What is an 'Algorithm'?

An **algorithm** is an unambiguous sequence of steps that can be followed to complete a task. An algorithm is *not* a computer program - a computer program is the implementation of an algorithm.

Representing Algorithms

Key Word	Definition
Input(s)	Data given to an algorithm (e.g. user input to a program).
Processes	Actions performed during an algorithm.
Output(s)	Data returned by an algorithm.
Decomp-osition	Dividing a problem or task into sub-tasks that are easier to accomplish. This makes creating an algorithm quicker and easier.
Abstraction	Removing unnecessary detail from a problem, thus making it easier to solve.
Flow Chart	A visual representation of an algorithm.
Pseudocode	A way of planning code that bears some resemblance to code, but is not necessarily specific to one particular programming language and would not be able to run. Each programmer has their own way of writing pseudocode.

Trace Tables

Trace tables are a technique used to manually analyse the output of an algorithm, which helps to determine the algorithm's purpose. Below is an example.

Algorithm (pseudocode):

```
x = USERINPUT
y = USERINPUT
z = 0
WHILE x < y
 x = x + 1
ENDWHILE
z = x + y
RETURN x, y, z
```

Trace Table:

x	y	z
2	6	0

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com

Trace Tables (cont)

3	6	0
4	6	0
5	6	0
6	6	12

Efficiency of Algorithms

Generally, more than one algorithm can be used to solve a given problem. The one that is usually preferred is the most efficient one.
Although there are multiple ways of measuring efficiency, the time taken to execute a program is the main one. Obviously, a quicker solution is best (so that no time is wasted), so shorter algorithms are preferred.

For the AQA specification, "exam questions in this area will only refer to time efficiency."

Searching Algorithms

Searching algorithms are procedures used to find a specific item of data in a dataset.
A **linear search** begins at the start of the dataset, then moves through it sequentially, testing every element for a match. This means that it can be used on **sorted or unsorted** data and on an array of **any data type**. However, if the array is long then a linear search can be quite **slow** and take a long time.
A **binary search** requires an **ordered** data set, but is generally **quicker** than a linear search. It divides a list in half and 'looks at' the middle element (if there is an even number of elements, the first item on the right is examined). If the element is not equal to the desired value, then the algorithm checks whether the element is greater than the desired value. If so, the left-hand side of the list is discarded. If not, the right-hand side is discarded. This is repeated until the correct element is found.. Both search algorithms **cannot search for multiple occurences** of an element in the same data set.

Sorting Algorithms

Sorting algorithms are procedures used to arrange a data set into an order (usually ascending).
A **bubble sort** starts at the beginning of the data set. It then goes through it sequentially, checking each item against the next one. If the first item is greater than the second one, they are swapped. This is repeated until it reaches the end of the data set. It then makes a second pass, doing the same thing. Once it makes one full pass without making any changes, it terminates because the list is in order.
A **merge sort** starts by dividing the list in half repeatedly, until it is left with individual elements. It then merges these elements into pairs, with the smallest element on the left-hand side. This is repeated until all of the sublists are merged into one, which is now ordered.
Bubble sorts are **extremely inefficient**, whereas merge sorts are **very fast**. However, merge sorts need **additional storage** because there are many more stages.

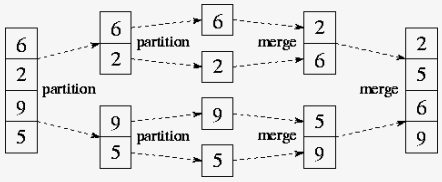
Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com

Bubble Sort

5 1 12 -5 16	unsorted
5 1 12 -5 16	5 > 1, swap
1 5 12 -5 16	5 < 12, ok
1 5 12 -5 16	12 > -5, swap
1 5 -5 12 16	12 < 16, ok
1 5 -5 12 16	1 < 5, ok
1 5 -5 12 16	5 > -5, swap
1 -5 5 12 16	5 < 12, ok
1 -5 5 12 16	1 > -5, swap
-5 1 5 12 16	1 < 5, ok
-5 1 5 12 16	-5 < 1, ok
-5 1 5 12 16	sorted

http://www.algolist.net/Algorithms/Sorting/Bubble_sort

Merge Sort



http://www.mcs.anl.gov/itf/dbpp/text/node127.html

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com