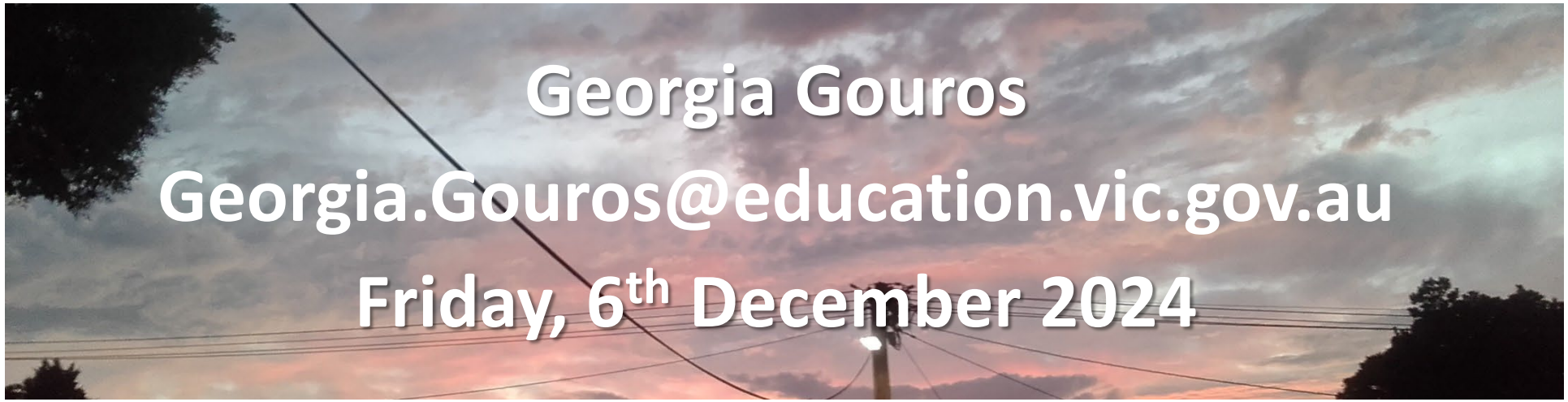
MAV 2024
Lesson activities for 9-10 algorithmic thinking referencing Mathematics 2.0



Georgia Gouros

Georgia.Gouros@education.vic.gov.au

Friday, 6th December 2024

YEAR 7

New Mathematics Version 2.0 curriculum
https://victoriancurriculum.vcaa.vic.edu.au/mathematics/mathematics-version-2-0/curriculum/f-10

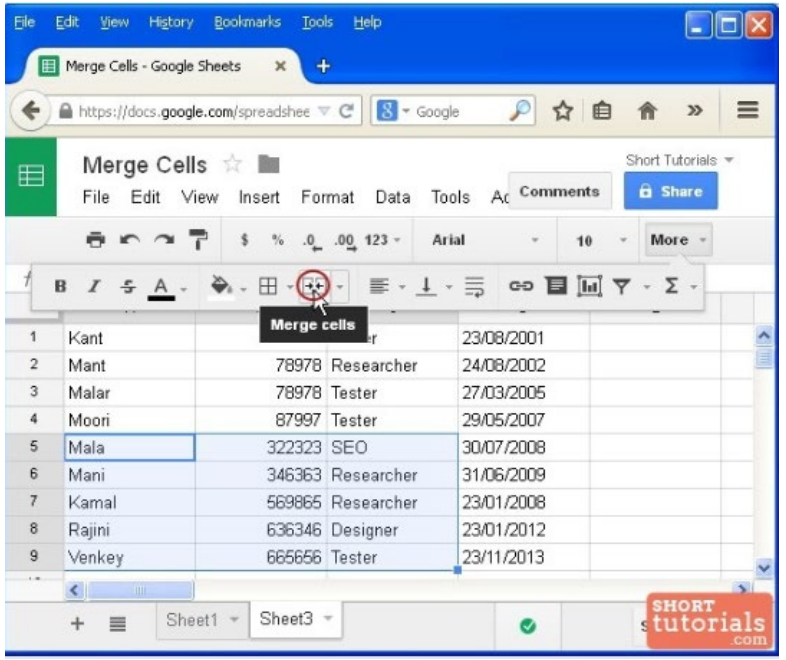
Y7 Number use mathematical modelling to solve practical problems involving rational numbers and percentages, including financial contexts such as 'best buys'; formulate problems, choosing representations and efficient calculation strategies, **designing algorithms** and using digital tools as appropriate; interpret and communicate solutions in terms of the situation, justifying choices made about the representation (VC2M7N10)

Y7 Space design algorithms involving a sequence of steps and decisions that will sort and classify sets of shapes according to their attributes, and describe how the algorithms work (VC2M7SP04)

Y7 Number use mathematical modelling to solve practical problems involving rational numbers and percentages, including financial contexts such as 'best buys'; formulate problems, choosing representations and efficient calculation strategies, **designing algorithms** and using digital tools as appropriate; interpret and communicate solutions in terms of the situation, justifying choices made about the representation (VC2M7N10)

Excel Spreadsheets/Google Sheets

- Favourite colours - carry out a class survey and use tech to display the data histogram/pie-chart
- Planning a party - create a simple model using a variety of operations for staying within a budget
- Simple and compound interest exploration generate next values



https://www.digitaltechnologieshub.edu.au/teach-and-assess/classroom-resources/lesson-ideas/a-spreadsheet-s-secret-weapon/

Y7 Number use mathematical modelling to solve practical problems involving rational numbers and percentages, including financial contexts such as 'best buys'; formulate problems, choosing representations and efficient calculation strategies, **designing algorithms** and using digital tools as appropriate; interpret and communicate solutions in terms of the situation, justifying choices made about the representation (VC2M7N10)

https://sites.google.com/a/dhsd7.net/mrs-maurus-technology-classroom/spreadsheet-shopping-spree

Google Sheets

Prompt: A distant relative in California, whom you have never met, has won the lottery. He was awarded $1,000,000 and has decided to be very generous with the money! He is giving every relative that he knows $1,000 to spend, and the timing could not be more perfect for the holiday gift-buying season! He attached a note to your check to remind you to be very generous when spending the money.

Restrictions: **He warns that you are not spend more than $100 on yourself, and you must not spend more than $200 on each person you buy gifts for. You MUST spend EXACTLY $1,000 (no more, no less) or all of the money will have to be repaid.**

Obligations/Considerations: You should probably know that this relative has two kids of his own, a four year-old son and a 7 year-old daughter. You might want to send them a little something too, to show your gratitude to the relative that gave you this money. Use the chart on the back to plan out your shopping trip. You do not need to use all of the lines, but don't forget your siblings and your parents!

Y7 Number use mathematical modelling to solve practical problems involving rational numbers and percentages, including financial contexts such as 'best buys'; formulate problems, choosing representations and efficient calculation strategies, **designing algorithms** and using digital tools as appropriate; interpret and communicate solutions in terms of the situation, justifying choices made about the representation (VC2M7N10)

Google Sheets Shopping List

https://sites.google.com/a/dhsd7.net/mrs-maurus-technology-classroom/spreadsheet-shopping-spree

3	Name of Toy	Cost of Each Item	Quantity (How Many Do You Want to Buy?)	
4	Remote Control Go Cart	☺ 49.84	1	
5	Ford F150 Harley	☺ 44.84	1	
6	1,000 Count Case of PaintBalls	☺ 26.77	3	

Creating Formulas

To calculate the **Total** of each row, Type the formula

=(B4*C4). = *Cost of Item * Quantity* in cell E4, F4, G4, H4....each cell underneath the *Total* heading. **(The number will change for each row)

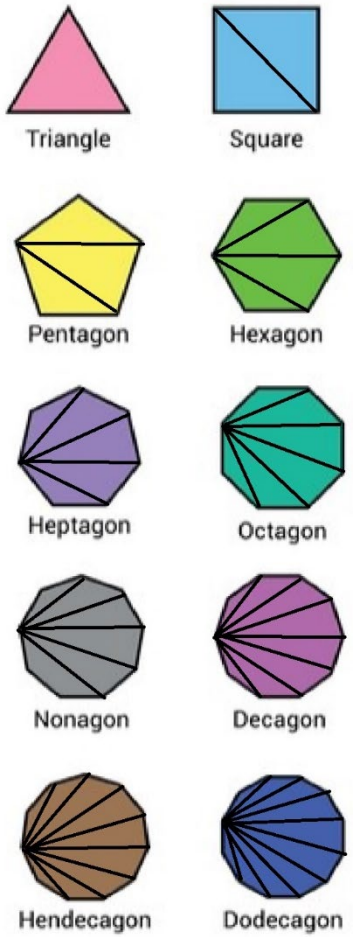
Subtotal: Type the formula =SUM (F4:F26) in cell F27

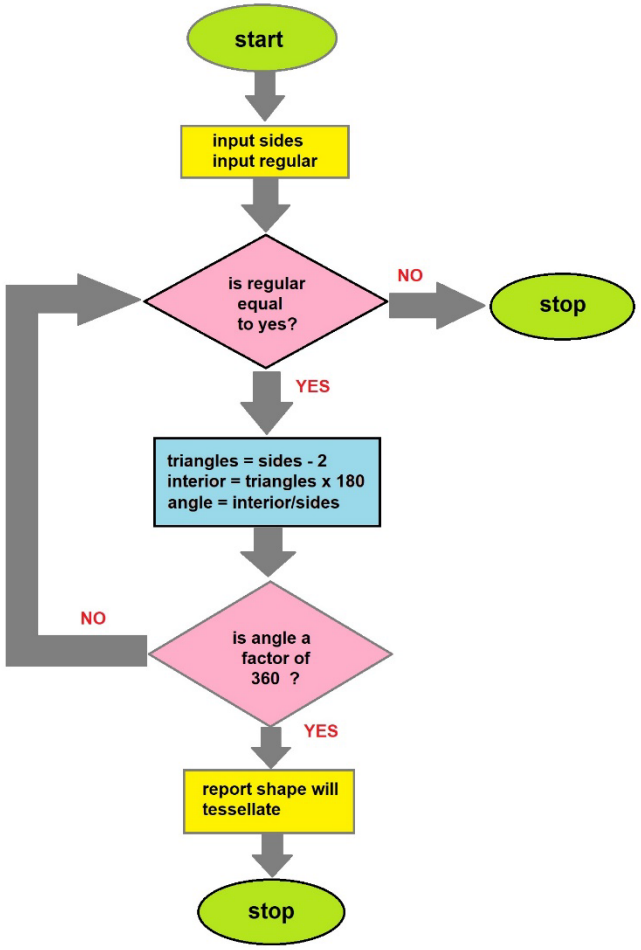
Tax: Type the formula =(F27*0.06) in cell F28

