Computational and
algorithmic thinking
in Mathematics

Unpacking the content descriptions

Level 9



Authorised and published by the Victorian Curriculum and Assessment Authority
Level 7, 2 Lonsdale Street
Melbourne VIC 3000

© Victorian Curriculum and Assessment Authority 2019

No part of this publication may be reproduced except as specified under the *Copyright Act 1968* or by permission from the VCAA. Excepting third-party elements, schools may use this resource in accordance with the [VCAA educational allowance](https://www.vcaa.vic.edu.au/Footer/Pages/Copyright.aspx#schools). For more information go to: <https://www.vcaa.vic.edu.au/Footer/Pages/Copyright.aspx>.

The VCAA provides the only official, up-to-date versions of VCAA publications. Details of updates can be found on the VCAA website: [www.vcaa.vic.edu.au](http://www.vcaa.vic.edu.au).

This publication may contain copyright material belonging to a third party. Every effort has been made to contact all copyright owners. If you believe that material in this publication is an infringement of your copyright, please email the Copyright Officer: vcaa.copyright@edumail.vic.gov.au

Copyright in materials appearing at any sites linked to this document rests with the copyright owner/s of those materials, subject to the Copyright Act. The VCAA recommends you refer to copyright statements at linked sites before using such materials.

The VCAA logo is a registered trademark of the Victorian Curriculum and Assessment Authority.

Contents

[Explicitly teaching computational and algorithmic thinking 4](#_Toc18415223)

[Overview of the resource 5](#_Toc18415224)

[Stuck in the middle with you 6](#_Toc18415225)

[Additional teaching resources 11](#_Toc18415226)

[Appendix 1 12](#_Toc18415227)

[Appendix 2 13](#_Toc18415228)

[Appendix 3 14](#_Toc18415229)

Explicitly teaching computational and
algorithmic thinking

The *Computational and algorithmic thinking – Unpacking the content descriptions* resources unpack the Victorian Curriculum F–10 Mathematics content descriptions that address computational thinking and algorithms at each level in the Patterns and algebra sub-strand of the Number and Algebra strand.

Each resource provides teachers with links between one Mathematics content description and extract from the achievement standard related to computational and algorithmic thinking and a teaching and learning activity that is designed to develop computational thinking and problem-solving skills and produce corresponding algorithms in a mathematical context. Teachers can also find excerpts from the Victorian Curriculum Mathematics and Digital Technologies glossaries in [Appendix 3](#Appendix3).

The resources have been developed with respect to teaching in the Mathematics learning area of the Victorian Curriculum and they also include suggestions how these activities could be extended to the Critical and Creating Thinking and Digital Technologies curriculums.

Teachers will find detailed ideas about how to integrate Mathematics with one or both of Digital Technologies and Critical and Creating Thinking in [Appendix 1](#Appendix1) and [Appendix 2](#Appendix2) respectively.

Overview of the resource

**Curriculum area and level:** Mathematics, Level 9

**Strand and sub-strand:** Number and Algebra, Patterns and algebra

**Content description:** Apply set structures to solve real-world problems [(VCMNA037)](http://victoriancurriculum.vcaa.vic.edu.au/Curriculum/ContentDescription/VCMNA307)

**Achievement standard (extract):** [Students solve problems]… using a range of strategies including the use of digital technology.

**Resource title:** Stuck in the middle with you

**Timing:** 2 or 3 lessons (approx. 100­–150 minutes)

**Description:** Students use sorting algorithms to facilitate finding the median of a set of numbers.

This activity can be completed ‘unplugged’, using flowcharts, diagrams or other visual representations of the sorting process. Alternatively, the activity may be conducted using a programming language such as [Python](https://www.python.org/) or [Wolfram Language](https://www.wolfram.com/language/) to implement the sorting algorithm with collected data.

An important aspect of this activity is that the algorithm is used to solve real-world problems. The emphasis in this activity is on students collecting real-life data to sort ­­– either primary data collected directly from the source or secondary data from experiments or research online – and then discussing the medians found in the context of this data.

**Learning objectives:** Students can:

* use a sorting algorithm to sort numbers and hence find the median of a set (either with or without the use of digital technology)
* describe some sorting algorithms, including how they work, advantages and disadvantages of both, and how they are similar or different from each other.

