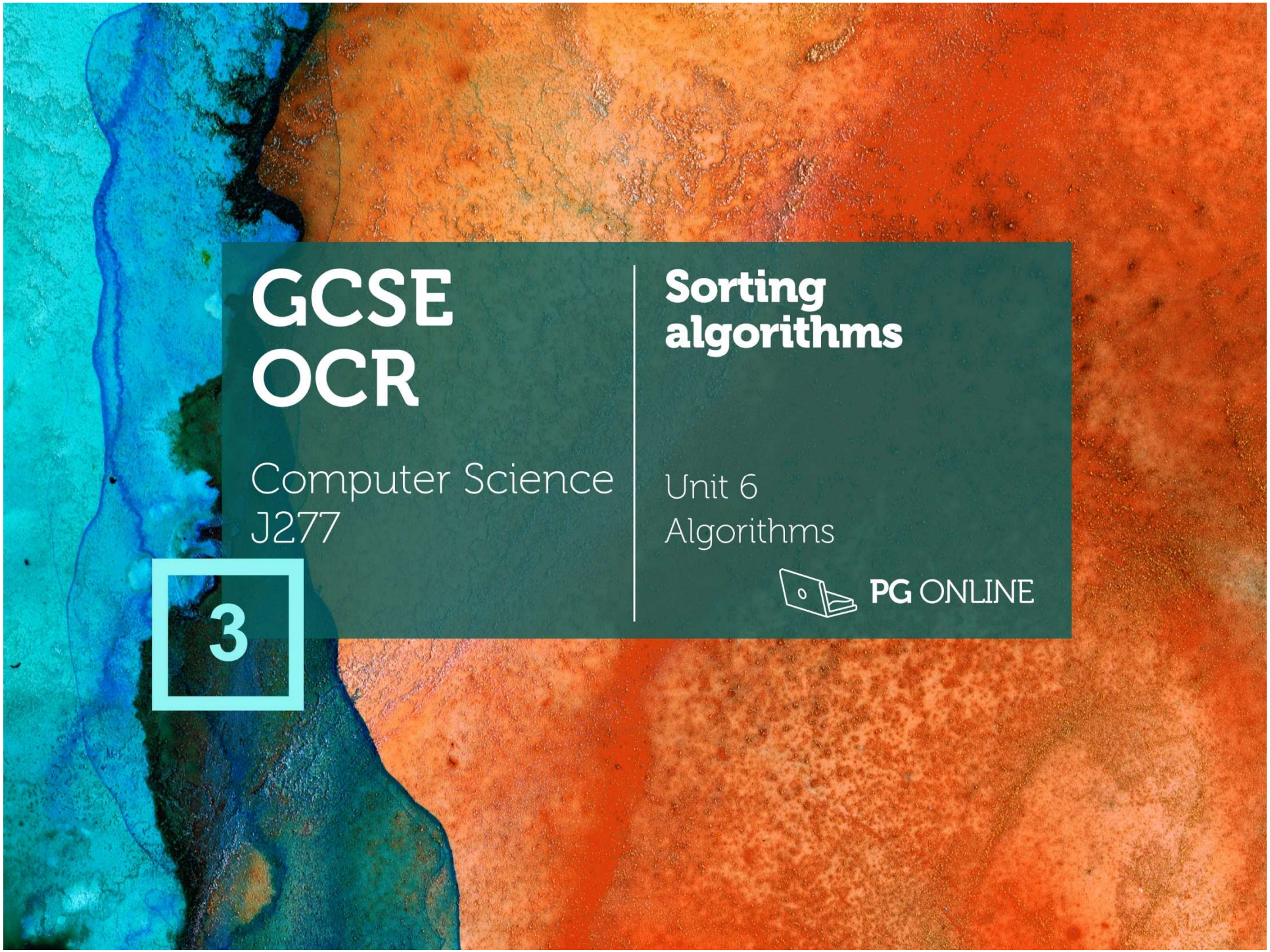
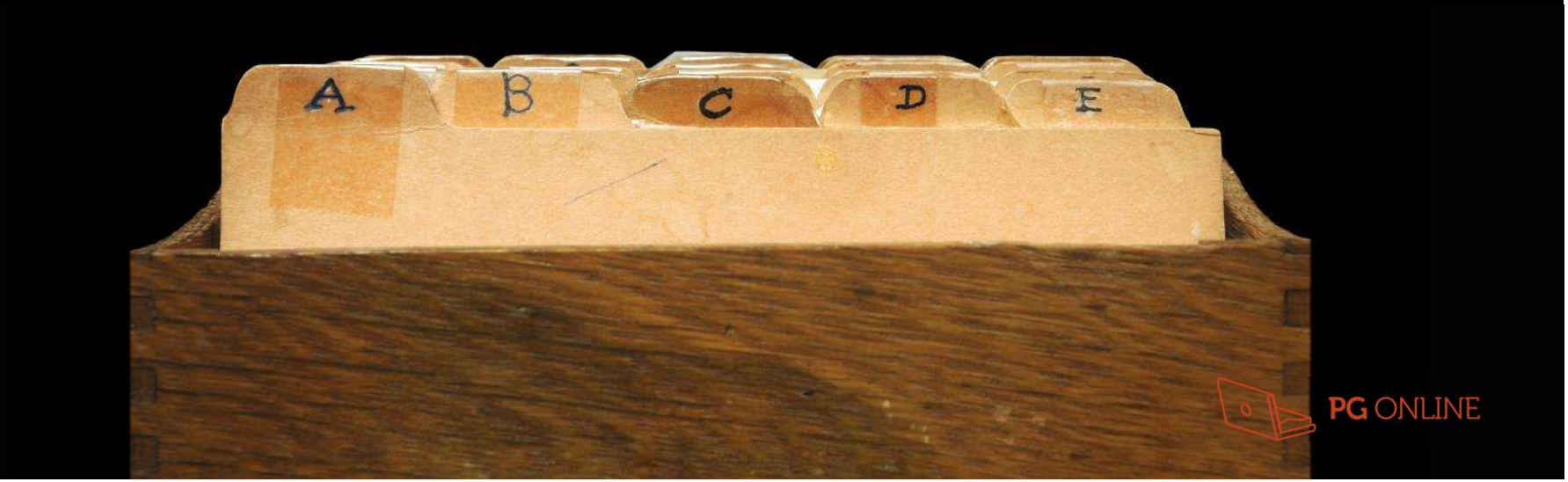

