

DOWNLOADED FROM (AS91885): http://ncea.tki.org.nz/Resources-for-Internally-Assessed-Achievement-Standards/Technology/Digital-technologies/Level-1-Digital-technologies-assessment-resources

**Internal Assessment Resource**

**Digital Technologies & Hangarau Matihiko Level 1**

This resource supports assessment against Achievement Standard 91885[[1]](#footnote-1)

**Standard title:** Demonstrate understanding of searching and sorting algorithms

**Credits:** 3

**Resource title:** Let’s Investigate!

**Resource reference:** Digital Technologies & Hangarau Matihiko 1.9B Version 1

|  |
| --- |
| This resource:   * Clarifies the requirements of the achievement standard * Supports good assessment practice * Should be subjected to the school’s usual assessment quality assurance process * Should be modified to make the context relevant to students in their school/kura environment and ensure that submitted evidence is authentic |

|  |  |
| --- | --- |
| Date version published by Ministry of Education | December 2017 Version 1  To support internal assessment from 2018 |
| Authenticity of evidence | Teachers/Kaiako must manage authenticity for any assessment from a public source, because students may have access to the assessment schedule or student/ākonga exemplar material.  Using this assessment resource without modification may mean that students’ work is not authentic. The teacher/kaiako may need to change figures, measurements or data sources or set a different context or topic to be investigated or a different text to read or perform. |

**Internal Assessment Resource**

**Achievement standard:** 91885

**Standard title:**  Demonstrate understanding of searching and sorting algorithms

**Credits:** 3

**Resource title:** Let’s Investigate!

**Resource reference:** Digital Technologies & Hangarau Matihiko1.9B Version 1

**Teacher/Kaiako guidelines**

The following guidelines are supplied to enable teachers/kaiako to carry out valid and consistent assessment using this internal assessment resource.

Teachers/kaiako need to be very familiar with the outcome being assessed by the achievement standard. The achievement criteria and the explanatory notes contain information, definitions, and requirements that are crucial when interpreting the standard and assessing students/ākonga against it.

**Context/Te Horopaki**

This activity requires students to work through a set of tasks, guided by their teacher, and finally do an evaluation, to demonstrate their comprehensive understanding of searching and sorting algorithms. Each task can be supported by the teacher with structured lessons, discussions and experiments. Alternatively the students could use this as an inquiry and could be guided through by the teacher. The tasks listed below provide a framework for this.

Students will collect evidence of them carrying out at least one searching algorithm and one sorting algorithm.

1. Students will be asked to provide evidence of an investigation into practical uses of searching and sorting algorithms in the real world. In this investigation they should outline their understanding of the purpose of these applications and the reasons why they are important. They should also discuss the relationship between searching and sorting algorithms and any implications that the use of these algorithms may have for the user.
2. Students will then carry out a demonstration of a search algorithm selected from either linear or binary. The teacher can choose from a variety of methods for this including cards, cups, phonebooks or other unplugged resources like those detailed in the “cs-unplugged” resource or using online interactive resources like those provided in the [“computer science field guide](http://csfieldguide.org.nz/en/chapters/algorithms.html#searching)”. Some preparation of resources may be required by the teacher if they wish to complete a comprehensive unplugged activity on searching. Activity description is found in the [Unplugged Searching Algorithms](http://csunplugged.org/wp-content/uploads/2014/12/unplugged-06-searching_algorithms.pdf) pdf. Multiple sets of these for a large class, or similar cards created by the teacher are also acceptable.
3. Similarly students will carry out a demonstration of a sorting algorithm selected from; selection sort, insertion sort, bubble sort, quick sort or merge sort. As before, students will need to show evidence that they have carried out this algorithm using their chosen method and they have provided sufficient evidence in their portfolio. Teachers may choose resources that they are comfortable with including the use of cards, straws (cut to different lengths) or the comprehensive [Unplugged Sorting Algorithms](http://csunplugged.org/wp-content/uploads/2014/12/unplugged-07-sorting_algorithms.pdf) activity outlined in the pdf resource. Alternatively students could show evidence of performing an algorithm with the interactive resources shown on the cs field guide.
4. It is important that not all values being sorted are consecutive. Sorting items where there are gaps (e.g. 20, 34, 1, 105) often reveal what is really happening a lot better than sorting consecutive items (e.g. 34, 35, 36 or Sally, Thomas, Ulanda).
5. Students should then discuss in writing how the cost of the chosen algorithm(s) changes as the size of the data set increases. They should also have had the opportunity to investigate and discuss the best, worst and average case scenarios for related algorithms and how different data sets may affect the results. This resource details how this could be done in the student instructions section.