**Resources:** Each student or student pair will need a computer and access to the internet for the research component.

For their presentation, students may use digital technologies, including video, or they may use paper, posters or a live demonstration and explanation/presentation. The sorting algorithms can be demonstrated on paper, with models or with the use of a programming language.

Stuck in the middle with you

Activity 1 – Let’s sort it out!

For Activity 1, students must research at least two sorting algorithms and answer the following questions:

* Why do we sort?
* What is a sorting algorithm?
* What does it do?
* What are some different types of sorting algorithm? How do they work?

Give students time to conduct online research in pairs to find one or two sorting algorithms. Students should then present their findings on their chosen algorithms. Students could create a video or visual presentation demonstrating how their chosen algorithms are used to sort numbers or objects (for example, objects of different heights or weights), and to compare and contrast the different algorithms.

Scaffold and support

A common sorting algorithm is the bubble sort algorithm. Students could view this [short YouTube video](https://www.youtube.com/watch?v=MtcrEhrt_K0), which uses LEGO and stop-motion animation to demonstrate this algorithm.

Teachers might like to show students this [YouTube video](https://youtu.be/aXXWXz5rF64) of an animated robot using the bubble sort to sort coloured balls. This video also demonstrates the quick sort algorithm versus the bubble sort algorithm, and students will see that the number of comparisons and the time taken to sort are both important considerations for sorting algorithms.

Students could refer to this [Sorting Algorithms Animations webpage](https://www.toptal.com/developers/sorting-algorithms), which uses a table of animations for different sorting algorithms to visually compare the method and speed of each algorithm.

**Tip:** Teachers are encouraged to use and clearly define the term ‘algorithm’ in this activity, to reinforce the process of following a set or sequence of instructions to solve a problem (see the [Victorian Curriculum Mathematics Glossary](http://victoriancurriculum.vcaa.vic.edu.au/LearningArea/LoadFile?learningArea=mathematics&subject=mathematics&name=Mathematics%20Glossary.docx&storage=Glossary) or [Appendix 3](#Appendix3)). Teachers could give students the opportunity to come up with their own agreed class definition before, during or after the activity, and then compare it with the definition in the glossary.

**Discussion prompts**

* Why do we need to sort a list of numerical data?
* How are the sorting algorithms similar? How are they different?
* Is one algorithm better suited to certain types of data? Why do you think that is?
* Where are sorting algorithms used? (Example answer: when searching for data.)

Once students have completed their research, have each pair report back to the class and demonstrate one of their algorithms, reflecting on the advantages and disadvantages of this sorting method. Students could do this by demonstrating the sorting algorithm in person (much like the LEGO or robot animations) while describing the process, or they could use some other visual representation.

Before moving on to the next activity, reflect with the class on their findings about the various algorithms.

**Tip:** As scaffolding for this activity, you could consider the problem of sorting a collection of mixed Australian coins.

Challenge and extend

**Programming sorting algorithms**

Students with Python experience who are interested in programming could work through the [interactive Python lesson](http://interactivepython.org/runestone/static/pythonds/SortSearch/TheBubbleSort.html) on the bubble sort.

They could also work through this [tutorial](https://www.tutorialspoint.com/python/python_sorting_algorithms.htm) demonstrating Python sorting algorithm code.

► **Cross-curricular links**

See [Appendix 1](#Appendix1) and [Appendix 2](#Appendix2) for ways to link this activity to the explicit teaching of Digital Technologies and/or Critical and Creative Thinking.

Activity 2 – In the middle

For Activity 2, students will be searching online to find real-world data to sort using their algorithm, sorting either by hand or using digital technology.

Before doing this, students could conduct a simple sorting activity by hand. Sorting involves comparing one value with another in a list. If the first number is larger than the second then the numbers must swap places. Now the second number is compared against the third, etc. Several passes are made to the list to ensure the numbers are sorted.

Now encourage students to select a topic in pairs or small groups. Students should decide on an appropriate number of data values to sort, depending on whether they are sorting by hand or writing a program using digital technologies.

In addition to deciding on a sorting algorithm, students will need to clearly define (visually or in writing) the algorithm (process) they will use to find the median for a set of numbers.

