

3.1.4 Sorting algorithms 2

Lesson plan and printable activities

Teacher notes

Excellent background information on algorithms here:

[khanacademy.org/computing/computer-science/algorithms](https://www.khanacademy.org/computing/computer-science/algorithms)

Materials needed

1. 3.1.4 [Lesson 2](#) PowerPoint.
2. [Quiz 1](#): Example of a merge sort.
3. [Quiz 2](#): Comparing the bubble sort and merge sort.

Lesson aims

1. To get students to think about how different approaches may be taken when sorting data into order.
2. To see that different methods of sorting vary in terms of complexity and efficiency.

Lesson objectives

1. Understand and explain how the merge sort algorithm works.
2. Compare and contrast merge sort and bubble sort algorithms.

Starter activity (~5 minutes)

1. Use this opportunity to emphasise that this is the final of what has effectively been a set of four related lessons.
2. Ask students to recall the key features of the earlier lessons on searching and to think about the preceding lesson on performing a bubble sort.
3. **Slide 3:** Explain that we are now going to look at a more complex mechanism of sorting data – the **merge sort**. Show YouTube video.

Main activities (35 minutes)

1. **Slide 4:** Explain the meaning of 'divide and conquer' algorithms.
2. **Slide 5:** Show YouTube video. This is an animation of the merge sort.
3. **Slides 6–7:** This first part of the lesson deals with the mechanics of performing a merge sort. It is strongly suggested that students are provided with a pen and paper to manually draw out the merge sort example in the presentation individually – allow 10 minutes for this activity. Following on from this, they can complete their own merge sort using [Quiz 1](#). This will take a further 10 minutes.
4. **Slides 8–9:** Recap the key features of the merge sort with reference to the previous example.

Plenary activity (10 minutes)

1. **Slides 10 – 12:** Comparing the key features, advantages and disadvantages, of bubble sort and merge sort. In the final part of the lesson, take the opportunity to consolidate students' learning about the efficiencies of operation in different searching and sorting algorithms. Emphasise again that a simpler underlying algorithm often leads to a less efficient process for searching *or* sorting.
2. Complete [Quiz 2](#).

Lesson

3.3 Fundamentals of data representation

3.1.4 Sorting algorithms 2

Lesson

© 2016 AQA. Created by Teachit for AQA

Objectives

Understand and explain how the merge sort algorithm works.

Compare and contrast merge sort and bubble sort algorithms.

© 2016 AQA. Created by Teachit for AQA

Introduction to the merge sort

In the previous lesson, we looked at the idea of applying the bubble sort algorithm when ordering datasets.

We concluded that the bubble sort was a simple but inefficient algorithm. Now we are going to look at a related but more sophisticated algorithm – the ‘merge sort’.

[youtube.com/watch?v=XagR3G_NVoo&nohtml5=False](https://www.youtube.com/watch?v=XagR3G_NVoo&nohtml5=False)

© 2016 AQA. Created by Teachit for AQA

How does a merge sort work?

A merge sort is a method of arranging data by splitting a list into discrete elements and merging the elements together until an ordered list is obtained.

Merge sorts are classed as as ‘divide-and-conquer’ algorithms.

- **Divide** (the problem into a small number of pieces),
- **Conquer** (solve each piece, by repeatedly applying divide-and-conquer to it),
- **Combine** (the pieces together into a solution).

© 2016 AQA. Created by Teachit for AQA

Visual demonstration of a merge sort

[youtube.com/watch?v=HsZ-YRQM8sE&nohtml5=False](https://www.youtube.com/watch?v=HsZ-YRQM8sE&nohtml5=False)

© 2016 AQA. Created by Teachit for AQA

Merge sorts

We will stick to examples where an **even** number of elements are sorted.

The merge sort starts off by repeatedly splitting the array in half until there are only individual elements left.

Array

5	8	4	2	19	34	56	7
---	---	---	---	----	----	----	---

Split

5	8	4	2	19	34	56	7
---	---	---	---	----	----	----	---

Split

5	8	4	2	19	34	56	7
---	---	---	---	----	----	----	---

Split

5	8	4	2	5	8	4	2
---	---	---	---	---	---	---	---

© 2016 AQA. Created by Teachit for AQA

3.1 Fundamentals of algorithms

Merge sorts

It is helpful to view the elements vertically for the next step. Merging occurs by making a 'bottom up' comparison and swapping values.

© 2016 AQA. Created by Teachit for AQA

Merge sorts

Our example is a simple one – the array contained just eight elements and is easily split into pairs. The full algorithm must cope with the possibility of odd numbers of array elements, of much larger arrays and numbers which do not easily split into pairs.

As with the searching algorithms, a more complex process requires fewer steps to complete than the simpler but more basic alternatives.

The binary search involves bisecting a dataset until a match is made and a similar process occurs here in the merge sort.

© 2016 AQA. Created by Teachit for AQA

Merge sorts

We talk about the **recursive** operation of components of the underlying algorithm, i.e. it repeats itself where required.

Recursive programming can consume a lot of memory and that is the price paid for having a more efficient algorithm.

© 2016 AQA. Created by Teachit for AQA

Comparing sorting algorithms

A quick review of the two sorting algorithms that we have seen in use so far.

- Bubble
- Merge

What comparisons might be made of these two techniques?

© 2016 AQA. Created by Teachit for AQA

Comparing sorting algorithms

Sorting algorithms	Bubble	Advantages	
		Disadvantages	
	Merge	Advantages	
		Disadvantages	

© 2016 AQA. Created by Teachit for AQA

To round things off...

That brings us to the end of two separate topics –

- Searching
- Sorting

Both of which have exhibited similar characteristics in that a clear trade-off between the efficiency and complexity of the underlying processes can be demonstrated.