Grand Total: Type the formula =SUM(F27:F28) in cell F29

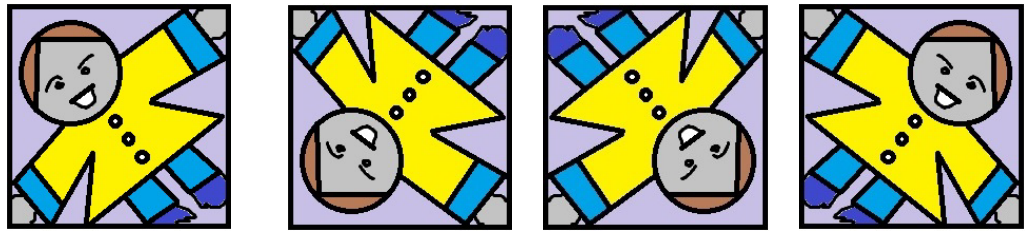
Money Left Over: Type the formula =(1000-F29) in cell F30

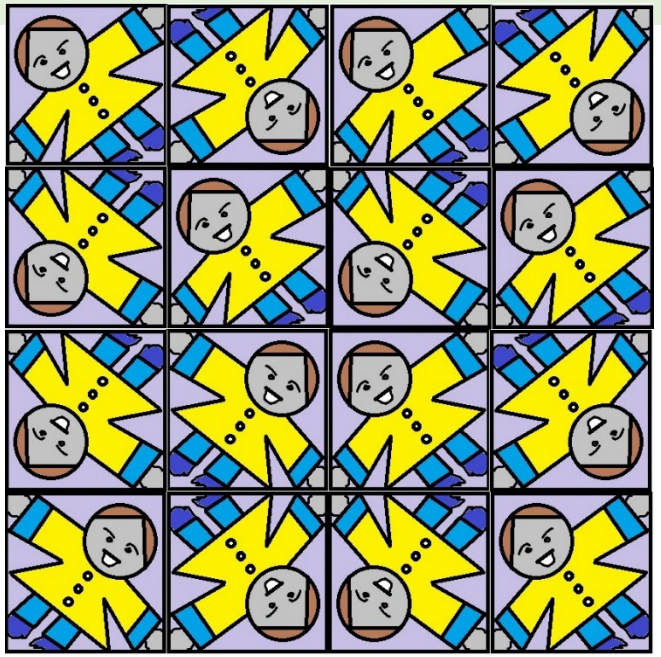
Y7 Space design algorithms involving a sequence of steps and decisions that will sort and classify sets of shapes according to their attributes, and describe how the algorithms work (VC2M7SP04)

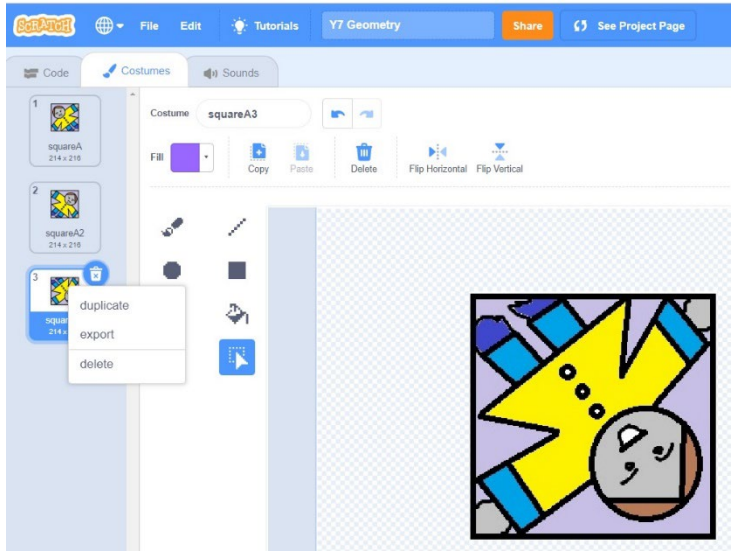




Y7 Space design algorithms involving a sequence of steps and decisions that will sort and classify sets of shapes according to their attributes, and describe how the algorithms work (VC2M7SP04)







YEAR 8

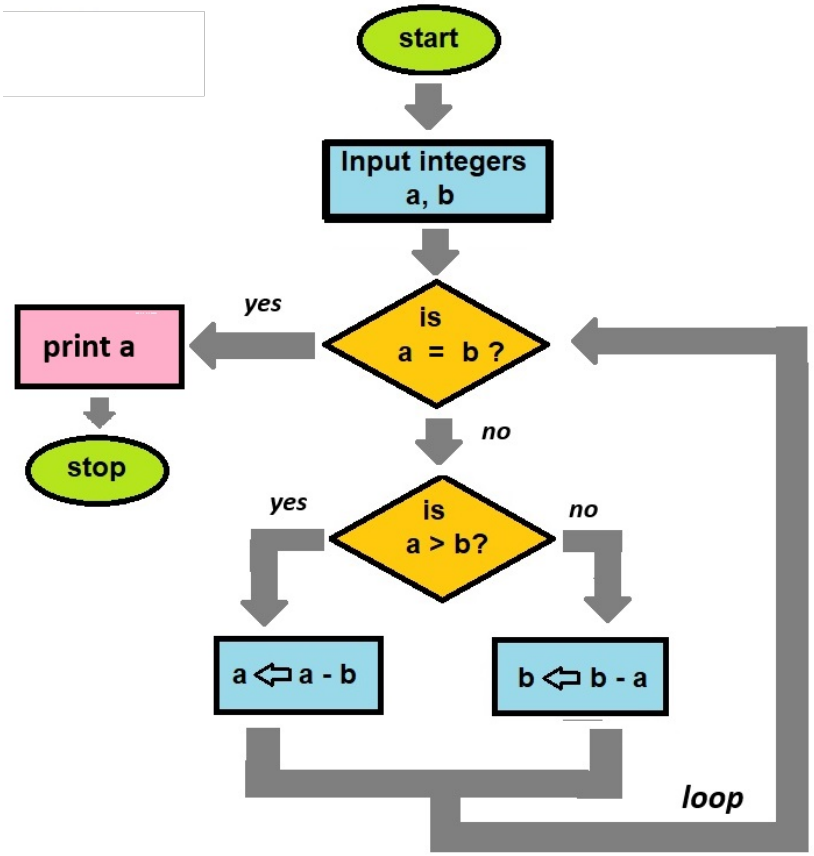
New Mathematics Version 2.0 curriculum
https://victoriancurriculum.vcaa.vic.edu.au/mathematics/mathematics-version-2-0/curriculum/f-10

Y8 Algebra use **algorithms** and related testing procedures to identify and correct errors (VC2M8A04)

Y8 Space design and test algorithms involving a sequence of steps and decisions that identify congruency or similarity of shapes, and describe how the **algorithm** works (VC2M8SP04)

Y8 Algebra - use **algorithms** and related testing procedures to identify and correct errors (VC2M8A04)

Greatest Common Divisor (GCD) – Euclid's Theorem circa 300 BC

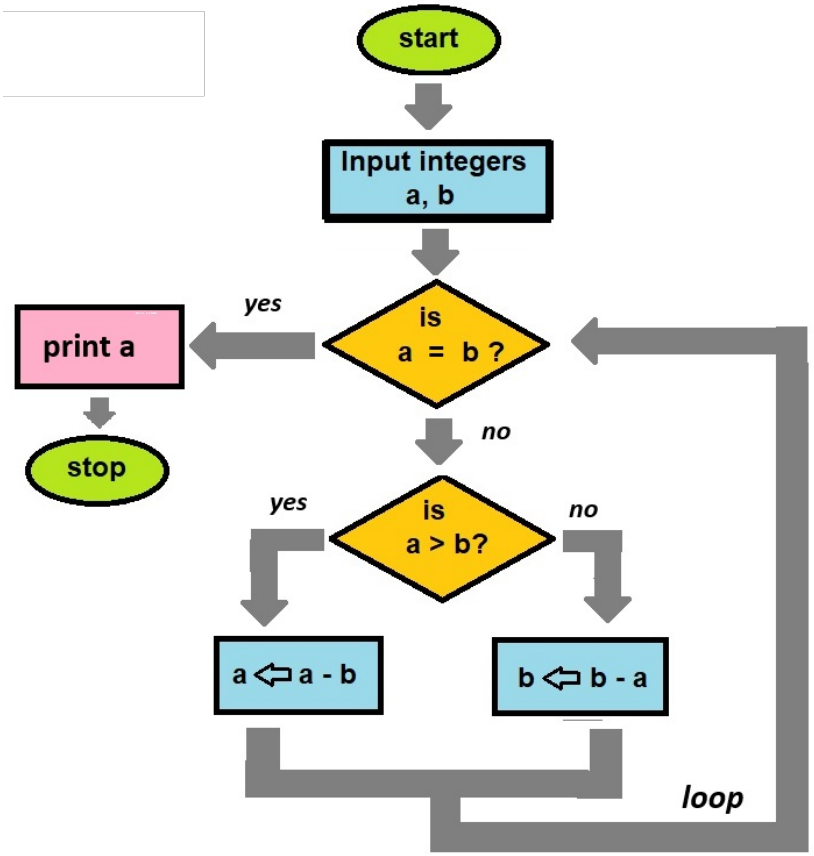


Students explore the algorithm which known multiples

a	b	actions
5	15	a = b? no a > b? no
	10	b <= b - a loop
5	10	a = b? no a > b? no
	5	b <= b – a loop
5	5	a = b? yes Print a stop

Y8 Algebra - use **algorithms** and related testing procedures to identify and correct errors (VC2M8A04)

Greatest Common Divisor (GCD) – Euclid's Theorem circa 300 BC



Students explore the algorithm with prime numbers

a	b	actions
7	11	a = b? no a > b? no
	4	b <= b - a loop
7	4	a = b? no a > b? yes
3		a <= a - b loop
3	4	a = b? no a > b? no
	1	b <= b - a loop
3	1	a = b? no a > b? yes
2		a <= a - b loop
2	1	a = b? no a > b? yes
1		a <= a - b loop
1	1	a = b? yes print a stop