**Tip:** For data, students could go to [Choose Your Own Statistics](http://education.abc.net.au/statistics-game/#/) (ABC Splash), which is a great resource for students who want to explore issues and data relating to topics such as weekly wages, life expectancy, and asylum seekers and refugees. To use the site, click on a topic to view a visual summary and then use the menu along the top, clicking ‘Interactive graph’ to modify data filters, ‘More Information’ for background and context, and ‘Data Table’ for data summaries.

[Australian Bureau of Statistics – Data by Region](http://stat.abs.gov.au/itt/r.jsp?databyregion&ref=CTA2) is another good source of data. Students can select a region and scroll down the page to investigate many topics.

Scaffold and support

Revise the definition of the median and how to find it.

The ‘median’ is the value in a set of ordered data that divides the data into two equal parts. It is frequently called the ‘middle value’. Where the number of observations is odd, the median is simply the middle value. For an even set, the median is taken to be the average of the two middle values. For example, the median of the numbers 1, 3, 4, 5, 7 is 4, and the median for 1, 3, 4, 5 is the average of the two middle values – that is, (3 + 4) ÷ 2 = 3.5.

The median provides a measure of location of a dataset that is suitable for both symmetric and skewed distributions and is also less sensitive to outliers than the mean.

**Tip:** Connect data collection to real-world activities. For example, consider collecting ‘beep test’ data from your class during Physical Education. The median might show that 50% of students are scoring below the National Standard for the beep test. Students could then hypothesise why this might be the case and suggest future investigation or data analysis to explore their hypothesis further.

Example – Life expectancy in Australia

A group chooses ‘Life Expectancy’ from the [Choose Your Own Statistics web resource](http://education.abc.net.au/statistics-game/#/) and uses the first of the two data tables, which contains the average life expectancy in Australia by state or territory, sex and Indigenous status.

Students decide to find the median average life expectancy for males and females across all states and territories, using part of the data table (shown below).

| **Location** | **Total male (years)** | **Total female (years)** |
| --- | --- | --- |
| NSW | 79.5 | 82.9 |
| Victoria | 79.5 | 82.9 |
| Queensland | 78.9 | 82.6 |
| South Australia | 79.5 | 83.2 |
| Western Australia | 79.5 | 83.2 |
| Tasmania | 79.5 | 82.9 |
| Northern Territory | 74.3 | 78.9 |
| ACT | 79.5 | 82.9 |

Students use their chosen sorting algorithm to sort each set of data. For example, for the Male data they produce the following:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Total male (years) (unsorted)** | 79.5 | 79.5 | 78.9 | 79.5 | 79.5 | 79.5 | 74.3 | 79.5 |

**Sorting algorithm.** (Students to show this process.)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Total male (years) (sorted)** | 74.3 | 78.9 | 79.5 | 7.95 | 79.5 | 79.5 | 79.5 | 79.5 |

**Algorithm for how to find the median. (**Students to show this process.)

Median average life expectancy for Australian males is 79.5 years.

Once students sort their data and find the median, they reflect on what this means with respect to the context of the data and then present their findings.

**Discussion prompts**

* What does this median mean?
* How does it compare with medians for other categories (e.g. in the example given, life expectancy for females)?
* Why do you think the median you found is higher or lower than another category?
* Was it easy to sort your data using your sorting algorithm? Why/why not?

Challenge and extend

Students could extend this investigation by collecting real-life data from experiments or from their class (with permission) via measurements or surveys. This data could then be analysed and even compared with averages from ABC Splash or the Australian Bureau of Statistics regional data.

Note: Sensitivity towards different backgrounds and experiences should always be considered when researching and presenting findings on research topics. Discuss relevant issues around collecting sensitive data from others before the activity, so students are mindful of this as they work.

► **Cross-curricular links**

See [Appendix 1](#Appendix1) and [Appendix 2](#Appendix2) for ways to link this activity to the explicit teaching of Digital Technologies and/or Critical and Creative Thinking.

Reflection

Students share their findings from their research with others, compare their main findings and reflect on these findings.

With the class, the teacher could ask the students to discuss the difficulties they encountered and discuss different strategies that students used to overcome these.

* What were some difficulties you faced in this task?
* How did you overcome these? (What were some strategies you used?)
* What were some skills you learnt that helped in a similar situation?
* Can you think of other problems where a sorting algorithm might be used to help solve a complicated problem?

