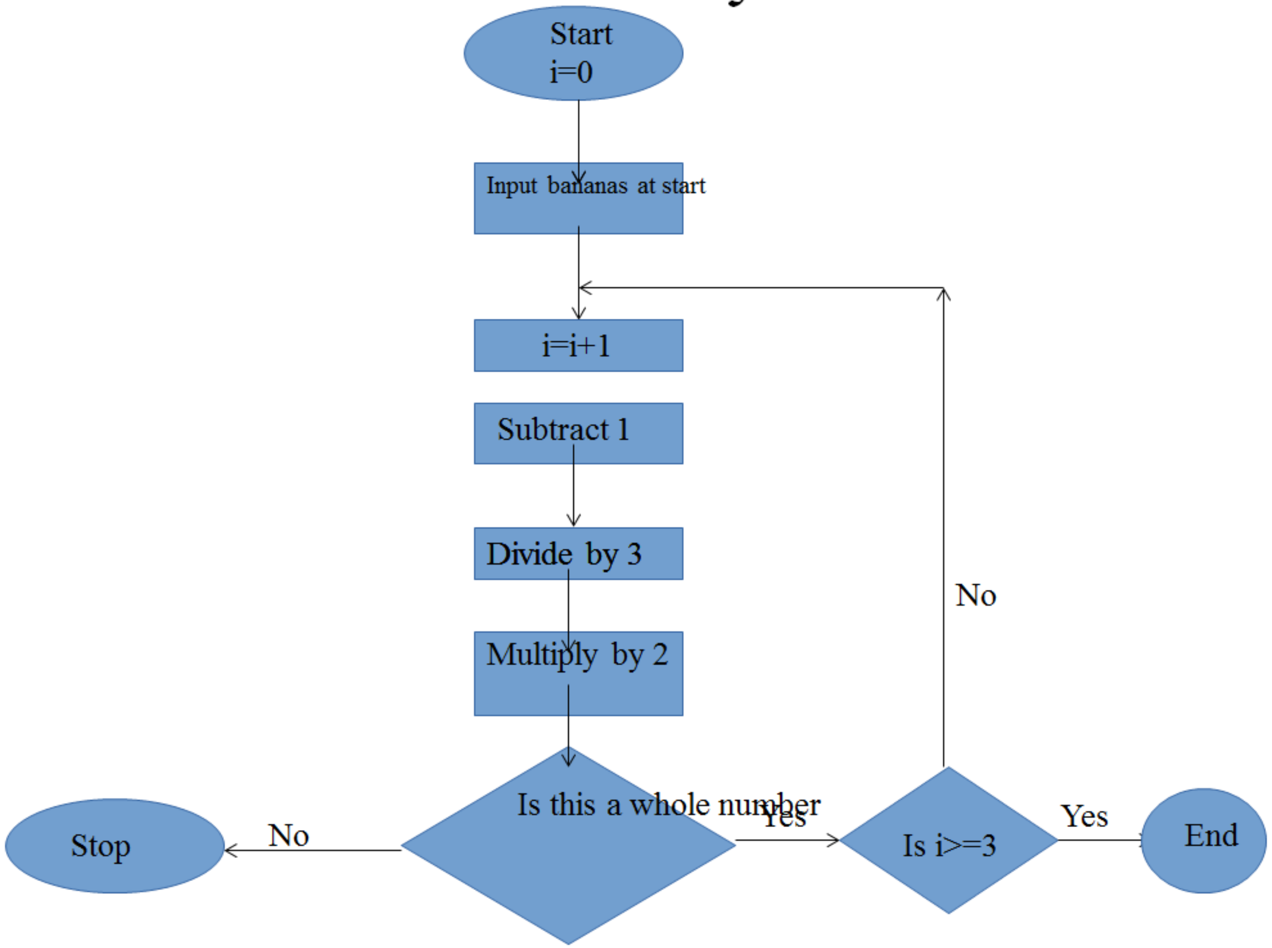
Start with the amount that the monkeys collected during the day. The first monkey takes one and keeps one third. The rule is Subtract 1 and take 2/3 of what is left.

PROBLEM:
But we want to repeat this only three times.

We don't want to go around the loop forever.