***Further teacher guidance***

* The intention of this standard is that students should demonstrate their own understanding, and the expectation is that the understanding will be at the depth appropriate to a Level 1 student.
* Students are only expected to use maths appropriate to NCEA Level 1 (for example, logarithms are not covered in Level 1 maths).
* Some of the sorting algorithms are naturally expressed recursively. Students are not expected to be able to code up these algorithms themselves. They can, however, give clear, informal, descriptions of the algorithms. Students need to be clear about the details of the algorithm that they are using, as this may affect efficiency.
* Students can use downloadable programs e.g. the CS Field Guide, for large data sets.
* Ensure that students do not plagiarise material, particularly from Wikipedia and the CS Field Guide.
* “Determine”, at this level, does not require the derivation of a formula for time complexity (at any level).
* Students should make use of the data that they have generated themselves to determine the behaviour of the algorithm.

***Project based learning and collaboration***

Students are encouraged to work with a partner to help them record their demonstration of one of these algorithms in photos or videos that help students easily communicate their understanding of the algorithm and how it works. Students should annotate the photographs to help explain their processes. If online tools are used, annotated screenshots should be used for their report.

**Conditions/Ngā Tikanga**

Where a group approach is used, the teacher/kaiako needs to ensure that each student creates their own portfolio outlining their understanding of searching and sorting algorithms in their own words.

The 3 credits for the achievement standard indicates that approximately 30 hours needs to be allocated for teaching, learning (in and out of the classroom) and assessment in a programme of study. You may want to give students guidance on appropriate style and format for their portfolio. This achievement standard does not assess writing format or style.

While a written portfolio gives opportunity to assess individual student understanding after all the tasks are complete, records of classroom observation can contribute to a student demonstrating their searching and sorting algorithms as is required by the standard. Electronic documents can have short embedded videos for the same purpose if the teacher deems this appropriate. Annotated photographs are also useful in explanations.

Some of these tasks could be done as a class sharing the results, or by each individual. If you choose to do this as a class, divide the students into groups and give each group one data set size. Then have each group do their 10 experiments each with a different data set of that fixed size and share the results with the rest of the class. Individual work needs to be identifiable.

Conditions of Assessment related to this achievement standard can be found at <http://ncea.tki.org.nz/Resources-for-Internally-Assessed-Achievement-Standards>

**Resource requirements/Ngā Rauemi**

Students will need access to the web, digital devices and information from a variety of sources, such as: extracts, and/or notes from textbooks, the computer science field guide or “cs-unplugged”. Useful resources include:

* <http://csfieldguide.org.nz/en/chapters/algorithms.html>
* <http://csunplugged.org>
* Unplugged Searching Algorithms activity: <http://csunplugged.org/wp-content/uploads/2014/12/unplugged-06-searching_algorithms.pdf>
* Unplugged Sorting Algorithm activity: <http://csunplugged.org/wp-content/uploads/2014/12/unplugged-07-sorting_algorithms.pdf>
* CS education research group. (2017). Computer Science Unplugged. Retrieved 21 May, 2017, from <http://csunplugged.org/>

YouTube videos for creating your own sort programs. Note: these will need to be modified to count the number of comparisons to compute the cost:

* Linear and Binary search: <https://www.youtube.com/watch?v=Y4h9w1FGmsY>
* Bubble sort: <https://www.youtube.com/watch?v=YHm_4bVOe1s&index=3&list=PLj8W7XIvO93rJHSYzkk7CgfiLQRUEC2Sq>

Note: Students who use this algorithm should be careful \*not\* to copy the table in Wikipedia describing the complexity of Bubble sort.