With the class, in small groups or individually, the teacher should ask students to reflect on their algorithmic thinking. Ask students:

* How does your algorithm work in each activity?
* What is the purpose of the sorting algorithm you created?
* What does your algorithm do well? Explain your reasoning.
* What doesn’t it do so well? Why?
* What do you think was the purpose of this activity?
* Which of two algorithms is more efficient?

**Tip:** Reflection is a very important part of any computational thinking–focused activity because it encourages students to consider the different aspects of the task, such as defining the problem, selection of tools and processes, problem-solving, teamwork and verifying their solution. This helps students reflect on the process of their own learning (meta-learning) and how the skills and tools they have used might be transferred to other contexts.

Additional teaching resources

* Try Computing has a range of [lesson plans](http://www.trycomputing.org/lesson-plans/fun-sorting-lesson) for various computer science activities. The sorting activity has a full lesson plan and teacher resources.
* There are many activities with lesson plans and accompanying resources at NRICH, including [statistical analysis activities](https://nrich.maths.org/8490) exploring data and the real world.
* The website for the reSolve: Maths by Inquiry project provides activities that encourage students to be critical and creative thinkers. The [Hot Streaks](https://www.resolve.edu.au/hot-streaks-trial) activity is for students who are familiar with using spreadsheet programs to analyse data and who are ready to create their own methodologies to explore more complex datasets. Students use statistical tools to distinguish between randomly generated and non-random results, using first a small then very large dataset collected from practical experiments.
* The online resource Yenka allows students to create 3D mathematical models to demonstrate statistics and probability. The [statistics activities](https://www.yenka.com/maths/) are particularly well suited to analysing real-world data.
* See also [Unpacking Digital Technologies Content Descriptions](https://www.vcaa.vic.edu.au/Documents/viccurric/digitech/Unpacking_the_Content_Descriptions/Unpacking_Digital_Technologies_Content_Descriptions-9-10.docx) for Levels 9 and 10.

Appendix 1

Suggestions for explicitly teaching Digital Technologies (stimulus only)

**Strand:** Creating Digital Solutions

**Band:** Levels 9 and 10

**Content description:** Design algorithms represented diagrammatically and in structured English and validate algorithms and programs through tracing and test cases [(VCDTCD052)](http://victoriancurriculum.vcaa.vic.edu.au/Curriculum/ContentDescription/VCDTCD052)

**Achievement standard (extract):** They design … algorithms … and test modular programs …

**Suggestions that link to the activities:**

* Creating algorithms that use various functions and data structures.
* Testing the expected output of algorithms using trace tables and desk checking if necessary in order to make modifications and record results.

See also [Unpacking Digital Technologies Content Descriptions](https://www.vcaa.vic.edu.au/Documents/viccurric/digitech/Unpacking_the_Content_Descriptions/Unpacking_Digital_Technologies_Content_Descriptions-9-10.docx) for Levels 9 and 10.

Appendix 2

Suggestions for explicitly teaching Critical and Creative Thinking (stimulus only)

**Strand:** Questions and possibilities

**Band:** Levels 9 and 10

**Content description:** Investigate the characteristics of effective questions in different contexts to examine information and test possibilities ([VCCCTQ043](http://victoriancurriculum.vcaa.vic.edu.au/Curriculum/ContentDescription/VCCCTQ043))

**Achievement standard (extract):** … students construct and evaluate questions, including their own, for their effectiveness.

**Suggestion that links to Activity 1:**

* Discussing how the selection of a sorting algorithm might change to suit different contexts, for example, the size of a dataset or the types of quantities being sorted.

**Strand:** Reasoning

**Band:** Levels 9 and 10

**Content description:** Examine a range of rhetorical devices and reasoning errors, including false dichotomies and begging the question ([VCCCTR046](http://victoriancurriculum.vcaa.vic.edu.au/Curriculum/ContentDescription/VCCCTR046))

**Achievement standard (extract):** They explain and apply a range of techniques to test validity within and between arguments

**Suggestion that links to Activity 2:**

* Analysing a range of examples to explore when particular forms of reasoning are legitimate and when they are fallacious. For example: When should the median be used? What type of data might be more suited to using the median than the mean? Why might the mean be used and reported (for example, in news articles)? Are statistics and data analysis always presented in a subjective way (with a bias)? Why might this be problematic?