Flowchart to represent the rule three Monkeys use

Bananas and monkeys: actual



Begin at the Start:

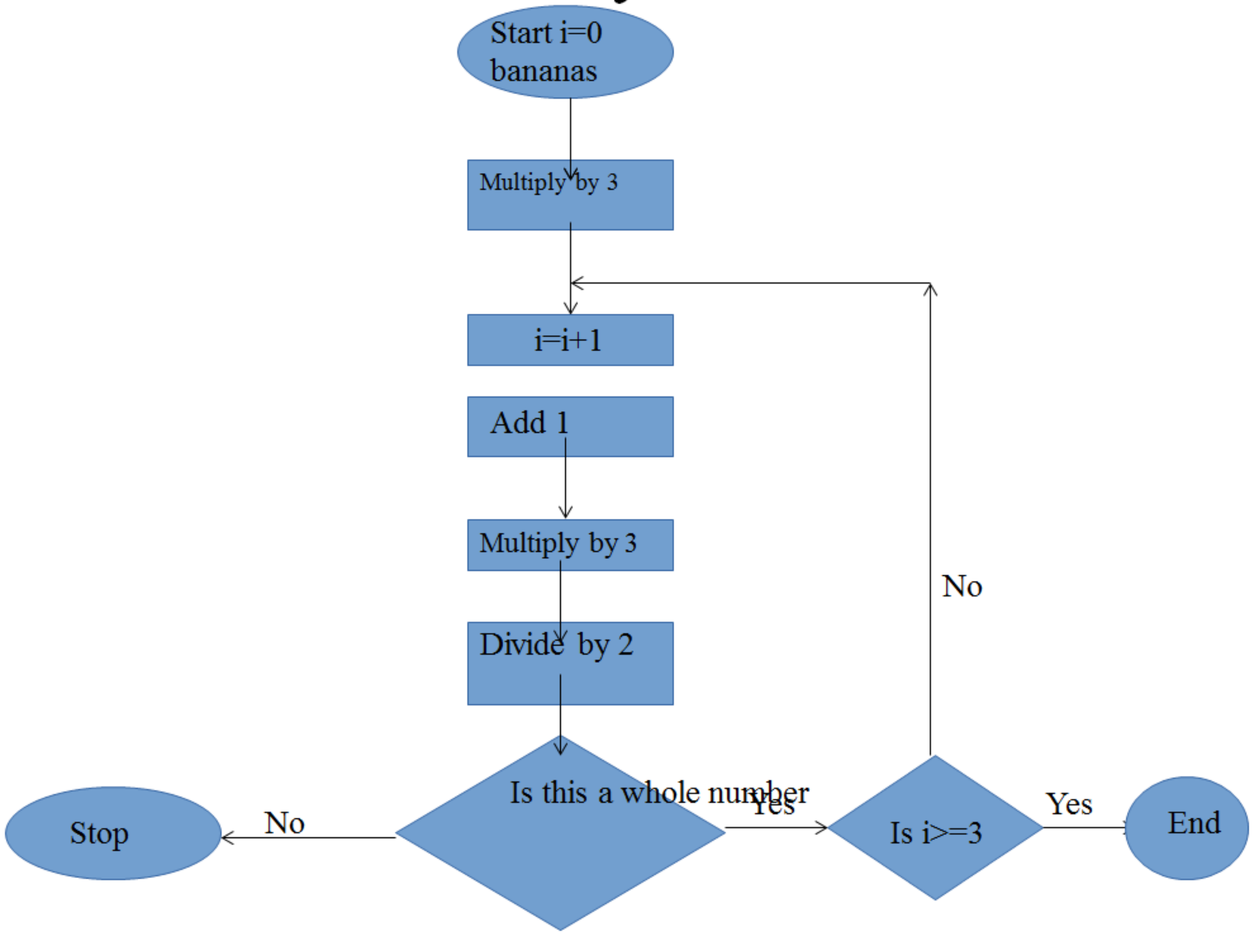
We want to repeat the loop three times.

Flowchart writers use a counter tool:
Start i = 0 and then i= I + 1

When i > 3 the process stops. Clever.

Flowchart to represent the rule the three Monkeys use

Bananas and monkeys: from the end



Working backwards:

Start with the final share for each monkey

Multiply it by 3 and that's what was left after Monkey 3

Then back to Monkey 2

Then back to Monkey 1

And End with the initial collection

.

Using a table (with EXCEL but not necessarily)

Working back from the final share

Final Share Each	Final pile	Pile × 3/2	Add 1	Pile × 3/2	Add 1	Pile × 3/2	Add 1
1	3	4.5					
2	**6**	**9**	**10**	**15**	**16**	**24**	**25**
3	9	13.5					
4	12	18	19				
5	15						
6	18	27	28	42	43		
7	21						
8	24	36	37				
9	27						
10	**30**	**45**	**46**	**69**	**70**	**105**	**106**
11	33						
12	36	54	55				
13	39						
14	42	63	64	96	97		
15	45						
16	48	72	73				
17	51						
18	**54**	**81**	**82**	**123**	**124**	**186**	**187**
19							
20							
21							
22							
23							
24							
25							
26							