* Insertion sort: <https://www.youtube.com/watch?v=Nkw6Jg_Gi4w&list=PLj8W7XIvO93rJHSYzkk7CgfiLQRUEC2Sq>
* Quick sort: <https://www.youtube.com/watch?v=CB_NCoxzQnk>
* Merge sort: <https://www.youtube.com/watch?v=Nso25TkBsYI>

**Scratch programs:**

* Linear and binary search: <https://www.youtube.com/watch?v=yyZ84AiQ4uc&t=50s>
* Bubble sort (& random number generator): <https://www.youtube.com/watch?v=S_eOTuRICQk>
* Merge, Insertion and Bubble sort: <https://www.youtube.com/watch?v=WLsMFZBsC6o&list=PL99XwQ3slYF1Azru0Ppe6bDvC59-3GoQs>

**Additional information/He Kōrero Atu**

Teachers could use this information to support student learning:

**Familiarisation**

* Algorithms are constantly used by computer scientists when solving real world problems. What common search algorithms are out there? This may be easiest to do as a table with name and a brief description of how it works.
* What common sorting algorithms are out there?
* In the real world where are searching and sorting algorithms actually used? You must have more than one real world example. (It will be very useful to show details of how much data needs to be searched/sorted for results in these real-world examples. Actual stats will assist you later)
* Why is it important that an efficient searching or sorting algorithm is chosen for each of these?

**Internal Assessment Resource**

**Achievement standard:** 91885

**Standard title:**  Demonstrate understanding of searching and sorting algorithms

**Credits:** 3

**Resource title:** Let’s Investigate!

**Resource reference:** Digital Technologies & Hangarau Matihiko 1.9B Version 1

**Student/Ākonga instructions**

**Introduction/Kupu Arataki**

This assessment activity requires you to complete a number of tasks about searching and sorting algorithms and identify where these are used in the real world.

You are going to be assessed on how comprehensively you demonstrate your understanding of searching and sorting algorithms.

You need to collect evidence showing your work, as well as an evaluation at the end, where you can demonstrate what you have learned about this topic.

You may work with others while investigating the topic; however, you must each submit your own evidence in your own words.

Teacher note: Insert due dates and timeframes

**Task/Hei Mahi**

You may want to work with a partner in order to capture photographic or video evidence of your demonstration. Use the resources your teacher will have chosen for you.

You will each need to provide evidence of your own demonstration and should annotate thepictures, screenshots, sketches, or videos describing the steps you took.

**Evidence and Resources:**

Your teacher will inform you on where to capture the evidence of your tasks - a Word document, Google docs, OneNote class notebook, or visual diary.

Your teacher will also supply the data sets you will use for your searching/sorting tasks.

**Activity**

Follow the instructions below and complete ALL tasks before attempting the final evaluation task.

**Task 1 - Searching algorithms**

* Choose a search algorithm as a class, either Linear search or Binary search, and describe in your own words how the algorithm works to find where a given piece of data is stored.
* Work in pairs to carry out the chosen algorithm using your own sets of data items. These could include finding a random word in a dictionary, finding a particular card in a pile of cards or any other activity that your teacher chooses. (You may use diagrams, annotated photos or a short video as evidence.) You must then add a description of the algorithm to your own evidence including you demonstrating the algorithm.
* What did this algorithm cost to complete? (Costs are often best measured in number of comparisons required to complete the algorithm successfully. This is easier than attempting to estimate how much time is taken to complete the algorithm etc.)
* Discuss results as a class. Create a table of results on the board. Did everyone get the same results? Why/why not? This will help you to determine the best, average and worst-case costs of the algorithm (a spreadsheet will be useful here). Include a copy of this in your evidence.
* Explain in your own words why the best, worst and average measures are the same/different for this algorithm.
* How will this cost change if you double the data set size? Try it and see if you are right. Remember to note the new cost in your evidence.

