Stage 6 Software Design and Development

# Representing sorting algorithms

This resource assists students to understand that any process can be described with an algorithm. It also demonstrates that algorithms can be expressed in different ways.

By focusing upon the control structures used to manipulate and process data within an array students interpret and analyse basic sorting techniques to better understand this fundamental and important function of computing. Students are provided the opportunity to use industry standard data modelling tools to represent the logic and flow of data within an algorithm or scenario.

Teachers may consider an extension activity of providing a Structure chart through Lucidchart and allocating students to model with algorithm flow charts each module of the Structure chart as would be achieved in industry to produce modular solutions.

## Outcomes

* **H1.2** differentiates between various methods used to construct software solutions
* **H4.2** applies appropriate development methods to solve software problems
* **H4.3** applies a modular approach to implement well-structured software solutions and evaluates their effectiveness

[Software Design and Development Stage 6 Syllabus (2010)](https://educationstandards.nsw.edu.au/wps/portal/nesa/11-12/stage-6-learning-areas/technologies/software-design-development) © NSW Education Standards Authority (NESA) for and on behalf of the Crown in right of the State of New South Wales, 2010.

## Delivery strategies

These activities were developed for Stage 6 Software Design and Development students with limited experience in coding. They are readily adapted for the Stage 5 Information Software and Technology programming unit.

This resource is adaptable for teachers to use with online platforms such as Google classroom. Links to the videos and websites could be posted for students to access during learning at home. The activities and questions could be set as classwork documents within Google classroom, which the students complete and submit for a grade or feedback. Alternatively, students could receive the activities and questions as worksheets to complete and submit at a later date.

The suggested activities and questions that relate to the listed resources are not an exhaustive list and may be added to or edited to suit your students learning needs.

Students are to refer to Software Design and Development Stage 6 Syllabus (2010) and the corresponding [Software Design and Development Stage 6 Syllabus (2010):](https://educationstandards.nsw.edu.au/wps/portal/nesa/11-12/stage-6-learning-areas/technologies/software-design-development) Course specifications (PDF, 87 pages, 2.6 MB).

Students should practise writing translating their code back into the pseudocode required by NESA.

## Sorting arrays

|  |  |
| --- | --- |
| Focus area | Sorting arrays |
| Syllabus content | **Understanding the problem (Year 11)**   * identification of inputs and required outputs * determining the steps that, when carried out, will solve the problem * Input Process Output (IPO) diagrams   **Structured algorithms (Year 11)**   * control structures which form the basic building blocks of all algorithms:   + sequence   + selection   + repetition (pre-test, post-test), including for…next loops * methods for representing algorithms:   + pseudocode   + flowcharts incorporating standard control structures * interpret and create algorithms represented in both pseudocode and flowcharts that use standard control structures * gather solutions from a number of sources and modify them to form an appropriate solution to a specified problem * represent code from different sources as an algorithm to assist in understanding its purpose and to assess its relevance in a proposed solution   **Standard algorithms (Year 12)**   * standard logic used in software solutions, namely:   + bubble sort   + insertion sort   + selection sort   (see Course Specifications document) |
| Resources | [Software and Course Specifications for Software Design and Development](https://educationstandards.nsw.edu.au/wps/portal/nesa/11-12/stage-6-learning-areas/technologies/software-design-development) Higher School Certificate 2012 (PDF, 87 pages, 2.6 MB) |

### Insertion sort

Watch [Insert-sort with Romanian folk dance](https://www.youtube.com/watch?v=ROalU379l3U) (duration 4:03)

What does each dancer represent?

|  |
| --- |

Look closely at how each data item interacts with its neighbour.

Describe the movements of each dancer (data item) within the array.

|  |
| --- |

Translate these movements using control structures (sequence, selection, iteration) into pseudocode to sort this array. Complete the process column of the Input, Process Output (IPO) chart by writing your pseudocode here. Annotate your pseudocode to demonstrate you understanding that this represents the dancers’ movements.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 10 unsorted number | Begin **insertion** sort | **10 sorted numbers** |

### Selection sort

Watch [Select-sort with gypsy folk dance](https://www.youtube.com/watch?v=Ns4TPTC8whw) (duration: 7:06)

Look closely at how each data item interacts with its neighbour.

Describe the movements of each dancer (data item) within the array.

|  |
| --- |

Translate these movements using control structures (sequence, selection, iteration) into pseudocode to sort this array. Complete the process column of the I P O chart by writing your pseudocode here. Annotate your pseudocode to demonstrate you understanding that this represents the dancers’ movements.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 10 unsorted number | Begin **select** sort | **10 sorted numbers** |

### Bubble sort

Watch [Bubble sort with Hungarian ("Csango") folk dance](https://www.youtube.com/watch?v=lyZQPjUT5B4) (duration: 5:15)

Look closely at how each data item interacts with its neighbour.

Describe the movements of each dancer (data item) within the array.

|  |
| --- |

Translate these movements using control structures (sequence, selection, iteration) into pseudocode to sort this array. Complete the process column of the I P O chart by writing your pseudocode here. Annotate your pseudocode to demonstrate you understanding that this represents the dancers’ movements.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 10 unsorted number | Begin **bubble** sort | **10 sorted numbers** |

## Research

What other sorting algorithms are commonly used?

|  |
| --- |

Compare the [efficiencies of the sorting algorithms](https://www.toptal.com/developers/sorting-algorithms)

1. Which is fastest?
2. Which is slowest?
3. Why is this important?

|  |
| --- |

Explain why sorting algorithms are an important part of computer science.

|  |
| --- |

What other ways could you describe an algorithm?

|  |
| --- |

## Representing algorithms

|  |  |
| --- | --- |
| Focus area | Methods for representing algorithms |
| Syllabus content | * identify control structures in an algorithm * interpret and create algorithms represented in both pseudocode and flowcharts that use standard control structures * detect logic errors in an algorithm by performing a desk check * represent code from different sources as an algorithm to assist in understanding its purpose and to assess its relevance in a proposed solution |
| Resources | Software Design and Development (SDD) 2019 HSC Examination and use links to access additional previous years.   * Software and Course Specifications for Software Design and Development * Lucidchart– access website through Google Apps for Education. * Lucidchart Basic Tutorials playlist. |
| Activities and or questions | 1. Go through SDD examination papers and select and collate all of the questions based on either algorithms or scenarios. 2. For selected algorithm or scenario:    * identify the variables or data structures (array or file)    * identify the control structures present or required    * perform desk check to observe variables changing and demonstrate understanding of the flow of logic    * and where applicable translate the algorithm from a flowchart into pseudocode or vice versa.    * write any required code to complete the question 3. Select the most appropriate tool(s) to visually represent the algorithm or scenario. The SDD course specifications should guide your choice. 4. Using your @education account, access Lucidchart in Google Apps for Education. 5. Use the tools in Lucidchart to visually describe the algorithm or scenario. Alternatively, use software of your choice to describe algorithms. 6. Export these finished models to a presentation that includes the text in the original algorithm or scenario. 7. (Optional) Provide an answer to the question 8. Submit this for teacher and/or peer review. |