Identifying patterns in the Table

Final Share Each	Final pile	Pile × 3/2	Add 1	Pile × 3/2	Add 1	Pile × 3/2	Add 1
1	3	4.5					
2	**6**	**9**	**10**	**15**	**16**	**24**	**25**
3	9	13.5					
4	12	18	19				
5	15						
6	18	27	28	42	43		
7	21						
8	24	36	37				
9	27						
10	**30**	**45**	**46**	**69**	**70**	**105**	**106**
11	33						
12	36	54	55				
13	39						
14	42	63	64	96	97		
15	45						
16	48	72	73				
17	51						
18	**54**	**81**	**82**	**123**	**124**	**186**	**187**
19							
20							
21							
22							
23							
24							
25							
26	**78**	**117**	**118**	**177**	**178**	**267**	**268**

Identifying patterns in the Table

Final Share Each	Final pile	Pile × 3/2	Add 1	Pile × 3/2	Add 1	Pile × 3/2	Add 1
1	3	4.5					
2	**6**	**9**	**10**	**15**	**16**	**24**	**25**
3	9	13.5					
4	12	18	19				
5	15						
6	18	27	28	42	43		
7	21						
8	24	36	37				
9	27						
10	**30**	**45**	**46**	**69**	**70**	**105**	**106**
11	33						
12	36	54	55				
13	39						
14	42	63	64	96	97		
15	45						
16	48	72	73				
17	51						
18	**54**	**81**	**82**	**123**	**124**	**186**	**187**
19							
20							
21							
22							
23							
24							
25							
26	**78**	**117**	**118**	**177**	**178**	**267**	**268**

B3 =(3/2)*(((3/2)*A3+1)*(3/2)+1)+1

	A	B	C	D	E
1		Table 1			
2	Number being checked	Equation Answer	Contains Answer if number = 3N	Contains Equation Answer if it is a whole number	3N
3	1	8.125	FALSE	FALSE	0
4	2	11.5	FALSE	FALSE	0
5	3	14.875	14.875	FALSE	0
6	4	18.25	FALSE	FALSE	0
7	5	21.625	FALSE	FALSE	0
8	6	25	25	25	6
9	7	28.375	FALSE	FALSE	0
10	8	31.75	FALSE	FALSE	0
11	9	35.125	35.125	FALSE	0
12	10	38.5	FALSE	FALSE	0
13	11	41.875	FALSE	FALSE	0
14	12	45.25	45.25	FALSE	0
15	13	48.625	FALSE	FALSE	0
16	14	52	FALSE	FALSE	0
17	15	55.375	55.375	FALSE	0
18	16	58.75	FALSE	FALSE	0

This is the Excel program working backwards from the final share.
Suitable for high school. The algorithm is written in a form suitable for Excel:

fx =(3/2)*(((3/2)*A3+1)*(3/2)+1)+1

A simple but longer definition for Algorithmic thinking

Building on the Monkeys and Bananas Problem:

- When given any problem, you can formulate it so that it can solved using the power of computers (even a human computer)?
- As a computational thinker, you collect data and analyse it to understand the problem. Then you decompose (break it down) into simpler problems.
- Instead of solving only that problem, you look for patterns, remove details and abstract so you can solve all problems of that type.
- By defining the steps to solve the problem, you get an algorithm; and if possible, you build a model to simulate, test and debug the solution.
    - https://www.edsurge.com/news/2016-08-06-what-s-the-difference-between-coding-and-computational-thinkingoodle below explaining these components of computational thinking.

What’s important for the Primary school years

- Algorithmic thinking should grow naturally out of the mathematics curriculum
- Help students develop mathematical reasoning and problem solving
- Looking for patterns, breaking down problems into simple steps, being able to communicate the steps that can be followed by others/computer
- Able to scale the steps to other more complex problems and numbers
- Algorithms can be represented in different ways – words/instructions, visually (flowchart), programming language
- Programming language may be block/robotic instructions, spreadsheet, Geogebra, or standard programming language
- The thinking behind the algorithm or programming or coding remains the most important thing to cultivate.

What's important:

- Developing the language of communicating Algorithms (Computational Thinking) is critical at all stages of the curriculum
- No particular language or programming medium is required
- Linked to the digital technologies curriculum but quite distinct purposes
- Algorithms focus on mathematics of number, shape, patterns and algebra
- Student group work is particularly powerful to develop and refine algorithmic thinking. So is recording if thinking is to be evident.
- Display recorded student work; students can discuss ideas and formulate understanding as a group; group presentations also

Acknowledgements

- Megan Blanch, Mathspace Victoria
- Gregory Breese, Glen Waverley Secondary College
- Mark Stephens, Tarneit Senior College
- Doug Williams, Maths 300

- 1. For pre-service

Pre service teacher free Mathspace Account

https://mathspace.co/blog/Free-Student-Access/

Pick up the handout from Mathspace

- 2. Free Mathspace Essentials

mathspace.co and click FREE SIGN UP

- 3. Help website

help.mathspace.co

Example for Year 6 (possibly earlier)

Design algorithms involving branching and iteration to solve specific classes of mathematical problems