**Task 2 - Sorting Algorithms**

* As for task 1, but using sorting algorithms instead. Choose from Bubble sort, Quick sort, Insertion sort and Merge sort. Your teacher will guide you with some experiments you can do using cards, weights or straws. Alternatively, there are several excellent experiments you can do online at <http://csfieldguide.org.nz/en/chapters/algorithms.html#sorting> (Note: You only need to describe **one** sorting algorithm in detail but it is good to understand more than one algorithm).
* What did this algorithm cost to complete? (Costs are often best measured in number of comparisons required to complete the algorithm successfully. This is easier than attempting to estimate the time taken to complete the algorithm etc.)
* Discuss results as a class. Create a table of results on the board. Did everyone get the same results? Why/why not? Put a copy of this in your evidence. This will help you to determine the best, average and worst-case costs of the algorithm (a spreadsheet will be useful here).
* Explain in your own words why the best, worst and average measures are the same/different for this algorithm.
* How will this cost change if you double the data set size? Try it and see if you are right. Remember to note the new cost in your evidence.

**You now need to do task 3 or 4 (but not both):**

**Task 3 - Searching algorithms: Investigating changing costs as data set size changes**

To do this task you will need a random number generator that creates lists of numbers between 1 and 1000 in set sizes of your choice. You will also need some computer programs that search through these lists using linear search and binary search so that you can compare the results. These may be supplied by your teacher.

Make sure that the program calculates the number of comparisons it does when searching. You will also need a program that sorts the data sets so that the binary search will work. It is recommended that you sort the data before running either algorithm to make it a fair test. You should use the same data set with each algorithm each time.

To investigate how the cost change compares between two different algorithms, we will run the following experiment.

* You need to choose manageable data set sizes with regular intervals between them, for example 100, 200, 300, 400, 500. Investigate how this cost changes as the size of the data set increases. Run 10 experiments with each data size so that you can calculate the best, average and worst-case costs experimentally for each.
* Create a table of your results showing each data set size for each algorithm.
* In each case, determine the best, average and worst-case costs of each data set. If you are using a spreadsheet, the min, average and max functions can help you to do this.
* Graph the average results of all the data sets for the two different algorithms on the same line graph, being careful to label the axes correctly.
* Explain what happens to the costs of each algorithm as the data set size increases.
* Now that you can compare the two algorithms, which would you choose to use for large data sets and why?
* Explain the implications of the use of this algorithm in your real-world applications. For example, which one would produce the search result in a reasonable amount of time on very large data sets? What would be the impact on a real-world application of using a less efficient algorithm? Why do you think this is so?

**Task 4 - Sorting algorithms: Investigating changing costs as data set size changes**

To do this task you will need a random number generator that creates lists of numbers between 1 and 1000 in set sizes of your choice. You will also need some computer programs that sort through these lists using two of the following sorts of your choice: bubble sort, insertion sort, quick sort and merge sort. This will allow you to calculate the best, average and worst-case costs experimentally and compare the results. Make sure that the program calculates the number of comparisons it does when sorting.

To investigate how the cost change compares between two different algorithms, we will run the following experiment. To make it a fair test, you need to use the same data set with each algorithm each time.

* You need to choose manageable data set sizes with regular intervals between them, for example 100, 200, 300, 400, 500. Investigate how the cost changes as the size of the data set increases. Run 10 experiments with each data size so that you can calculate the best, average and worst-case costs experimentally for each data set size.
* Create a table of your results showing each data set size for each algorithm.
* In each case, determine the best, average and worst-case costs of each data set. If you are using a spreadsheet, then the min, average and max functions can help you to do this.
* Graph the average results of all the data sets for the two different algorithms on the same line graph, being careful to label the axes correctly.
* Explain what happens to the costs of each algorithm as the data set size increases.
* Now that you can compare the two algorithms, which would you choose to use for large data sets and why?
* Explain the implications of the use of this algorithm in your real-world applications. For example, which one would sort the data in a reasonable amount of time on very large data sets? What would be the impact on a real-world application of using a less efficient algorithm? Why do you think this is so?

**Task 5 - Final Evaluation**

* Consider the best, average and worst-case costs of the two related searching algorithms you used and explain the implications of these in the real world. When would you choose to use them and why? **OR** Consider the best, average and worst-case costs of the two related sorting algorithms you used and explain the implications of these in the real world. When would you choose to use them and why?