Now complete **Quiz 1** in order to round off this final lesson.

© 2016 AQA. Created by Teachit for AQA

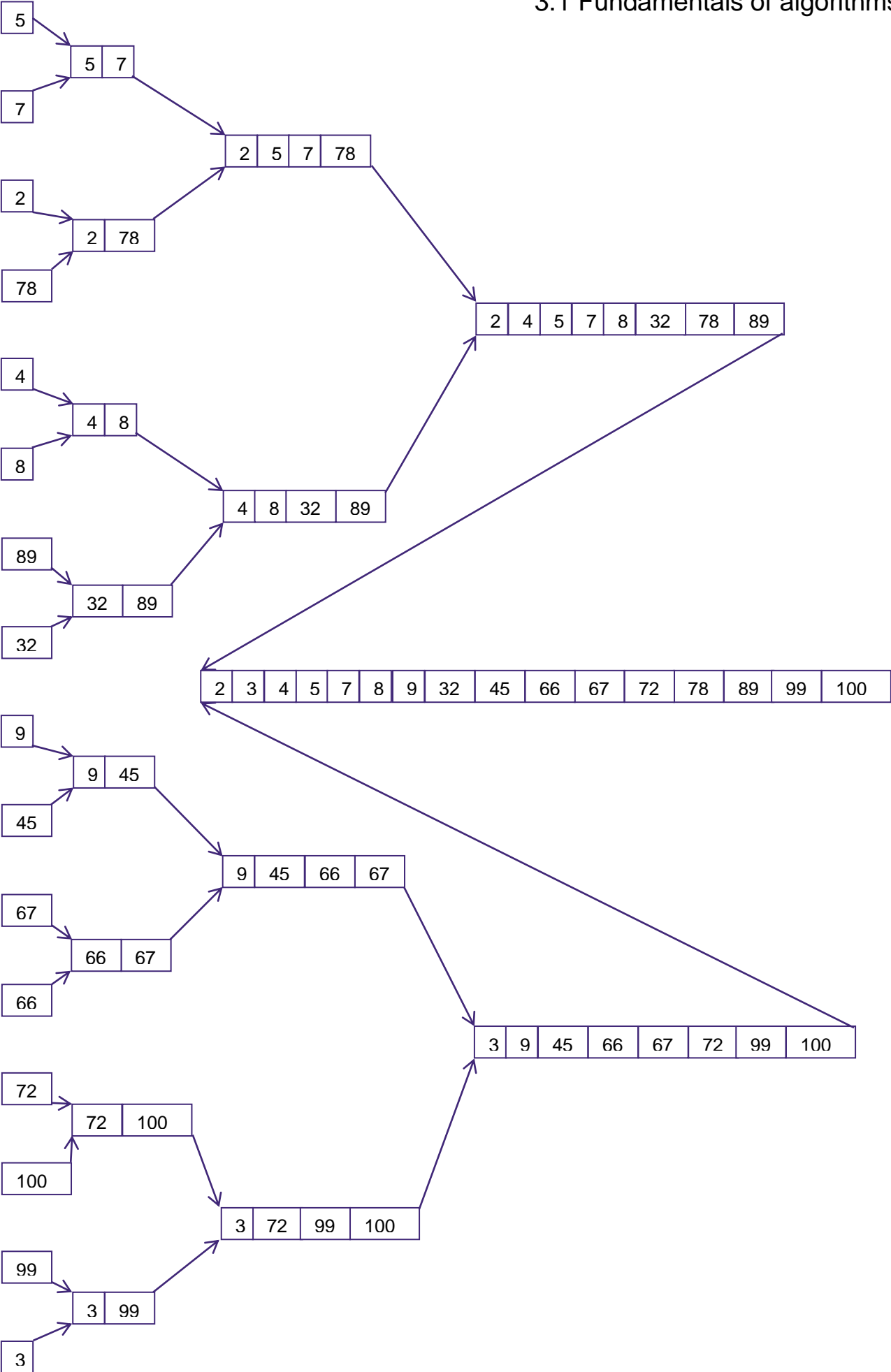
Quiz 1 – Merge sort – answers

Using an appropriate diagrammatic representation, show how a merge sort will proceed for the following unsorted array of numbers.

5	7	2	78	4	8	89	32	9	45	67	66	72	100	99	3
---	---	---	----	---	---	----	----	---	----	----	----	----	-----	----	---

5	7	2	78	4	8	89	32	9	45	67	66	72	100	99	3
5	7	2	78	4	8	89	32	9	45	67	66	72	100	99	3
5	7	2	78	4	8	89	32	9	45	67	66	72	100	99	3
5	7	2	78	4	8	89	32	9	45	67	66	72	100	99	3

3.1 Fundamentals of algorithms



Quiz 2 – Comparing sorting techniques

Question	Answer
Write a short explanation outlining the key differences between carrying out a bubble sort and a merge sort.	
Give reasons why one technique might be used in preference to the other.	

Quiz 2 – Comparing sorting techniques – answers

Question	Answer
Write a short explanation outlining the key differences between carrying out a bubble sort and a merge sort.	<p>Bubble/Merge input is an array (dataset) of unsorted data.</p> <p>Output is data sorted by numerical/alphabetical order as required.</p> <p>Bubble sort repeatedly steps through the array.</p> <p>Comparison of adjacent data pairs and swapping if required.</p> <p>Merge sort splits data array into individual elements then compares values and merges elements back together again (until finally array reassembled).</p>
Give reasons why one technique might be used in preference to the other.	<p>Bubble sort is a simpler algorithm to construct, but far less efficient/much slower than merge sort.</p>