**Strand:** Meta-Cognition

**Band:** Levels 9 and 10

**Content description:** Critically examine their own and others’ thinking processes and discuss factors that influence thinking, including cognitive biases ([VCCCTM051](http://victoriancurriculum.vcaa.vic.edu.au/Curriculum/ContentDescription/VCCCTM051))

**Achievement standard (extract):** Students identify, articulate, analyse and reflect on their own and others’ thinking processes.

**Suggestion that links to Activity 2:**

* Researching, comparing and reflecting (before or after the activity) on thinking processes used by professionals in a range of areas, for example, computer scientists, journalists, data scientists or mathematicians.

Appendix 3

Excerpts from the Victorian Curriculum Mathematics and Digital Technologies glossaries

There are some commonalities in the terms used when explicitly teaching computational and algorithmic thinking in Mathematics and Digital Technologies; however, there are also some subtle but important differences in the definitions of terms. Some of these common terms and their definitions are listed below, under the two different curriculum areas.

If you are going to explicitly teach Mathematics, please refer to the [Victorian Curriculum Mathematics Glossary](http://victoriancurriculum.vcaa.vic.edu.au/LearningArea/LoadFile?learningArea=mathematics&subject=mathematics&name=Mathematics%20Glossary.docx&storage=Glossary). If you are also going to explicitly teach Digital Technologies, refer to the [Victorian Curriculum Digital Technologies Glossary](https://victoriancurriculum.vcaa.vic.edu.au/LearningArea/LoadFile?learningArea=technologies&subject=digital-technologies&name=Digital%20Technologies%20Glossary.docx&storage=Glossary).

Mathematics

Algorithm

An **algorithm** is a process that can be carried out mechanically, using a well-defined set of instructions, to perform a particular task or solve a type of problem. Examples of mathematical algorithms include processes for tasks such as ordering a set of numbers from smallest to largest, multiplying many-digit decimal numbers, factorising linear expressions, determining which of two fractions is larger, bisecting an angle, or calculating the mean of a set of numbers.

Algorithmic thinking

**Algorithmic thinking** is the type of thinking required to design, test and evaluate problem-solving processes in a systematic way, using algorithms.

Coding

A process by which algorithms are represented for implementation. For computers, this is done using a coding language such as block coding, C++, JavaScript, Python, Wolfram Language.

Computational thinking

In this context, computational thinking is considered to be linked to algorithmic thinking. This type of thinking is usually considered specific to computers which involves solving problems, designing systems and implementation.

Sequence (number)

A **sequence** is an ordered set of elements such as numbers, instructions or objects. From an algorithmic point of view, a sequence is an ordered set of instructions or actions.

Unplugged

A commonly used term for computational thinking activities carried out without digital technology. “Unplugged” representations of algorithms may include structured mathematical processes, English representations (steps) or flowcharts.

Digital Technologies

Algorithm

A description of the steps and decisions required to solve a problem. For example, to find the largest number in a list of positive numbers:

1. Note the first number as the largest
2. Look through the remaining numbers, in turn, and if a number is larger than the number found in 1, note it as the largest.
3. Repeat this process until complete. The last noted number is the largest in the list.

Flowcharts are often useful in visualising an algorithm.

Computational thinking

A problem-solving method that involves various techniques and strategies in order to solve problems that can be implemented by digital systems, such as organising data logically, breaking down problems into components, and the design and use of algorithms, patterns and models.

Structured English

The use of the English language to describe the steps of an algorithmin clear, unambiguous statements that can be read from start to finish. The use of keywords, such as START, END, IF, UNTIL, provides a syntax similar to that of a programming language to assist with identifying logical steps necessary to properly describe the algorithm.

An example of the use of structured language can be demonstrated using the following problem:

Description of the problem:

Describing the decision a person makes about how to get to a destination based on the weather and the distance from their current location to their destination.

Structured English example:

START

IF it is raining outside THEN

 Catch the bus

ELSE

 IF it is less than 2km to the destination THEN

 Walk

 ELSE IF it is less than 10km to the destination THEN

 Ride a bicycle

 ELSE

 Catch the bus

 ENDIF

ENDIF

END

The Structured English description can easily be translated into code using a programming language and accurately captures the logical elements that must be followed to answer the question posed.