GCSE
OCR
Computer Science
J277
3
Sorting algorithms
Unit 6
Algorithms
PG ONLINE

Objectives

- Understand the standard sort algorithms:
  - Bubble sort
  - Insertion sort
  - Merge sort
- Be able to apply each algorithm to a data set
- Be able to Identify an algorithm if given the code for it

Starter

- Both people and computers often need data to be sorted
  - Give **five** types of data that people need to be sorted
  - Give **one** reason why a computer would work more efficiently if a list was sorted



Starter

Answers

- Types of data that people need sorted
  - Index cards/records of customer details
  - Directories and dictionaries
  - House numbers
  - Library books
  - Stock in a warehouse
  - and many more…
- Reasons why a computer will work more efficiently with sorted lists
  - Computers can use a binary search with sorted lists which is far more efficient than a linear search

Sorting

- Data sets frequently need to be sorted
  - Look at the two data sets of the top 10 boys and girls names in 2019
  - What other orders could they be sorted in?
  - Why might you need to sort the data sets?

Rank	Name	Count
1	Oliver	5 390
2	George	4 960
3	Harry	4 512
4	Noah	4 107
5	Jack	3 988
6	Leo	3 721
7	Arthur	3 644
8	Muhammad	3 507
9	Oscar	3 459
10	Charlie	3 365

Rank	Name	Count
1	Olivia	4 598
2	Amelia	3 941
3	Ava	3 110
4	Isla	3 046
5	Emily	2 676
6	Mia	2 490
7	Isabella	2 369
8	Sophia	2 344
9	Ella	2 326
10	Grace	2 301

Source: ONS (2019)

Sorting

Answers

- The data sets could be sorted as follows:
  - Count order (which will be the same as the rank order)
  - Alphabetical order
  - Reverse alphabetical or count order
- The girls and boys data sets could be combined to show the top 20 names in England and Wales
  - Again these could be sorted by count or name