**Assessment schedule/Mahere Aromatawai: Digital Technologies & Hangarau Matihiko 91885 – Let’s investigate!**

|  |  |  |
| --- | --- | --- |
| **Evidence/Judgements for Achievement/Paetae** | **Evidence/Judgements for Achievement with Merit/Kaiaka** | **Evidence/Judgements for Achievement with Excellence/Kairangi** |
| Demonstrate understanding of searching and sorting algorithms.  The student:   * describes applications of searching and sorting algorithms   **For example (partial evidence):**  *Searching and sorting are used in a library to help us find the books that we want. The books are stored on the shelves in a sorted order which makes searching for the book you want a lot easier….the books are organised in a way that makes it easier for us to find them….*  *All of the names in a phone book are sorted so they appear in alphabetical order...this makes it much easier for us to find the name that we need….*   * carries out a searching algorithm   *There is evidence of an understanding of the steps required to undertake one of the searching algorithms like a linear or binary search. This evidence can be supported by teacher observation if the written, annotated or video evidence within the report is incomplete or unclear.*   * carries out a sorting algorithm   *There is evidence of understanding of the steps required to perform a sorting algorithm, for example the selection sort or quick sort. This evidence can be supported by teacher observation if the written, annotated or video evidence is incomplete or unclear.*   * describes how the cost for a chosen searching or sorting algorithm changes as the size of the problem increases   **For example (partial evidence):**  *“When we were given the bigger list of words to look through it would have taken us twice as long to check each word to find the one we needed because there were twice as many words…....*  *using binary search if the list was twice as long you would only need to do one more search….*  *but what if the list was twice as long?….. you would have to…….*  *...but with a pile of 20 cards we would have to go through the algorithm…….”*  *The examples above are indicative samples only* | Demonstrate in-depth understanding of searching and sorting algorithms.  The student:   * explains the relationship between searching and sorting algorithms   **For example (partial evidence):**  *“When we were given an unsorted list we could not easily find the word we were looking for because…….*  *….the data has to be sorted before the algorithm works…..*  *...if they were not sorted before we tried to find something…..this is because*  *...you can only use binary search on data that has been sorted because…”*   * determines the best, average and worst case costs of two related searching or sorting algorithms and explains the implications   **For example (partial evidence):**  *“When we used the linear search the worst case would be that we might have to look through all the data in order to…..this is not the case with the binary search because….*  *the best case would be that we find the ball under the cup on the first try…...but on average……..this means that…..*  *if you weigh the heaviest container first……*  *….the best case would be that…..this means...*  *....“the average” case would be that I would need 6 tries…..but if you use binary search…..*  *if the boxes were already in order….this would be the best case….but on average…..”*  *since the number of comparisons for …. algorithm increases so much faster than it does for the …. algorithm, I would choose …. algorithm to sort lots of data or it could take years to do. However, if there are only 10 or less items to sort, it doesn’t make much difference which algorithm you use.*  *The examples above are indicative samples only* | Demonstrate comprehensive understanding of searching and sorting algorithms.  The student:   * discusses real-world usage and implications of searching and sorting algorithms   **For example (partial evidence):**  *...with 3.5 billion searches a day on average...imagine how long it would take…..*  *...Amazon’s 5 million digital documents...if the data wasn’t sorted….*  *...quicksort is used most commonly even on large data sets like……..*  *... medical and scientific advances, such as searching for DNA sequences, have been made possible by….*   * investigates the cost of searching and sorting algorithms with different data sets   **For example (partial evidence):**  *….in our experiment it took 6 more iterations because…*  *...when the data was reversed...the algorithm still took……*  *...when there were 10 times as many words the linear search would take….. while the binary search would only take...*  *...we see a more linear increase in the cost...when compared to...the difference is much more notable as the size of the sample increases….*    *The examples above are indicative samples only* |

Final grades will be decided using professional judgement based on a holistic examination of the evidence provided against the criteria in the Achievement Standard

1. This achievement standard is derived from both *The New Zealand Curriculum* and *Te* *Marautanga o Aotearoa.* [↑](#footnote-ref-1)