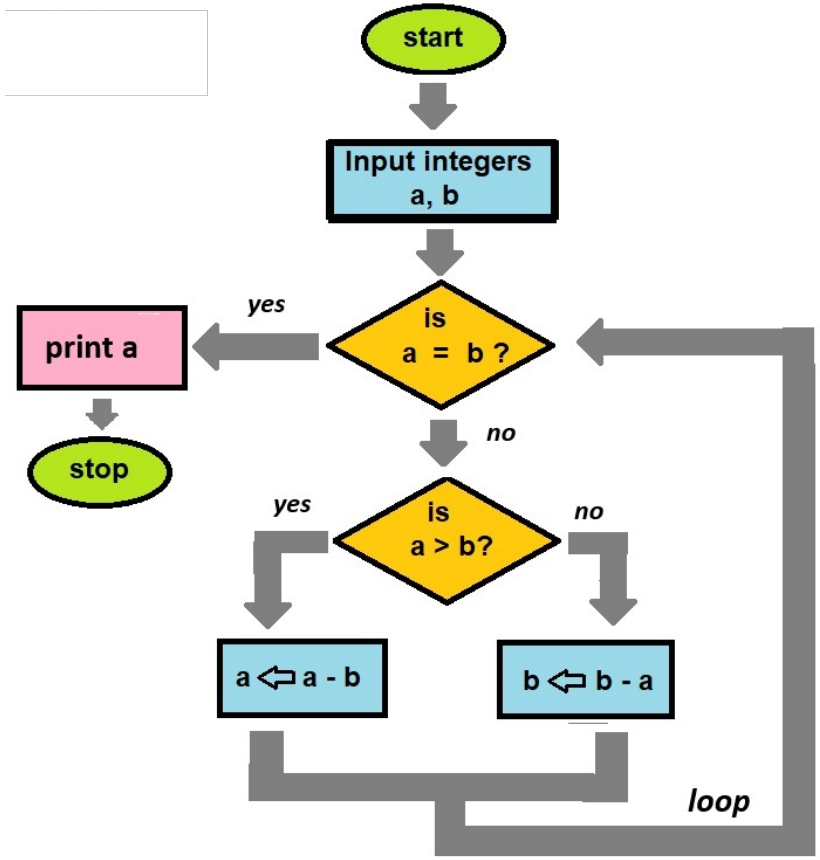
Y8 Algebra - use **algorithms** and related testing procedures to identify and correct errors (VC2M8A04)

Greatest Common Divisor (GCD) – Euclid's Theorem circa 300 BC

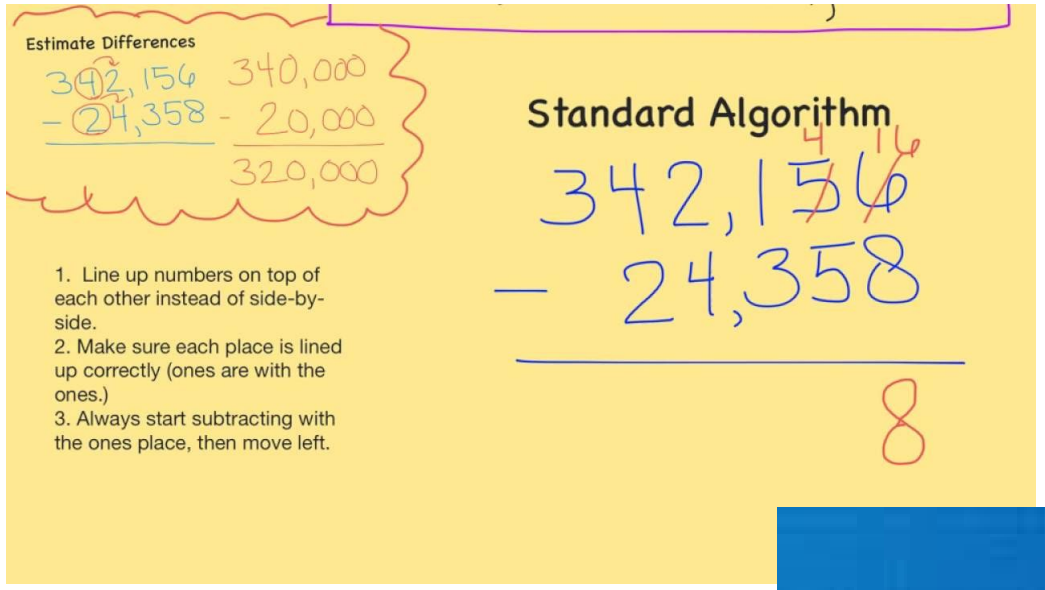
Pseudocode Algorithm

```
Input a, b
while a ≠ b loop
  if a > b
     a := a – b
  else
     b := b – a
end loop
report a
```

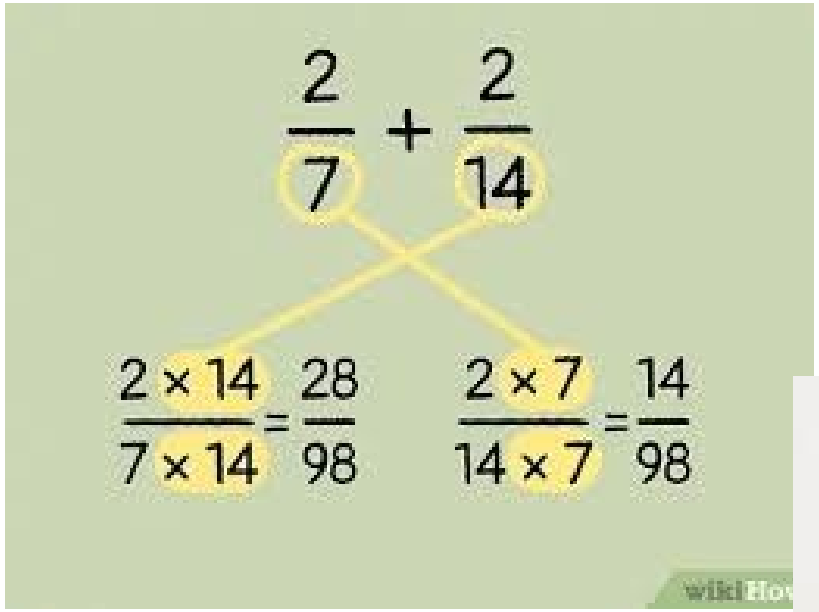
Flowchart Algorithm

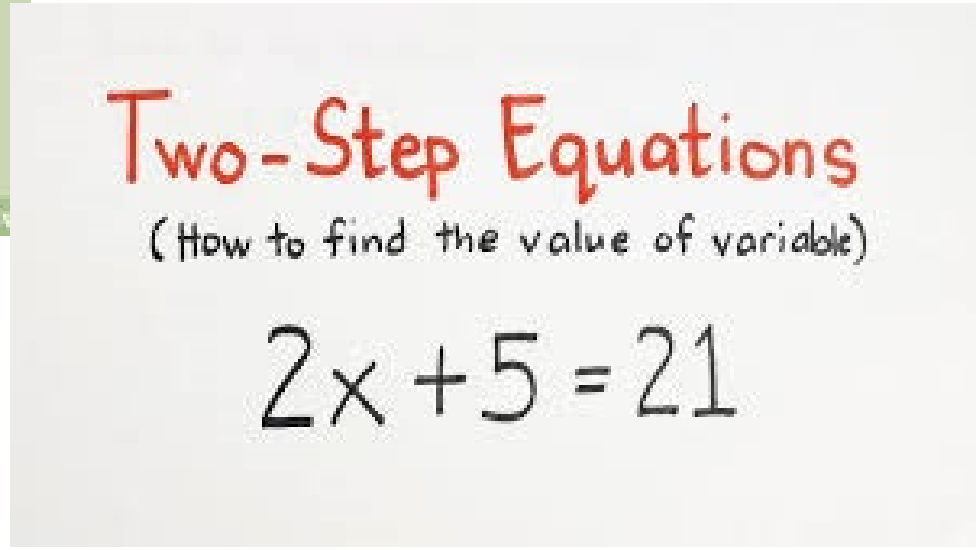


Y8 Algebra - use **algorithms** and related testing procedures to identify and correct errors [(VC2M8A04)]

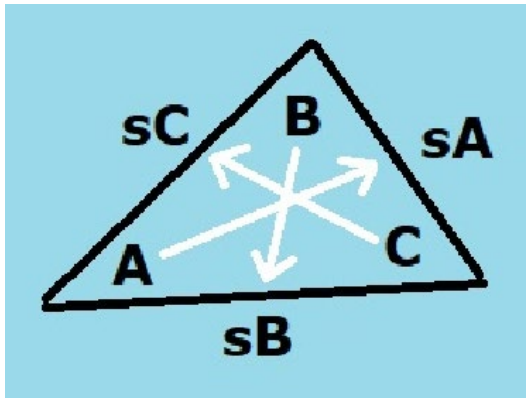


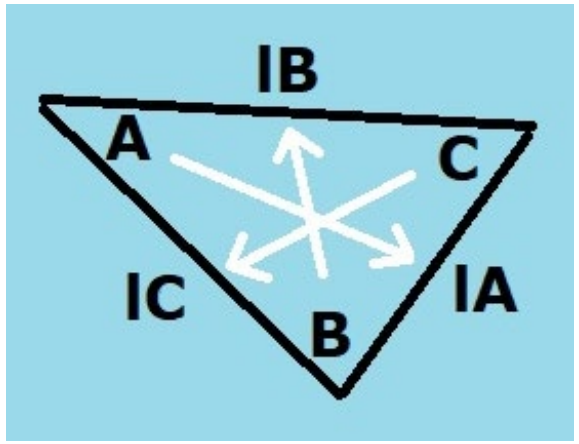
Y8 Algebra - use **algorithms** and related testing procedures to identify and correct errors (VC2M8A04)