The bubble sort

- Start with the leftmost item
  - Compare this item with the one next to it
  - If the one next to it is less, swap the items
  - Repeat for all the other items
  - At the end of one pass through the list, the largest item is at the end of the list
- Repeat the process until the items are sorted
- Suppose you have a list of numbers to be sorted:

9	5	4	15	3	8	11	2

Bubble sort – First pass

9	5	4	15	3	8	11	2
5	9	4	15	3	8	11	2
5	4	9	15	3	8	11	2
5	4	9	15	3	8	11	2
5	4	9	3	15	8	11	2
5	4	9	3	8	15	11	2
5	4	9	3	8	11	15	2
5	4	9	3	8	11	2	15

- Each item is compared with the one on its right, and swapped if it is larger
- At the end of the first pass the largest item bubbles through to the end of the list
- (Mauve indicates sorted items)

First pass

9	5	4	15	3	8	11	2
5	9	4	15	3	8	11	2
5	4	9	15	3	8	11	2
5	4	9	15	3	8	11	2
5	4	9	3	15	8	11	2
5	4	9	3	8	15	11	2
5	4	9	3	8	11	15	2
5	4	9	3	8	11	2	15

Second pass

5	4	9	3	8	11	2	15
4	5	9	3	8	11	2	15
4	5	9	3	8	11	2	15
4	5	3	9	8	11	2	15
4	5	3	8	9	11	2	15
4	5	3	8	9	11	2	15
4	5	3	8	9	2	11	15

Second pass

5	4	9	3	8	11	2	15
4	5	9	3	8	11	2	15
4	5	9	3	8	11	2	15
4	5	3	9	8	11	2	15
4	5	3	8	9	11	2	15
4	5	3	8	9	11	2	15
4	5	3	8	9	2	11	15

Third pass

4	5	3	8	9	2	11	15
4	5	3	8	9	2	11	15
4	3	5	8	9	2	11	15
4	3	5	8	9	2	11	15
4	3	5	8	9	2	11	15
4	3	5	8	2	9	11	15

Third pass

4	5	3	8	9	2	11	15
4	5	3	8	9	2	11	15
4	3	5	8	9	2	11	15
4	3	5	8	9	2	11	15
4	3	5	8	9	2	11	15
4	3	5	8	2	9	11	15

Fourth pass

3	4	5	8	2	9	11	15
3	4	5	8	2	9	11	15
3	4	5	8	2	9	11	15
3	4	5	8	2	9	11	15
3	4	5	2	8	9	11	15

Fourth pass

3	4	5	8	2	9	11	15
3	4	5	8	2	9	11	15
3	4	5	8	2	9	11	15
3	4	5	8	2	9	11	15
3	4	5	2	8	9	11	15

Fifth pass

3	4	5	2	8	9	11	15
3	4	5	2	8	9	11	15
3	4	5	2	8	9	11	15
3	4	2	5	8	9	11	15

Fifth pass

3	4	5	2	8	9	11	15
3	4	5	2	8	9	11	15
3	4	5	2	8	9	11	15
3	4	2	5	8	9	11	15

Sixth pass

3	4	2	5	8	9	11	15
3	4	2	5	8	9	11	15
3	2	4	5	8	9	11	15

Sixth pass

3	4	2	5	8	9	11	15
3	4	2	5	8	9	11	15
3	2	4	5	8	9	11	15

Seventh pass

3	2	4	5	8	9	11	15
2	3	4	5	8	9	11	15

Seven passes through the list of eight numbers ensures that they are sorted

Worksheet 3

- Now complete **Task 1** on **Worksheet 3**

Insertion sort

- This algorithm sorts one data item at a time
- It is similar to how you might sort a deck of cards
  - One item is taken from the list, and placed in the correct position
  - This is repeated until there are no more unsorted items in the list

Insertion sort

List to be sorted:

9	5	4	15	3	8	11	2

Insertion sort

List to be sorted:

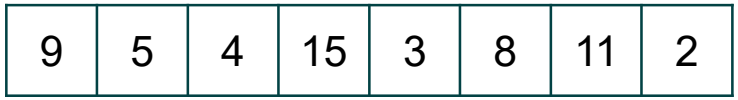
9	5	4	15	3	8	11	2

Leave the first item at the start:

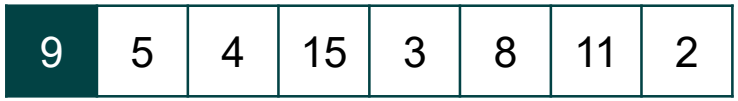
9	5	4	15	3	8	11	2

Insertion sort

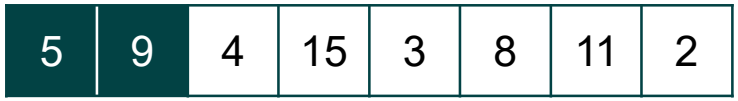
List to be sorted:



Leave the first item at the start:

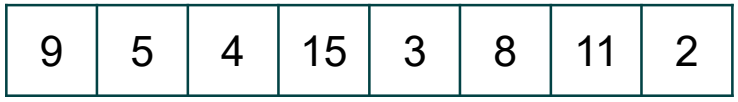


5 is now inserted into the sorted list:

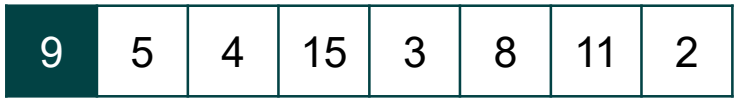


Insertion sort

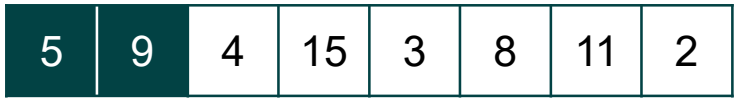
List to be sorted:



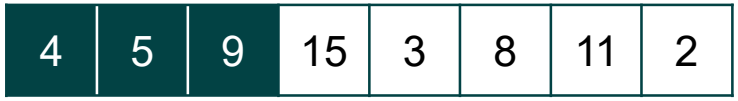
Leave the first item at the start:



5 is now inserted into the sorted list:

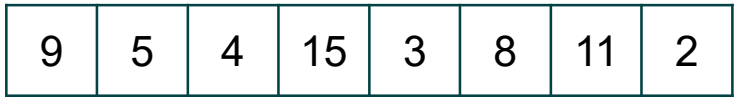


4 is now inserted into the sorted list:

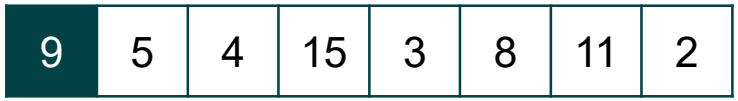


Insertion sort

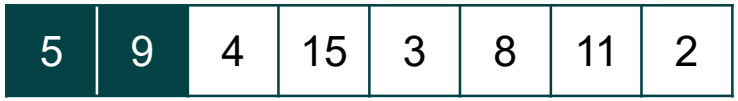
List to be sorted:



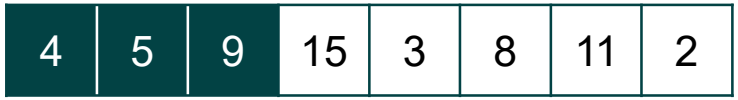
Leave the first item at the start:



5 is now inserted into the sorted list:



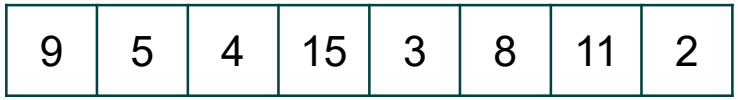
4 is now inserted into the sorted list:



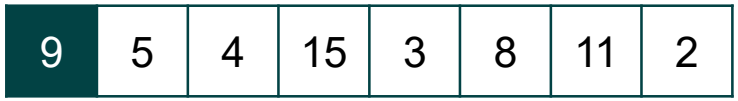
15 is now inserted into the sorted list:

Insertion sort

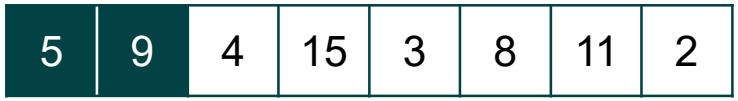
List to be sorted:



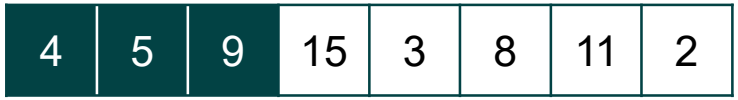
Leave the first item at the start:



5 is now inserted into the sorted list:



4 is now inserted into the sorted list:



15 is now inserted into the sorted list:

3 is now inserted into the sorted list:

Insertion sort

List to be sorted:	9 5 4 15 3 8 11 2
Leave the first item at the start:	9 | 5 4 15 3 8 11 2
5 is now inserted into the sorted list:	5 9 | 4 15 3 8 11 2
4 is now inserted into the sorted list:	4 5 9 | 15 3 8 11 2
15 is now inserted into the sorted list:	4 5 9 15 | 3 8 11 2
3 is now inserted into the sorted list:	3 4 5 9 15 | 8 11 2
8 is now inserted into the sorted list:	3 4 5 8 9 15 | 11 2

Insertion sort

List to be sorted:	9 5 4 15 3 8 11 2
Leave the first item at the start:	9 | 5 4 15 3 8 11 2
5 is now inserted into the sorted list:	5 9 | 4 15 3 8 11 2
4 is now inserted into the sorted list:	4 5 9 | 15 3 8 11 2
15 is now inserted into the sorted list:	4 5 9 15 | 3 8 11 2
3 is now inserted into the sorted list:	3 4 5 9 15 | 8 11 2
8 is now inserted into the sorted list:	3 4 5 8 9 15 | 11 2
11 is now inserted into the sorted list:	3 4 5 8 9 11 15 | 2

Insertion sort

List to be sorted:	9 5 4 15 3 8 11 2
Leave the first item at the start:	9 | 5 4 15 3 8 11 2
5 is now inserted into the sorted list:	5 9 | 4 15 3 8 11 2
4 is now inserted into the sorted list:	4 5 9 | 15 3 8 11 2
15 is now inserted into the sorted list:	4 5 9 15 | 3 8 11 2
3 is now inserted into the sorted list:	3 4 5 9 15 | 8 11 2
8 is now inserted into the sorted list:	3 4 5 8 9 15 | 11 2
11 is now inserted into the sorted list:	3 4 5 8 9 11 15 | 2
2 is now inserted into the sorted list:	2 3 4 5 8 9 11 15

Which item did not have to move?

Worksheet 3

- Now complete **Task 2** on **Worksheet 3**

Merging two lists

List A

6	8	13	20

List B

2	7	11	16

- The following slides show how to merge two sorted lists together
  - Read item from list A; Read item from list B.
  - Write smaller to output list.
  - Read next item from the list that held the smaller value
  - Repeat until all items written to output list

Merging two lists

List A

6	8	13	20

List B

2	7	11	16

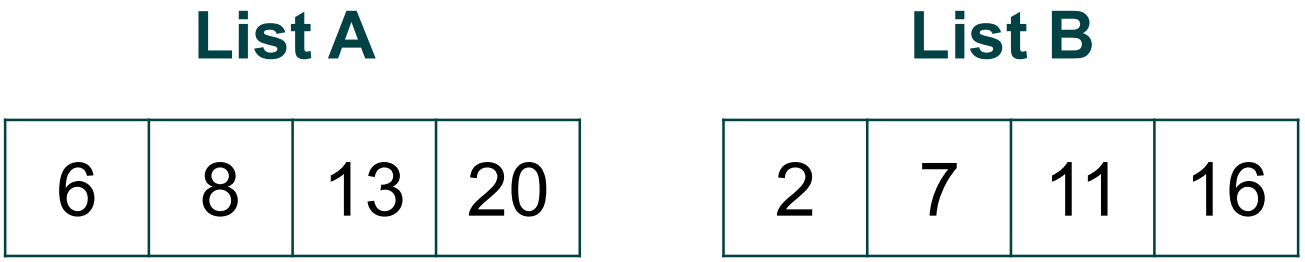
- Read item from list A — 6
- Read item from list B — 2

