

AS91074 Searching

This is a guide for students attempting the Algorithms topic of digital technologies achievement standard 1.44 (AS91074). If you follow this guide, then you do not need to follow the sorting algorithms one.

In order to fully cover the standard, you will also need to have done projects covering the topics of Programming Languages and Human Computer Interaction in the standard, and included these in your report.

Overview

The topic of Algorithms has the following bullet points in achievement standard 1.44, which this guide covers. This guide separates them into two categories.

Comparing algorithms, programs, and informal instructions

Achieved: "describing the key characteristics, and roles of algorithms, programs and informal instructions"

Merit: "explaining how algorithms are distinct from related concepts such as programs and informal instructions"

Excellence: "comparing and contrasting the concepts of algorithms, programs, and informal instructions"

Determining the cost of algorithms and understanding various kinds of steps in algorithms

Achieved: "describing an algorithm for a task, showing understanding of the kinds of steps that can be in an algorithm, and determining the cost of an algorithm for a problem of a particular size"

Merit: "showing understanding of the way steps in an algorithm for a task can be combined in sequential, conditional, and iterative structures and determining the cost of an iterative algorithm for a problem of size n"

Excellence: "determining and comparing the costs of two different iterative algorithms for the same problem of size n"

As with all externally assessed Digital Technology reports, you should base your explanations around personalised examples.

Reading from the Computer Science Field Guide

You should read and work through the interactives in the following sections of the CS Field Guide in order to prepare yourself for the assessed project.

[What's the big picture?](#)

[Searching Algorithms](#)

Project

This project involves understanding linear search and binary search.

Writing your report for the main bullet points that cover algorithms

Achieved

Ensure you have tried both of the box searching interactives which are in the part of the field guide which you read. For one of them you had to use linear search, and for the other you had to use binary search.

Take a screenshot of a completed search using the binary search interactive (if you get lucky and find the target within 2 clicks, keep restarting until it takes at least 3, so that you something sufficient to show in your report). Show on your screenshot which boxes you opened, and put how many boxes you opened. The number of boxes you opened is the cost of the algorithm for this particular problem. Include your screenshot in your report.

Describe (in your own words with 1 - 3 sentences) the overall process you carried out to search through the boxes. Try and make your explanation general, e.g. if you gave the instructions to somebody who needs to know how to search 100 boxes, or 500 boxes, the instructions would be meaningful.

You also need to show the kinds of steps that can be in an algorithm, such as iterative, conditional, and sequential. If you don't know what these terms mean, go have another look at the field guide. Get a Scratch program (or another language if you are fairly confident with understanding the language) that implements binary search. Take a screenshot of it, or a large part of it (you want to ensure that the screenshot takes up no more than half a page in the report, but is still readable) and open it in a drawing program such as paint. Add arrows and notes showing a part of the algorithm that is sequential, part that is conditional, and part that is iterative.

Merit/Excellence

It should be obvious from your initial investigation that binary search is far better than linear search! Although you still might say, why not just use a faster computer? To explore this possibility, you are now going to analyse what happens with a huge amount of data. More specifically, you are going to answer the following question: "How do linear search and binary search compare when the amount of data to search is doubled?"

Start by picking a really large number (e.g. in the billions, or even bigger - this is the amount of data that large online companies such as Google or Facebook have to search). Imagine you have this number of boxes that you have to search. Also, imagine that you then have two times that number of boxes, four times that number of boxes, eight times that number of boxes, and sixteen times that number of boxes.

Now, using those 5 different amounts of boxes, you are going to determine how many boxes would have to be looked at on average to find a target for linear search and binary search. You will then also calculate the amount of time you could expect it to take, using the average number of boxes to be looked at and an estimate of how many boxes a really fast computer could check per second. As you do the various calculations, you should add them into a table, such as the one below. This will be a part of your report.

Values for n	Average for Linear Search	Average for Binary Search	Expected Time for Linear Search	Expected Time for Binary Search
Chosen number	???	???	???	???
Chosen number x 2	???	???	???	???
Chosen number x 4	???	???	???	???
Chosen number x 8	???	???	???	???
Chosen number x 16	???	???	???	???