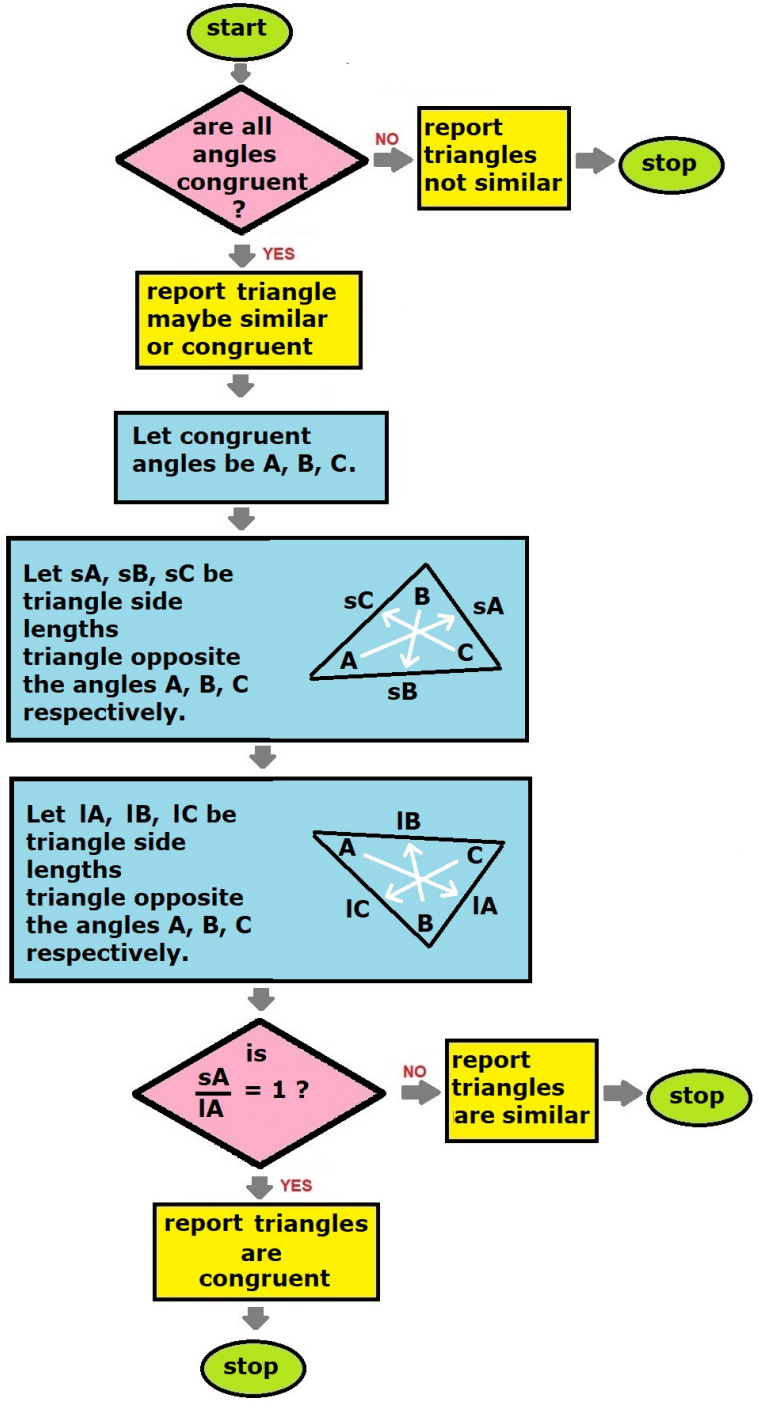




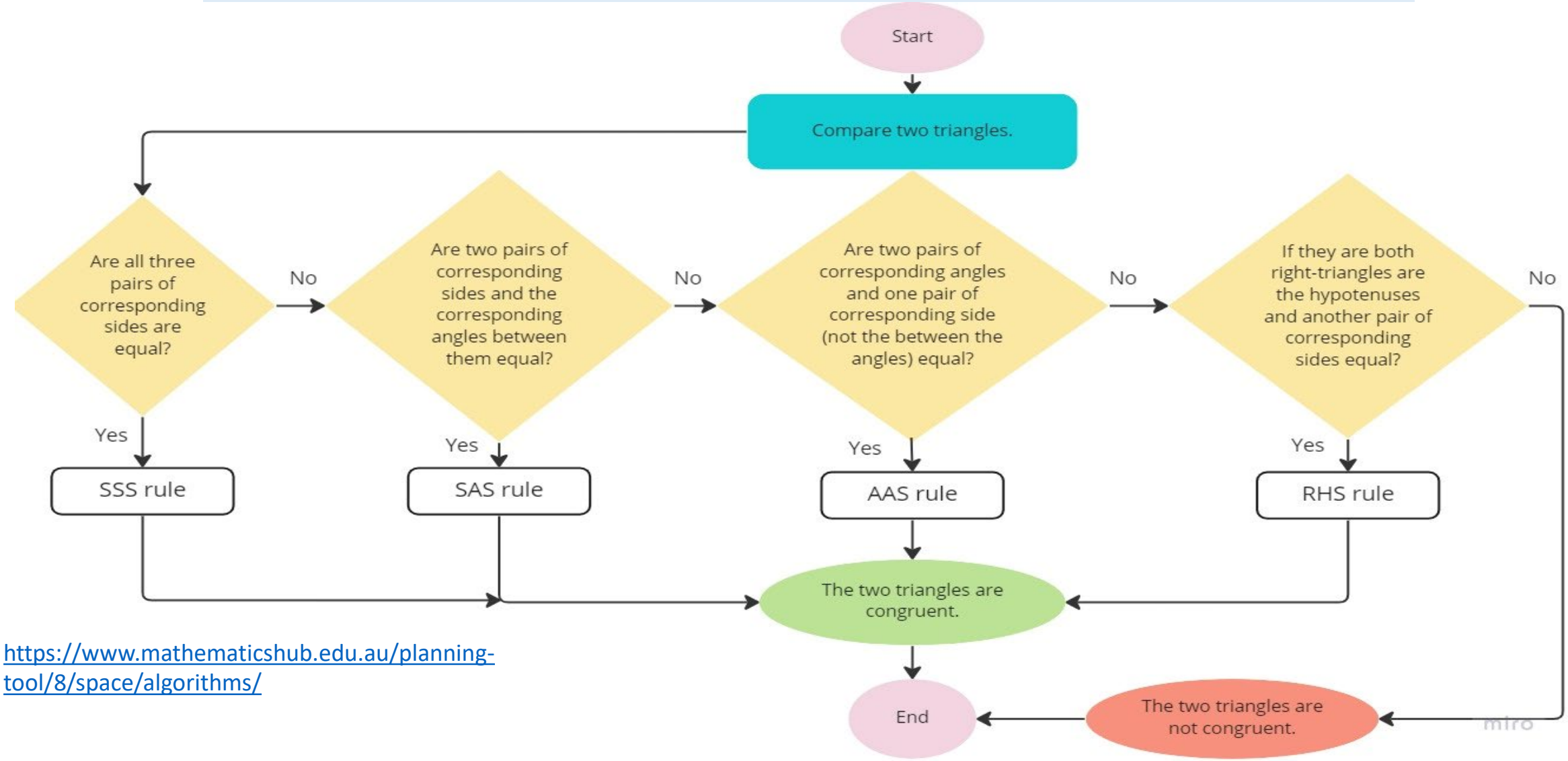
Y8 Space design and test algorithms involving a sequence of steps and decisions that identify congruency or similarity of shapes, and describe how the **algorithm** works (VC2M8SP04)







Y8 Space design and test algorithms involving a sequence of steps and decisions that identify congruency or similarity of shapes, and describe how the **algorithm** works (VC2M8SP04)

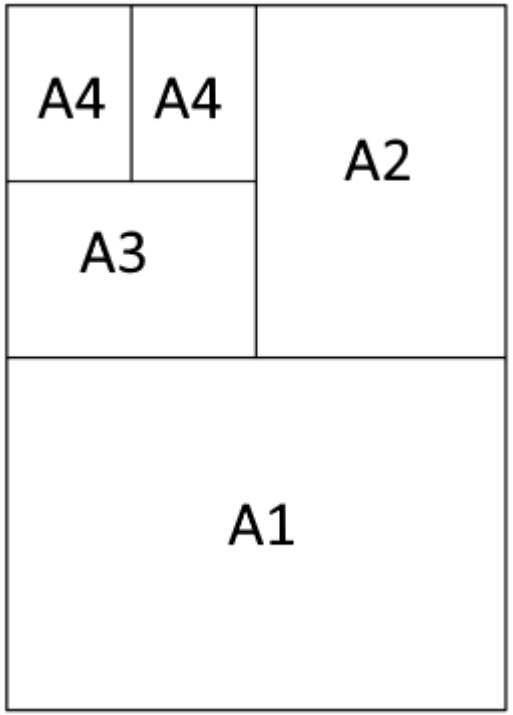


https://www.mathematicshub.edu.au/planning-tool/8/space/algorithms/

Y8 Space design and test algorithms involving a sequence of steps and decisions that identify congruency or similarity of shapes, and describe how the **algorithm** works (VC2M8SP04)

Group Activity: Design an algorithm for finding the length and width relationship of different paper sizes.

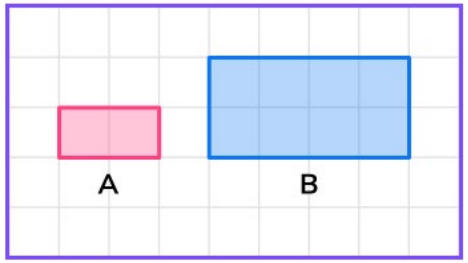
A sheet of A0 has an area of 1 square metre.

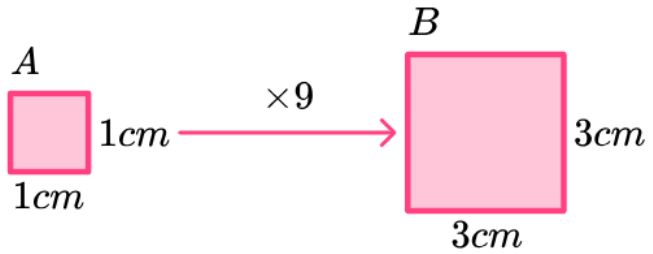


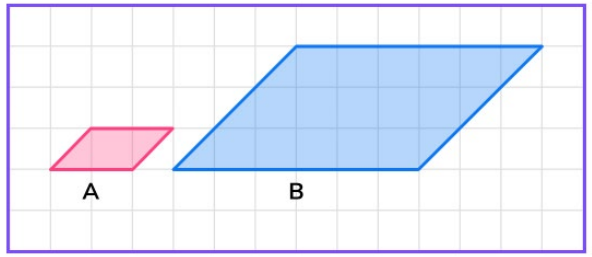
https://nrich.maths.org/problems/fit-photocopying

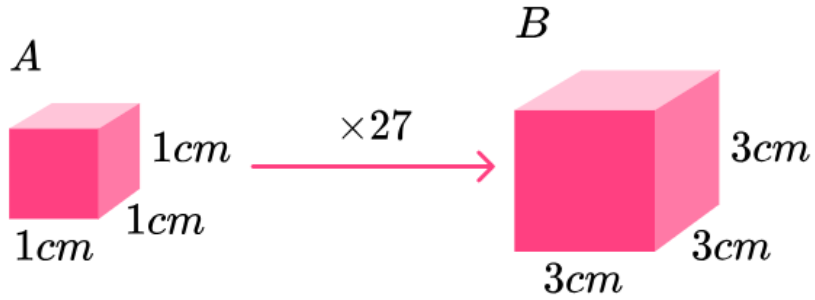
Y8 Space design and test algorithms involving a sequence of steps and decisions that identify congruency or similarity of shapes, and describe how the **algorithm** works (VC2M8SP04)

Group Activity: Design an algorithm for finding congruency or similarity of shapes.