} Compare and write smaller item to output

Output list

2							

Merging two lists



- Read item from list A
- Read item from list B

6
7

Compare and write smaller item to output

Merging two lists

List A

6	8	13	20

List B

2	7	11	16

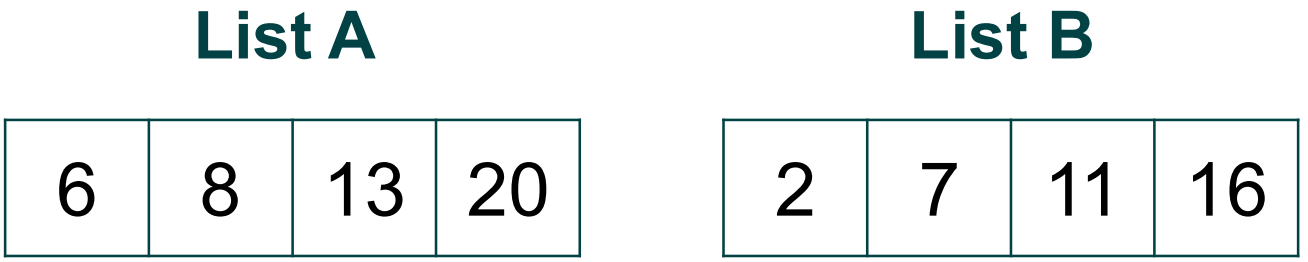
- Read item from list A — 8
- Read item from list B — 7