Rather than actually carrying out the searching (the interactive is not big enough!), you are going to calculate the expected averages. Computer scientists call this analysing an algorithm, and often it is better to work out how long an algorithm can be expected to take before waiting years for it to run and wondering if it will ever complete. Remember that you can use the [big number calculator](#) and the [time calculator](#) in the field guide to help you with the math. If you are really keen, you could make a spreadsheet to do the calculations and graph trends.

Hint for estimating linear search: Remember that in the worst case, you would have to look at every box (if the target turned out to be the last one), and on average you'll have to check half of them. Therefore, to calculate the average number of boxes that linear search would have to look at, just halve the total number of boxes.

Hint for estimating binary search: Remember that with each box you look at, you are able to throw away half (give or take 1) of the boxes. Therefore, to calculate the average number of boxes that binary search would have to look at, repeatedly divide the number by 2 until it gets down to 1. However many times you divide by 2 is the average cost for binary search. Don't worry if your answer isn't perfect; it's okay to be within 3 or so of the correct answer. This means that if while halving your number it never gets down to exactly 1 (e.g. it gets down to 1.43 and then 0.715), your answer will be near enough. As long as you have halved your number repeatedly until it gets down to a number that is less than 1, your answer will be accurate.

Now that you have calculated the average number of boxes for each algorithm, you can calculate how long it would take on a high end computer for each algorithm with each problem size. Assume that the computer can look at 1 billion boxes per second. Don't worry about being too accurate (e.g. just round to the nearest millisecond (1/1000 of a second), second, minute, hour, day, month, or year). Some of the values will be a tiny fraction of a millisecond. For those,

just write something like "Less than 1 millisecond". You can get the number by dividing the expected number of boxes to check by 1 billion.

You should notice some obvious trends in your table. Explain these trends, and in particular explain how the amount of time each algorithm takes changes as the problem size doubles.

Does it have a significant impact on the amount of time the algorithm will take to run?

Remember the original question you were asked to investigate: "How do linear search and binary search compare when the amount of data to search is doubled?".

Using your findings to guide you, discuss one of the following scenarios:

- Imagine that you are a data analyst with the task of searching these boxes, and in order to do your work you need to search for many pieces of data each day. What would happen if you were trying to use linear search instead of binary search?
- Imagine that you have a web server that has to search a large amount of data and then return a response to a user in a web browser (for example searching for a person on Facebook). A general rule of computer systems is that if they take longer than 1/10 of a second (100 milliseconds) to return an answer, the delay will be noticeable to a human. How do binary search and linear search compare when it comes to ensuring there is not a noticeable delay?

Writing the part of your report for the other algorithms bullet points

Achieved/Merit/Excellence

We recommend doing this part after you have done programming languages.

All three levels (A/M/E) are cleanly subsumed by the E requirement, so you should try to do that i.e. "comparing and contrasting the concepts of algorithms, programs, and informal instructions".

You should refer to examples you used in your report or include additional examples (e.g. a program used as an example in the programming languages topic, or an algorithm describing the searching process, etc). If you are confused, have another look at the field guide. You should only need to write a few sentences to address this requirement.

Hints for success

- Don't confuse "algorithm cost" with the "algorithm length". The number of lines in the algorithm or program normally unrelated to the cost. Cost is the time the algorithm actually takes to run, or the number of comparisons that have to be made. You can find more information in the Field Guide if you are not sure.
- Resize screenshots/photos so that they are large enough to see what is on them, but not taking up unnecessary space.

Recommended Number of Pages

Within the 3 to 4 pages recommended for algorithms, a possible breakdown is:

- ½ page: Screenshot of you carrying out binary search with the interactive. (Achieved)
- ¼ page: Explanation of your binary search screenshot. (Achieved)

- $\frac{1}{4}$ page: General instructions for carrying out binary search. (Achieved)
- $\frac{1}{2}$ page: Your example of the iterative, conditional, and sequential steps that can be in an algorithm (Merit)
- $1\frac{1}{2}$ pages: Your investigation and data collected for merit/ excellence. Including results and discussion (Merit/ Excellence)
- $\frac{1}{4}$ page to $\frac{1}{2}$ page: Explanation of the difference between algorithms, programs, and informal instructions (Achieved/ Merit/ Excellence)

These are maximums, not targets!

For the topic of searching algorithms you probably won't need this much space (sorting algorithms tends to require more space).

Note that if you go over 4 pages for Algorithms, then you may have to use fewer pages for one of the other two topics, which could be problematic. No other material should be included for Algorithms.