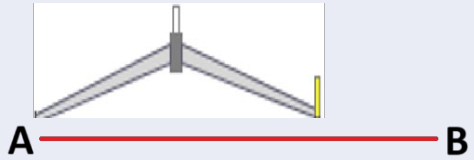
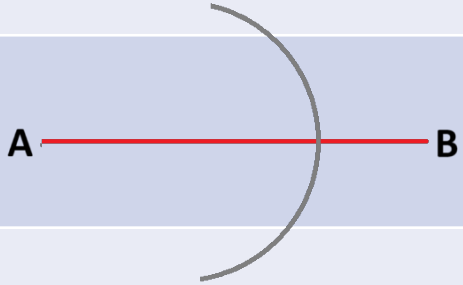
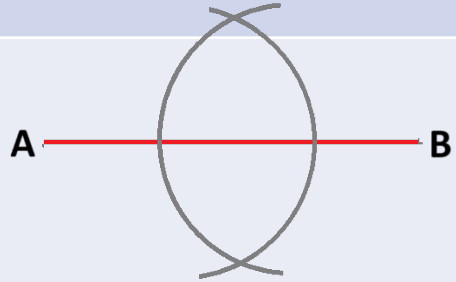
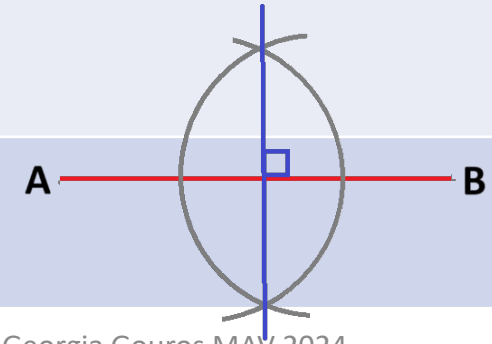


YEAR 9

New Mathematics Version 2.0 curriculum
https://victoriancurriculum.vcaa.vic.edu.au/mathematics/mathematics-version-2-0/curriculum/f-10

Y9 Space - design, test and refine **algorithms** involving a sequence of steps and decisions based on geometric constructions and theorems; **discuss and evaluate refinements** (VC2M9SP03)

Algorithm: Constructing the perpendicular bisector of a line segment

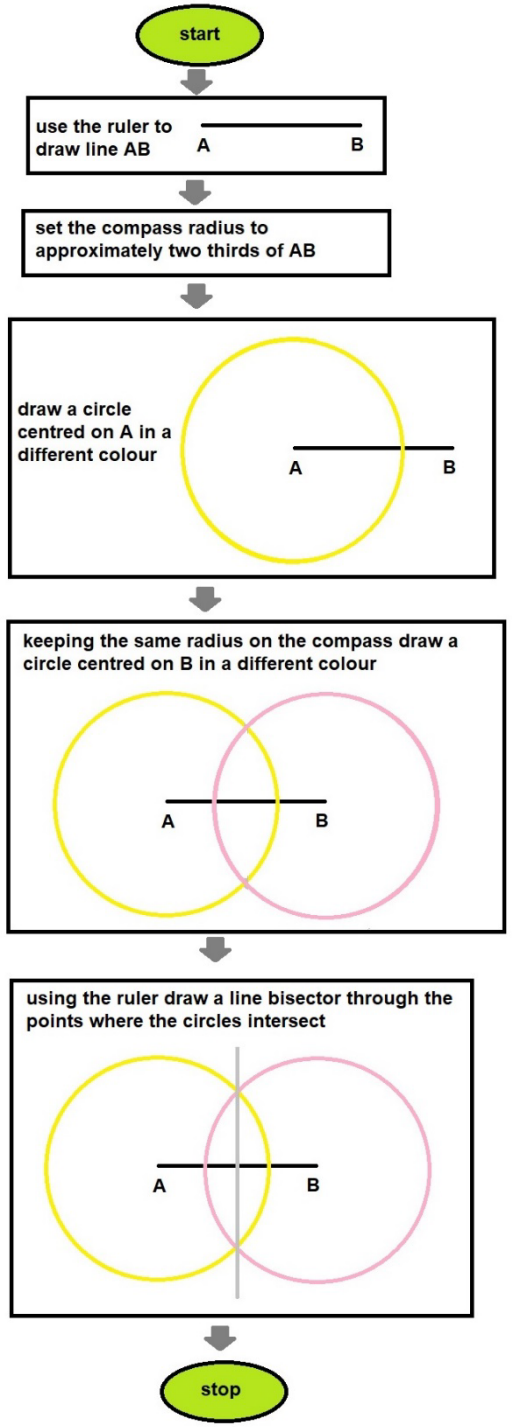
Step	After this step	Your construction should look like this
1	With a ruler draw with a line segment AB.	
2	Place the compass at point A set the compasses' width to approximately two thirds the line length.	
3	Without changing the compasses' width, draw a semi-circle arc above and below the line.	
4	Without changing the compasses' width, place the compasses' point at point B. Draw a semi-circle arc above and below the line so that the arcs cross above and below the line	
5	Using a ruler, draw a line between the points where the arcs intersect.	

Step	Algorithm: Constructing the perpendicular bisector of a line segment
1	With a ruler draw with a line segment AB.
2	Place the compass at point A set the compasses' width to approximately two thirds the line length.
3	Without changing the compasses' width, draw a semi-circle arc above and below the line.
4	Without changing the compasses' width, place the compasses' point at point B. Draw a semi-circle arc above and below the line so that the arcs cross above and below the line
5	Using a ruler, draw a line between the points where the arcs intersect.

Algorithm: Constructing the perpendicular bisector of a line segment

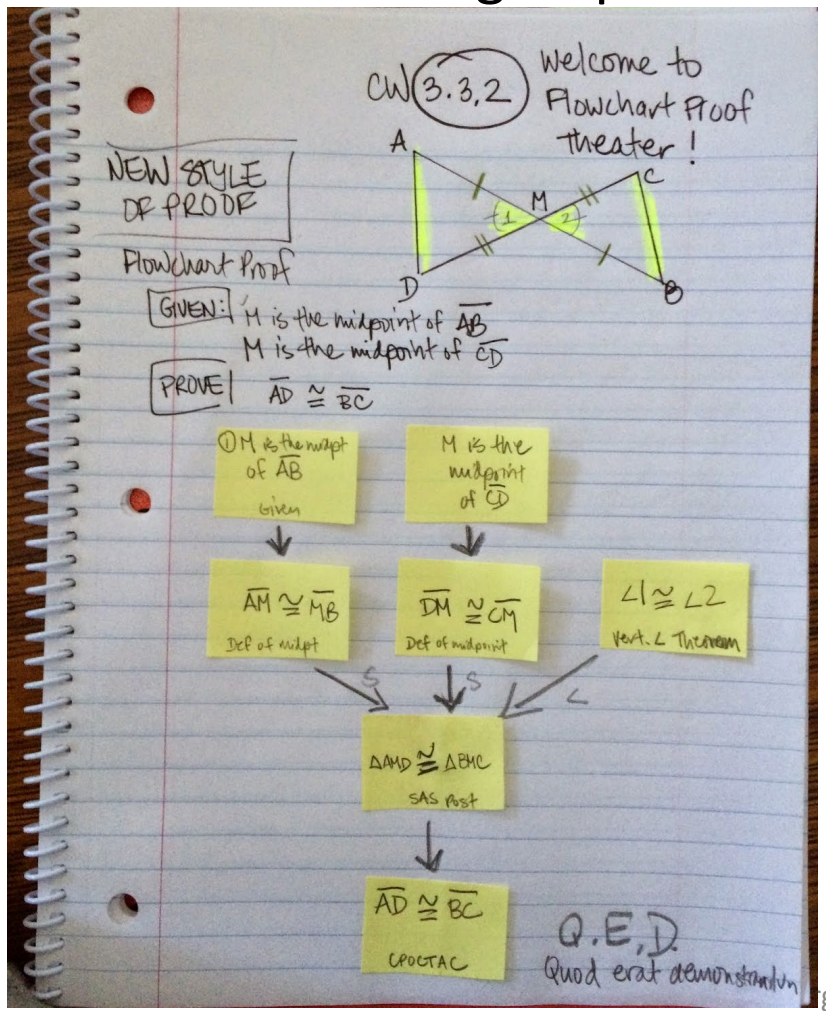
Compare the step by step approach to the flowchart approach.

Class discussion: which is better, justify responses. Class insights.

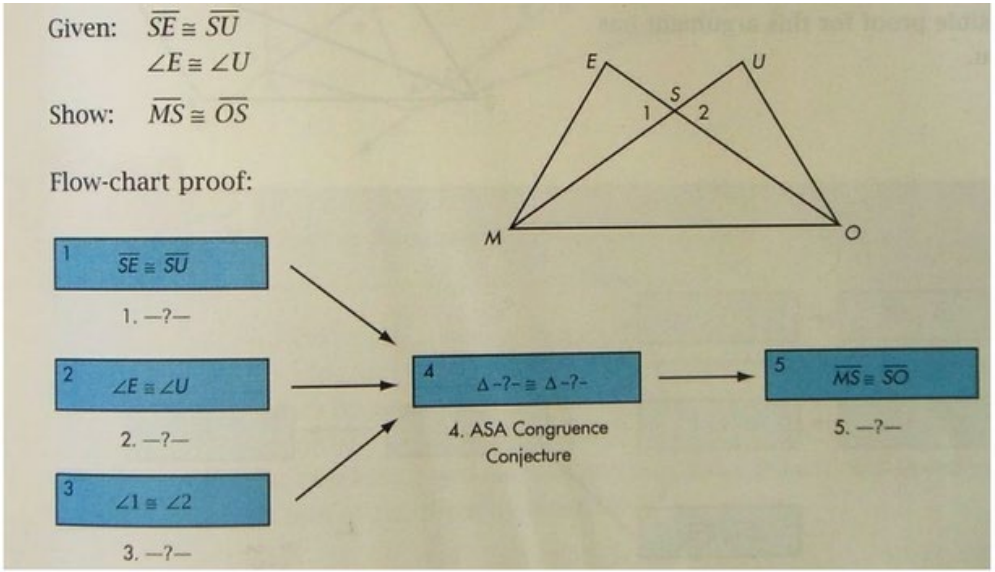


Group work Idea: Students construct & test each other's algorithms

Post-it flowchart algorithm construction for group work.