} Compare and write smaller item to output

Output list

2	6	7					

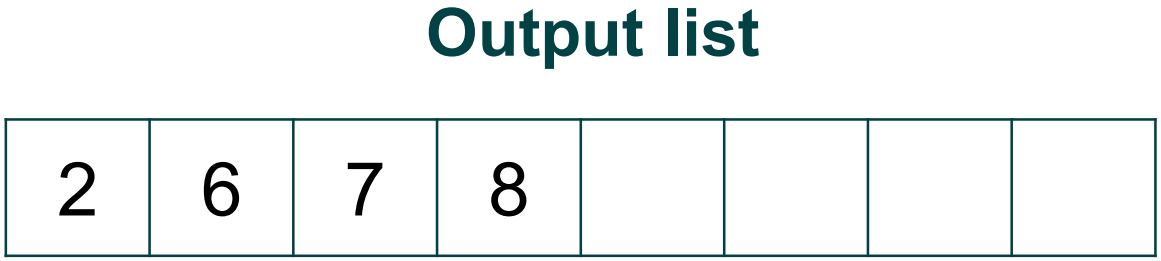
Merging two lists



- Read item from list A
- Read item from list B

8
11

Compare and write smaller item to output



Merging two lists

List A

6	8	13	20

List B

2	7	11	16

- Continue with the rest of the comparisons

Output list

2	6	7	8	11	13	16	20

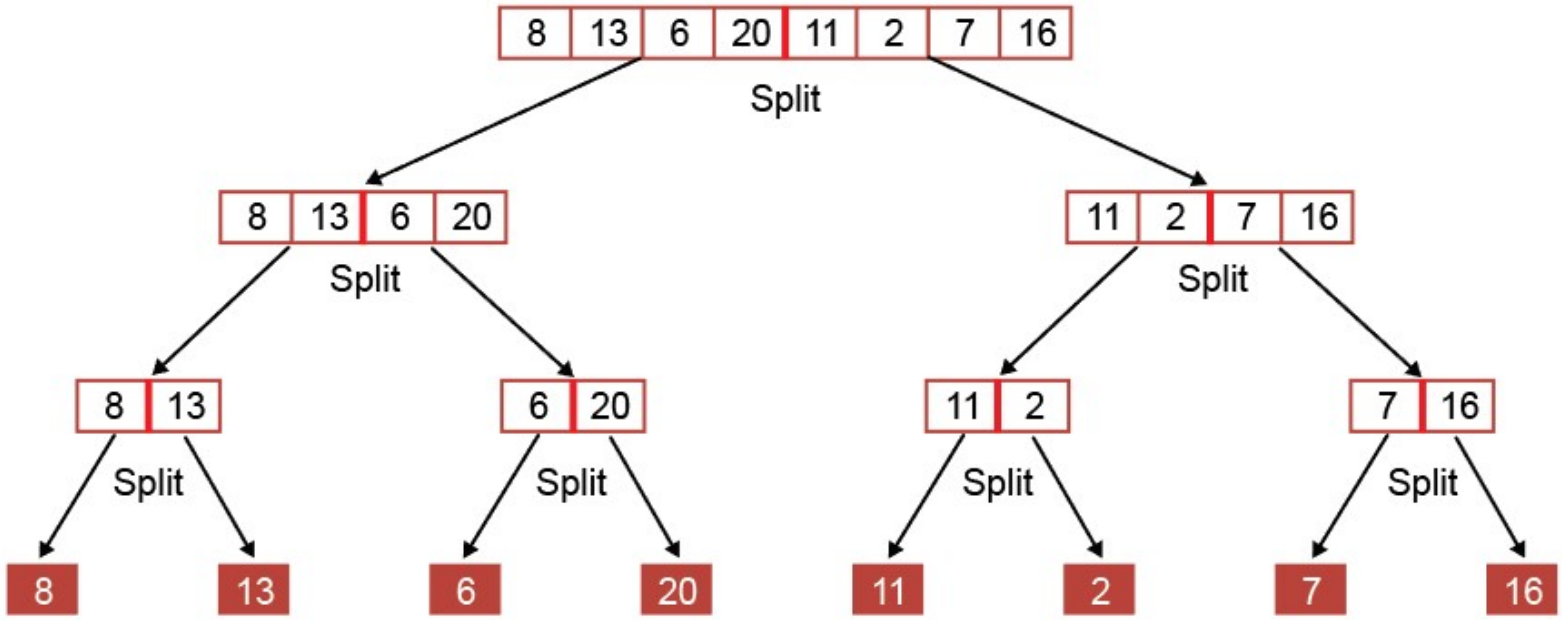
Merge sort

- This is much more efficient than the bubble sort
- The basic steps are:
  - Divide the unsorted list in two
  - Continue to divide these lists into two until there is just one item in each list
  - Now merge each list back until there is only one list remaining – which will be the fully sorted list
- Perform a merge sort on the following list:

8	13	6	20	11	2	7	16

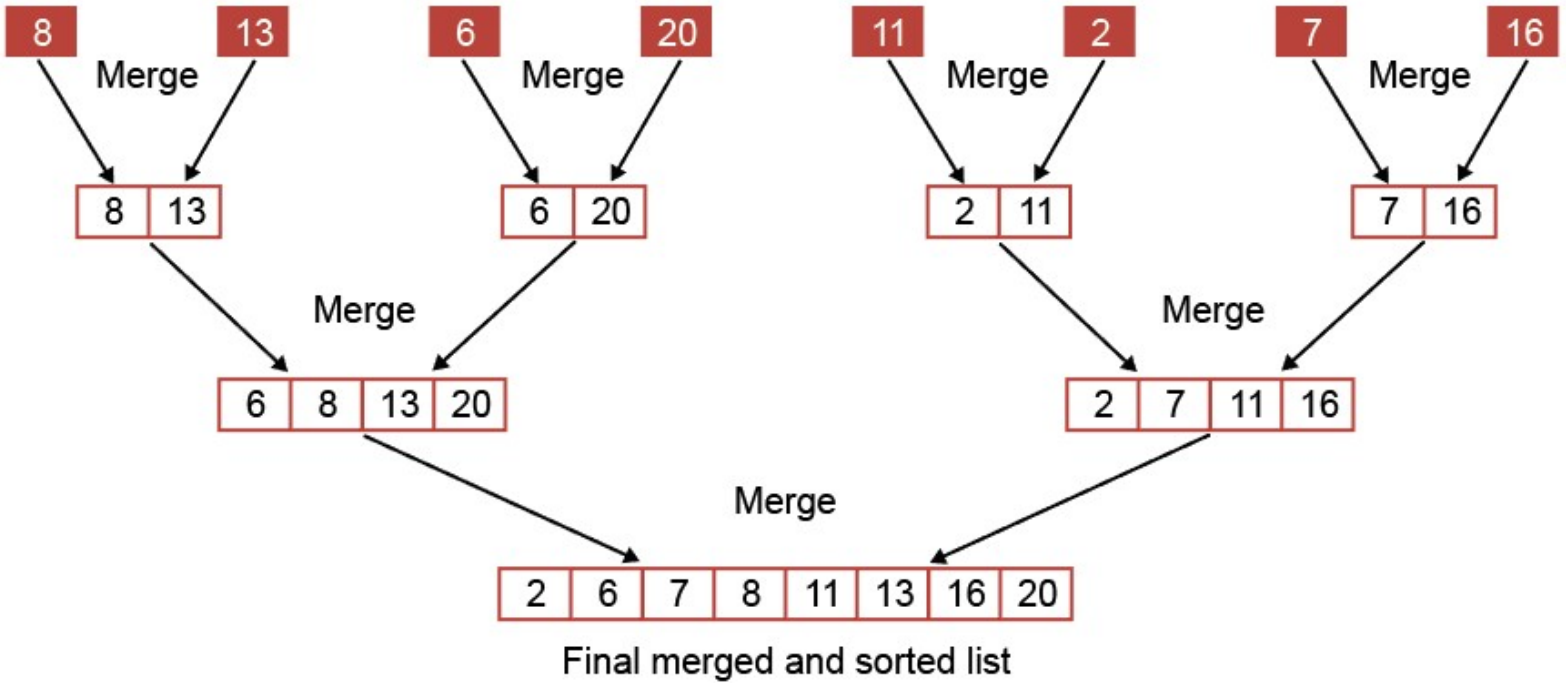
Merge sort part 1

- Divide the unsorted list into n sublists



Merge sort part 2

- Now merge the pairs of sub-lists into a single sorted list

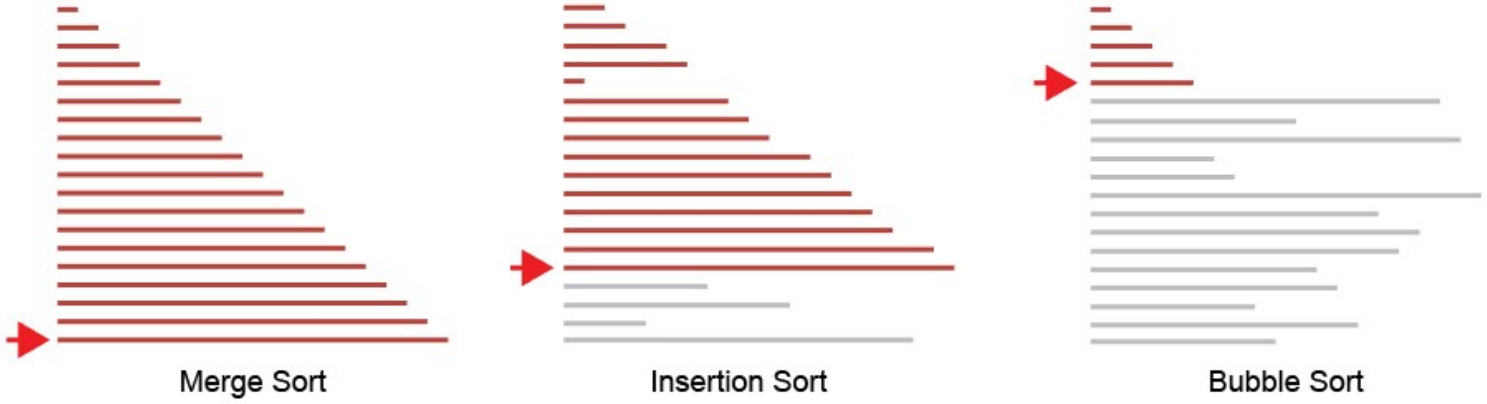


Worksheet 3

- Now complete **Task 3**, **Task 4** and **Task 5** on **Worksheet 3**

Comparing sorts

- The Insertion sort is much quicker than the Bubble sort, but the Merge sort is faster still
- Be sure you can explain how each sort works



Plenary

- Name three algorithms which can be used to sort lists of data
  - Explain briefly how each one works
- Which of these algorithms would you generally choose? Why?

Plenary

Answers

- Sorting algorithms
  - Bubble sort
  - Insertion sort
  - Merge sort
- Choice of algorithm
  - Merge sort is generally faster to sort lists, so would be the recommended algorithm

Copyright

Licence agreement