Fill in the blanks flowchart algorithm construction for group work.

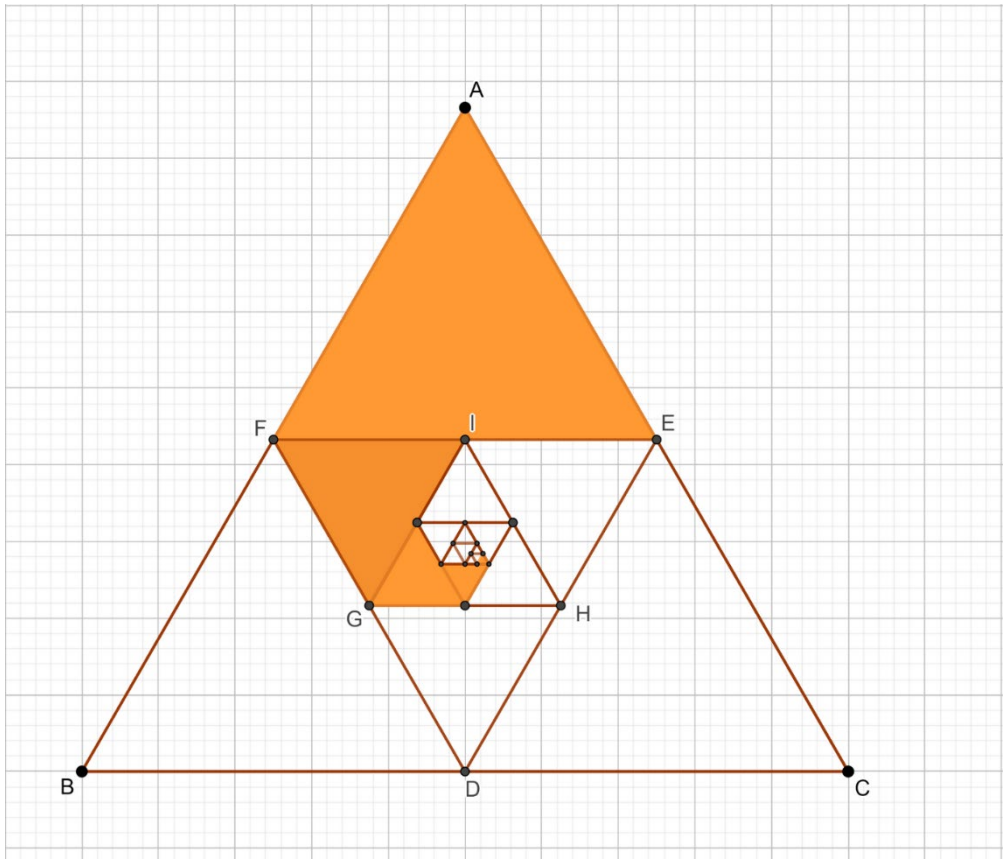


Y9 Space - design, test and refine **algorithms** involving a sequence of steps and decisions based on geometric constructions and theorems; **discuss and evaluate refinements** (VC2M9SP03)

Group work: Congruent Shape Activities – design the algorithm for the pinwheel design 1, 2 and 3

Y9 Space - design, test and refine **algorithms** involving a sequence of steps and decisions based on geometric constructions and theorems; **discuss and evaluate refinements** (VC2M9SP03)

Group work: Similar Shape Activities – design the algorithm for creating Sierpinksi's triangle.



YEAR 10

New Mathematics Version 2.0 curriculum
https://victoriancurriculum.vcaa.vic.edu.au/mathematics/mathematics-version-2-0/curriculum/f-10

Y10 Algebra implement algorithms that use data structures using pseudocode or a general purpose programming language (VC2M10A06)

Y10 Probability use the language of **'if ... then ...',** 'given', 'of' and 'knowing that' to investigate conditional statements and identify common mistakes in interpreting such language, and describe and interpret situations involving conditional probability; design and conduct simulations using digital tools to model conditional probability and interpret results (VC2M10P01)

Y10A Algebra devise and use algorithms and simulations to solve mathematical problems (VC2M10AA02)

Y10A Space design, test and refine solutions to spatial problems using algorithms and digital tools; communicate and justify solutions (VC2M10ASP06)

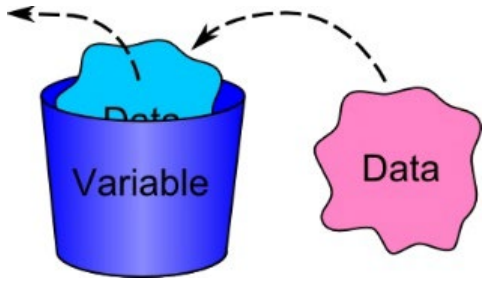
Y10 Algebra implement algorithms that use data structures using pseudocode or a general purpose programming language (VC2M10A06)

Variables – simple containers

- **A variable** holds one value for an algorithm or a computer program.
- The variable name is a way of labelling data with a descriptive name that is understood by the reader of the algorithm or computer program.
- It is helpful to think of variables as containers that hold data information.

```
Set the sum to zero
For i = 1 to 10 do
    Add i to the sum
Print sum
```


```
#Python variable
sum=0
```



Y10 Algebra implement algorithms that use data structures using pseudocode or a general purpose programming language (VC2M10A06)

Coding activity to engage with algorithmic thinking and control of repeated actions for summation of integers, squares and cubes.

Q: Find the missing terms and add terms in sequences shown

a) $1 + 2 + [\quad] + 4 + [\quad] + 6$
b) $1 + 4 + [\quad] + 16 + [\quad] + 36$
c) $1 + 8 + [\quad] + 64 + [\quad] + 216$

this is what the sum of cubes looks like

Y10 Algebra implement algorithms that use data structures using pseudocode or a general purpose programming language (VC2M10A06)

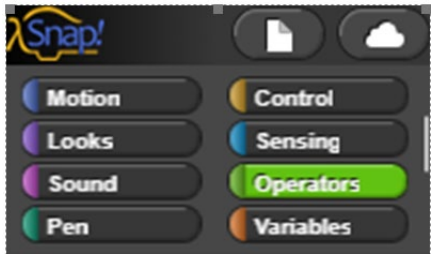
Coding activity to engage with algorithmic thinking and control of repeated actions for summation of integers, squares and cubes.

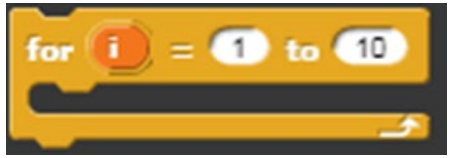
Pseudocode

```
Set the sum to zero
For i = 1 to 10 do
    Add i to the sum
Print sum
```

SNAP Implementation

SNAP variables, operations, actions





Y10 Algebra implement algorithms that use data structures using pseudocode or a general purpose programming language (VC2M10A06)

Coding activity to engage with algorithmic thinking and control of repeated actions for summation of integers, squares and cubes.

Pseudocode

```
Set the sum to zero
For i = 1 to 10 do
    Add i to the sum
Print sum
```

Python3 Implementation

```
#Python3
sum=0
for i in range(10 + 1):
    sum = sum + I
print(sum)
```

Y10 Algebra implement algorithms that use data structures using pseudocode or a general purpose programming language (VC2M10A06)

Coding activity to engage with algorithmic thinking and control of repeated actions for summation of integers, squares and cubes.

Pseudocode

```
Set the sum to zero
For i = 1 to 10 do
    Add i^3 to the sum
Print sum
```

Python3 Implementation

```
#Python3
sum=0
for i in range(10 + 1):
    sum = sum + i**3
print(sum)
```

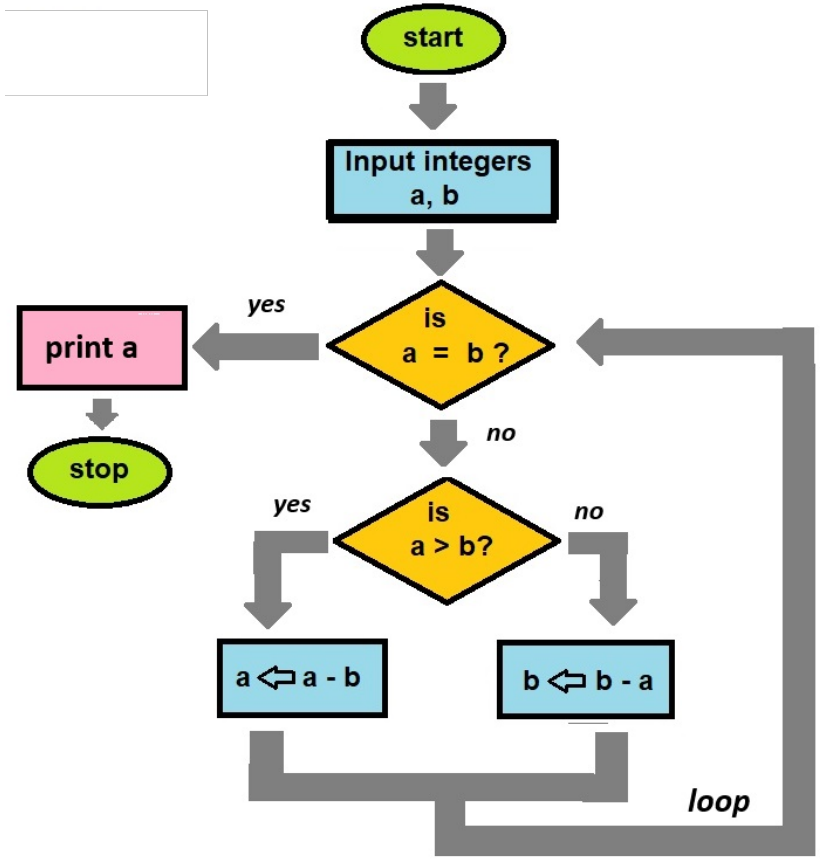
Y10 Algebra implement algorithms that use data structures using pseudocode or a general purpose programming language (VC2M10A06)

Greatest Common Divisor (GCD) – Euclid's Theorem circa 300 BC

Pseudocode Algorithm

```
Input a, b
while a ≠ b loop
  if a > b
     a := a – b
  else
     b := b – a
end loop
report a
```

Flowchart Algorithm



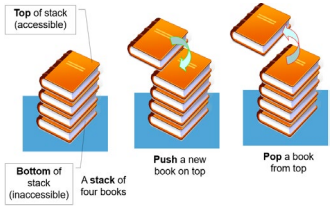
Variables: Lists and Arrays

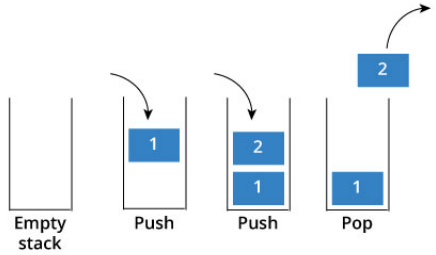
- **A List** is a collection of items, **Lists can grow and shrink** as items are added and deleted.
- **An Array** is a collection of items **fixed length structure** of items that can be accessed by index position.
- Coding languages such as Python conflate lists and arrays in their implementation.

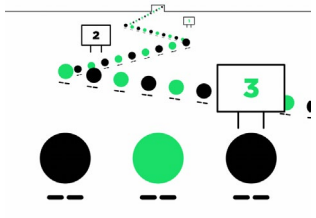

```
#Python3 List – example recording die values
diethrowList = [4,5,6,2,3,1,2,4,5,6]
print(diethrowList)
xyList = [[1,1],[2,4],[3,9],[4,16]]
print(len(xyList)) #prints length of list

#Python3 array – example profit each quarter
profitA = [1004,999,788,807]
print(profitA[0]) #prints first element of profitA
```

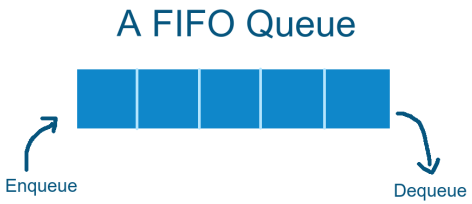
Variables: Stack & Queue & Priority Queue

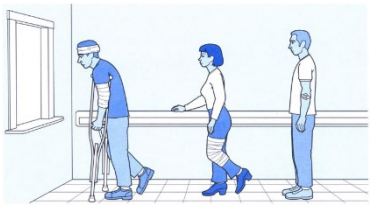
[Stack in Python – GeeksforGeeks]
[Queue in Python – GeeksforGeeks]
[Priority Queue in Python - GeeksforGeeks]







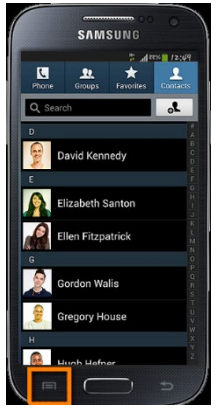


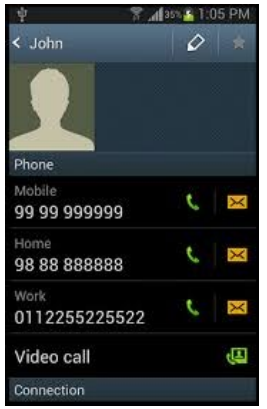


```
# Python3 Stack LIFO order using lists
stack=[]
stack.append(5)
stack.append(7)
stack.append(2)
print(stack)
for i in range(len(stack)):
  item = stack.pop() # last item LIFO
  print(item)

# Python3 Queue FIFO order using lists
queue = []
queue.append(5)
queue.append(7)
queue.append(2)
print(queue)
for i in range(len(queue)):
  item = queue.pop(0) # first item FIFO
  print(item)
```

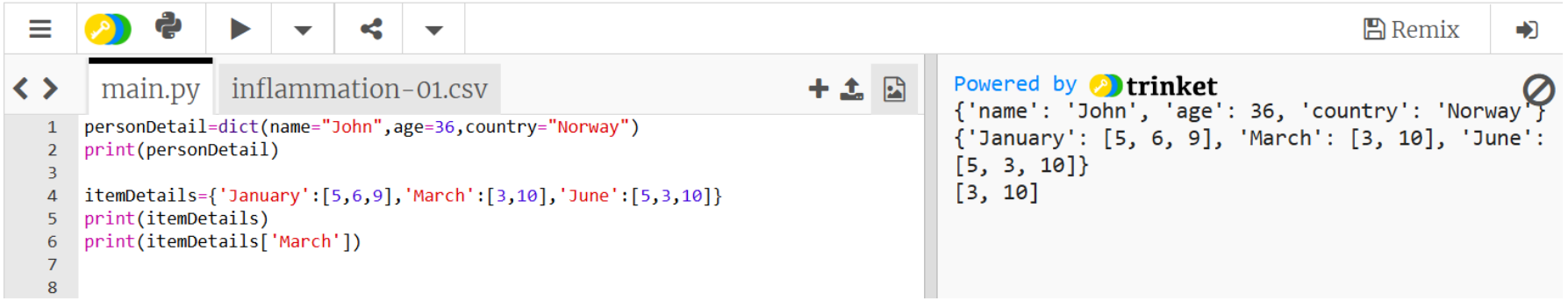
Dictionary

- Items stored in a Dictionary have two components, a key and a description.
- The description is accessed by the key.

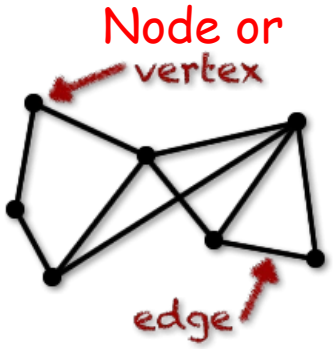





```
#Python3 dictionary examples
personDetail=dict(name="John",age=36,country="Norway")
print(personDetail)

itemDetails={'January':[5,6,9],'March':[3,10],'June':[5,3,10]}
print(itemDetails)
print(itemDetails['March'])
```



Network Graphs as variables

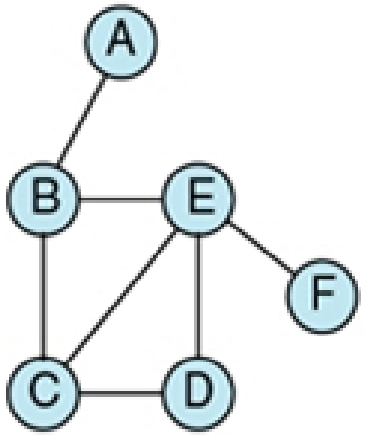


Usually expressed in Mathematical notation as sets

G={V,E}

V={set of nodes/vertices} e.g. **V={A,B,C,D,E,F}**

E={set of edges} e.g. **E={A-B,B-C,B-E,C-D,C-E,D-E,E-F}**




```
# Python3 graph structure defined with sets
vertices={'A','B','C','D','E','F'}
graph={'A':{'B'},'B':{'C','E'},'C':{'D','E'},'D':{'E'},'E':{'F'}}
print(graph)
```

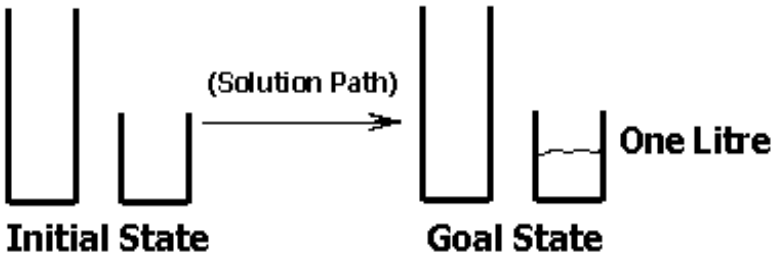
Die Hard movie – water jug, bomb diffuse scene

- https://youtu.be/2vdF6NASMiE

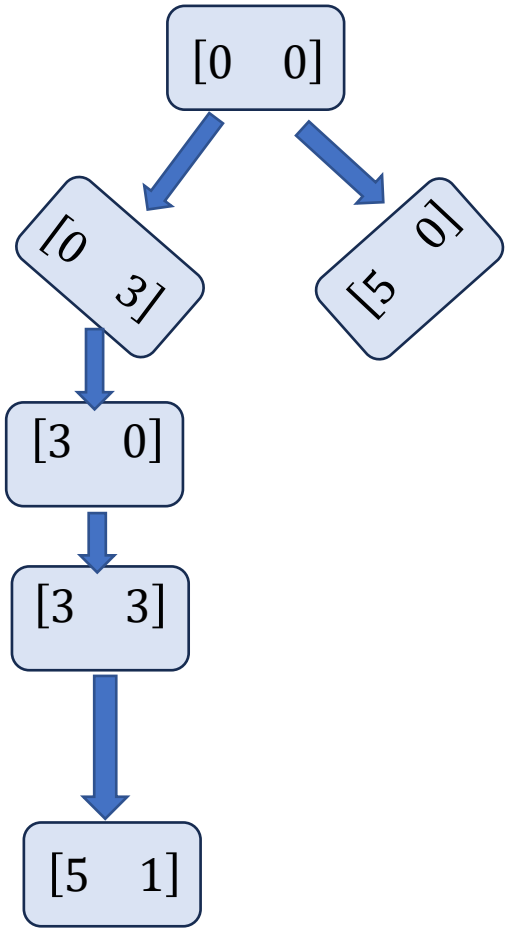
Y10 Algebra implement algorithms that use data structures using pseudocode or a general purpose programming language (VC2M10A06)

Water Jug Planning Problem

- Consider the following measuring liquid puzzle. Suppose you have a 3 litre jug and a 5 litre jug, and access to a tap with unlimited water, the jugs do not have measurement marks on them.
- How would you measure out exactly 1 litre of water?



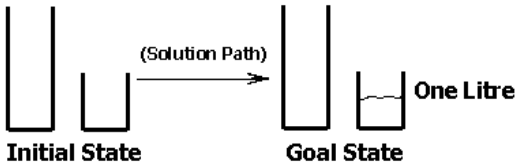
Array $[5L \quad 3L]$



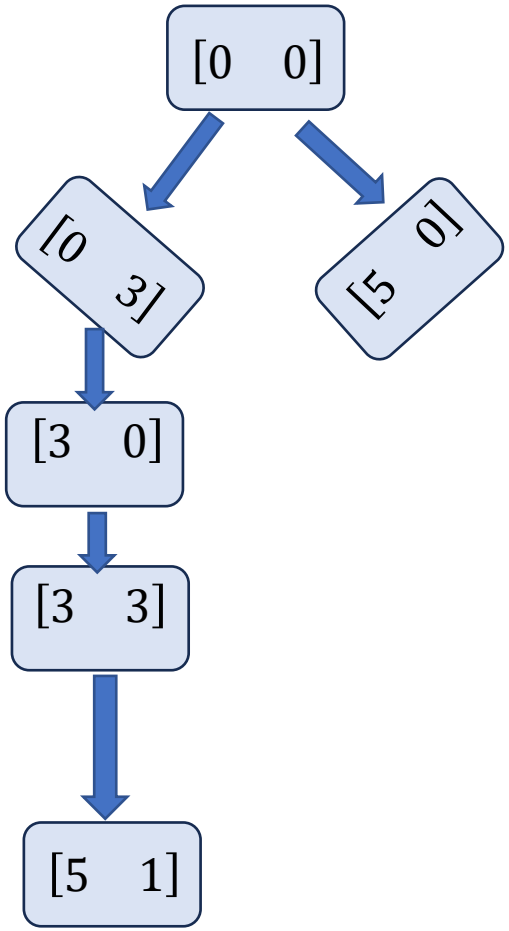
Y10 Algebra implement algorithms that use data structures using pseudocode or a general purpose programming language (VC2M10A06)

Water Jug Planning Problem

```
# Python3 graph structure defined with sets
planningGraph={'[0,0]':{'[0,3]','[5,0]'},
'[0,3]':{'[3,0]'},
'[3,0]':{'[3,3]'},
'[3,3]':{'[5,1]'}}
print(planningGraph)
```



Array $[5L \quad 3L]$

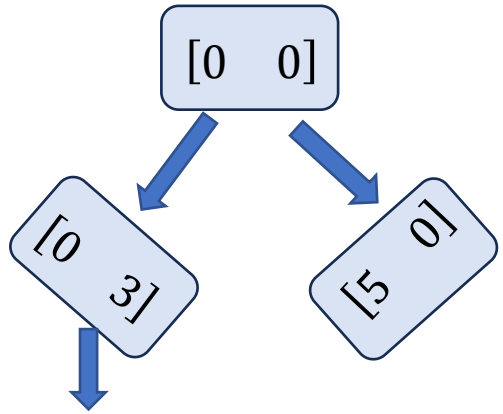


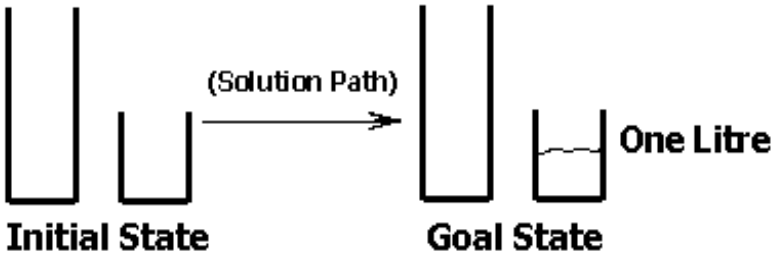
Y10 Algebra implement algorithms that use data structures using pseudocode or a general purpose programming language (VC2M10A06)

Water Jug Planning Problem

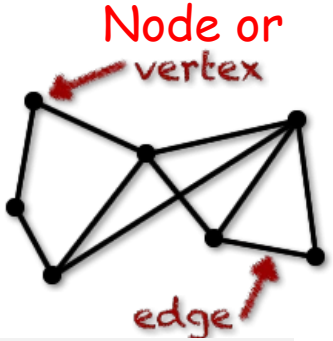
- Consider the following measuring liquid puzzle. Suppose you have a 3 litre jug and a 5 litre jug, and access to a tap with unlimited water, the jugs do not have measurement marks on them.
- How would you measure out exactly **2 or 4 litres** of water?

Array $[5L \quad 3L]$

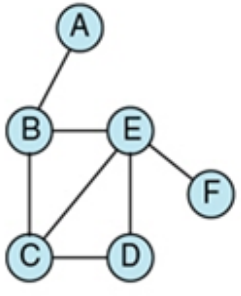




Network Graphs as variables



Adjacency Matrix alternative Network Graph representation - Examples



A
B
C
D
E
F

Vertex vector

	A	B	C	D	E	F
A	0	1	0	0	0	0
B	1	0	1	0	1	0
C	0	1	0	1	1	0
D	0	0	1	0	1	0
E	0	1	1	1	0	1
F	0	0	0	0	1	0

Adjacency matrix

```
# Python3 graph structure G defined
# with adjacency matrix
G=[[0,1,0,0,0,0],
[1,0,1,0,1,0],
[0,1,0,1,1,0],
[0,0,1,0,1,0],
[0,1,1,1,0,1],
[0,0,0,0,1,0]]
print(G)
print('G[5th row] =',G[4])
print('G[1st row][2nd column] =',G[0][1])
```

Y10A Space design, test and refine solutions to spatial problems using algorithms and digital tools; communicate and **justify solutions** (VC2M10ASP06)

```
#Python2
import turtle

def drawsquare(t,size):
  for i in range(4):
    t.forward(size)
    t.left(90)

def drawtriangle(t,size):
  for i in range(3):
    t.forward(size)
    t.left(120)

def drawpentagon(t,size):
  for i in range(5):
    t.forward(size)
    t.left(72)

def pencontrol(t,x,y,size,colour):
  t.penup()
  t.goto(x,y)
  t.pensize(size)
  t.color(colour)
  t.pendown()
```

```
# continued
# Create a turtle object
t = turtle.Turtle()
pencontrol(t,-100,50,3,'blue') # move to (-100,50)
for i in range(5):
  drawsquare(t, 100)
  t.left(5)
pencontrol(t,100,-50,3,'pink') # move to (100,-50)
drawpentagon(t,70)
pencontrol(t,-50,-100,3,'orange') # move to (-50,-100)
for j in range(30):
  drawtriangle(t, 100)
  t.left(12)
turtle.done()
```

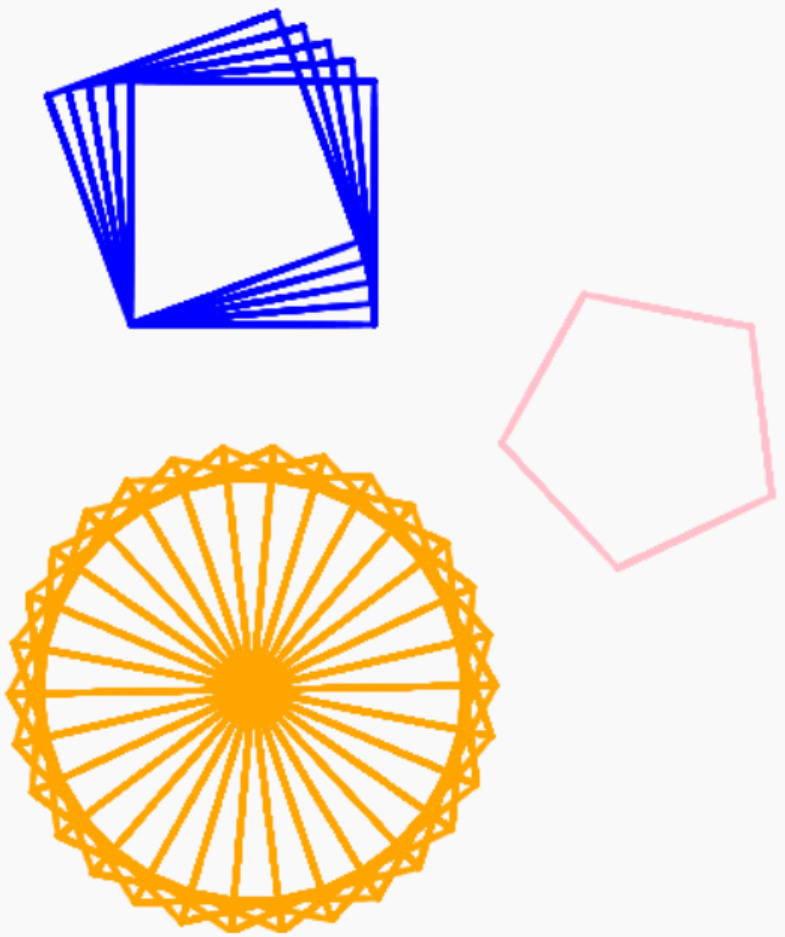
Y10A Space design, test and refine solutions to spatial problems using algorithms and digital tools; communicate and **justify solutions** (VC2M10ASP06)

trinket ▶ Run ? Modules Draft saved

main.py

```
1  import turtle
2  def drawsquare(t,size):
3    for i in range(4):
4      t.forward(size)
5      t.left(90)
6  def drawtriangle(t,size):
7    for i in range(3):
8      t.forward(size)
9      t.left(120)
10 def drawpentagon(t,size):
11   for i in range(5):
12     t.forward(size)
13     t.left(72)
14 def pencontrol(t,x,y,size,colour):
15   t.penup()
16   t.goto(x,y)
17   t.pensize(size)
18   t.color(colour)
19   t.pendown()
20
21 # Create a turtle object
22 t = turtle.Turtle()
23 pencontrol(t,-100,50,3,'blue') # move to (-100,50)
24 for i in range(5):
25   drawsquare(t, 100)
26   t.left(5)
27 pencontrol(t,100,-50,3,'pink') # move to (100,-50)
28 drawpentagon(t,70)
29 pencontrol(t,-50,-100,3,'orange') # move to (-50,-100)
30 for j in range(30):
31   drawtriangle(t, 100)
32   t.left(12)
33 turtle.done()
```

Result | Instructions



THE MATHEMATICAL ASSOCIATION OF VICTORIA

MAV24 CONFERENCE

Event App

App Download Instructions

Step 1: Download the App 'Arinex One' from the App Store or Google Play

App Store

Google Play

Step 2: Enter Event Code: **mav**

Step 3: Enter the email you registered with

Step 4: Enter the Passcode you receive via email and click 'Verify'. Please be sure to check your Junk Mail for the email, or see the Registration Desk if you require further assistance.

THE MATHEMATICAL ASSOCIATION OF VICTORIA

MAV24 CONFERENCE

Be in it to WIN!