Tackling Computer Science at Key Stage 3

Jane Sinclair and Russell Boyatt

Department of Computer Science, University of Warwick,
Coventry, CV4 7AL, UK

19th April 2012

3 comes before 4!

- Main topic for today is KS 4 and CS GCSE
- But ideally it doesn't just start there.
- Good for students to start "thinking like a computer scientist" and trying out practival activities as early as possible.
- Makes good sense from a teaching perspective too: progression; reaching *all* students.
- Integrating with other aspects of the curriculum from the start.

In this session

- Consider one specific example of one specific area of KS3 computer science
- Discuss some ideas/resources used on a recent Y8 day.
- Hear from you about the current state of play/prospects/support needs etc for KS3.
- Provide a starting point to consider possibilities and needs at this level.

Steve's homework

The CAS curriculum covers a range of interesting and inspirational topics - *at all stages*.

"Every pupil at KS2 and KS3 should have the opportunity to learn material that is recognisably Computer Science".

"Every pupil should appreciate that computational ideas inform and illuminate other disciplines and this should be reflected in the teaching of these disciplines at school."

Steve's homework

The CAS curriculum covers a range of interesting and inspirational topics - *at all stages*.

"Every pupil at KS2 and KS3 should have the opportunity to learn material that is recognisably Computer Science".

"Every pupil should appreciate that computational ideas inform and illuminate other disciplines and this should be reflected in the teaching of these disciplines at school."

Great syllabus - covers many really exciting areas.

Steve's homework

The CAS curriculum covers a range of interesting and inspirational topics - *at all stages*.

"Every pupil at KS2 and KS3 should have the opportunity to learn material that is recognisably Computer Science".

"Every pupil should appreciate that computational ideas inform and illuminate other disciplines and this should be reflected in the teaching of these disciplines at school."

Great syllabus - covers many really exciting areas.

But how to tackle it in practice? Where are the resources? How to link to other aspects of the curriculum?

Resources

The good news

There's lots of resources - many free online.

Many great ideas; lesson activities; free technologies/open source software.

It's a subject which lends itself to engaging presentation, student participation and discovery-led learning.

Resources

The good news

There's lots of resources - many free online.

Many great ideas; lesson activities; free technologies/open source software.

It's a subject which lends itself to engaging presentation, student participation and discovery-led learning.

The bad news

May not be obvious where resources are/which to choose/how to use them.

Difficulty even in deciding which programming language to choose!

"Learning in abundance"?

Resources

The good news

There's lots of resources - many free online.

Many great ideas; lesson activities; free technologies/open source software.

It's a subject which lends itself to engaging presentation, student participation and discovery-led learning.

The bad news

May not be obvious where resources are/which to choose/how to use them.

Difficulty even in deciding which programming language to choose!

"Learning in abundance"?

The language issue is important (more later)

. . . but not the only area where resources and activities are needed.

Algorithms

One of the 5 key areas identified in the curriculum.

Essential to other areas too (programming, data structures).

Algorithms

One of the 5 key areas identified in the curriculum.

Essential to other areas too (programming, data structures).

An algorithm is a sequence of instructions for achieving a goal

Algorithms

One of the 5 key areas identified in the curriculum.

Essential to other areas too (programming, data structures).

An algorithm is a sequence of instructions for achieving a goal

An algorithm is an effective method expressed as a finite list of well-defined instructions for calculating a function

Algorithms

One of the 5 key areas identified in the curriculum.

Essential to other areas too (programming, data structures).

An algorithm is a sequence of instructions for achieving a goal

An algorithm is an effective method expressed as a finite list of well-defined instructions for calculating a function

Fundamental for computer science.

From very simple tasks (eg: summing a sequence of numbers)

. . . to standard, ubiquitous functions (eg: searching for a required value in some data structure)

. . . to clever stuff (eg: factorising very large primes).

Algorithms - starting young!

KS1 (5-7)
The idea of setting out predefined steps to achieve a goal (eg: in natural language, story board etc).

Algorithms - starting young!

KS1 (5-7)
The idea of setting out predefined steps to achieve a goal (eg: in natural language, story board etc).

KS2 (7-11)
Representation in formal language or flow chart. Use of conditional and iteration statements.

Algorithms - starting young!

KS1 (5-7)
The idea of setting out predefined steps to achieve a goal (eg: in natural language, story board etc).

KS2 (7-11)
Representation in formal language or flow chart. Use of conditional and iteration statements.

KS3 (11-14)
Understanding different algorithms to achieve the same goal. Factors of comparison (complexity, memory, etc).

Algorithms - starting young!

KS1 (5-7)
The idea of setting out predefined steps to achieve a goal (eg: in natural language, story board etc).

KS2 (7-11)
Representation in formal language or flow chart. Use of conditional and iteration statements.

KS3 (11-14)
Understanding different algorithms to achieve the same goal. Factors of comparison (complexity, memory, etc).

KS4 (14-16)
Relationship to datastructures; design of algorithms; familiarity with some well-known ones.

Algorithms - starting young!

KS1 (5-7)
The idea of setting out predefined steps to achieve a goal (eg: in natural language, story board etc).

KS2 (7-11)
Representation in formal language or flow chart. Use of conditional and iteration statements.

KS3 (11-14)
Understanding different algorithms to achieve the same goal. Factors of comparison (complexity, memory, etc).

KS4 (14-16)
Relationship to datastructures; design of algorithms; familiarity with some well-known ones.

And that's just the beginning.

Year 8 day

Mixed year group from St Alban's Academy, Birmingham.

Range of activities - including algorithms.

Introduced ideas through a variety of physical activities - "computing unplugged".

Motivation through discussion and relationship to everyday activities.

We didn't have time to move on to programming - but would be obvious next step.

Some opportunity to bring in relationship to other areas.

Chose the fertile algorithm area of sorting.

Motivating discussion

Keeping things in order so you can find them. Eg: putting things away!

Motivating discussion

Keeping things in order so you can find them.

Eg: putting things away!

Lots of things in everyday life are much easier to work with if put into a useful order.

Eg: books in school library; printed dictionary; train timetable; clothes in a shop

Motivating discussion

Keeping things in order so you can find them.

Eg: putting things away!

Lots of things in everyday life are much easier to work with if put into a useful order.

Eg: books in school library; printed dictionary; train timetable; clothes in a shop

Different orderings. Nested orderings.

Eg: alphabetical; day/time; type of clothes/sizes

Motivating discussion

Keeping things in order so you can find them.

Eg: putting things away!

Lots of things in everyday life are much easier to work with if put into a useful order.

Eg: books in school library; printed dictionary; train timetable; clothes in a shop

Different orderings. Nested orderings.

Eg: alphabetical; day/time; type of clothes/sizes

Sorting means putting things into order. May need to resort for different purposes.

Eg: to look up a number in printed phone directory

Motivating discussion

Keeping things in order so you can find them.

Eg: putting things away!

Lots of things in everyday life are much easier to work with if put into a useful order.

Eg: books in school library; printed dictionary; train timetable; clothes in a shop

Different orderings. Nested orderings.

Eg: alphabetical; day/time; type of clothes/sizes

Sorting means putting things into order. May need to resort for different purposes.

Eg: to look up a number in printed phone directory

Sorting means work to be done.

Motivating discussion

Keeping things in order so you can find them.

Eg: putting things away!

Lots of things in everyday life are much easier to work with if put into a useful order.

Eg: books in school library; printed dictionary; train timetable; clothes in a shop

Different orderings. Nested orderings.

Eg: alphabetical; day/time; type of clothes/sizes

Sorting means putting things into order. May need to resort for different purposes.

Eg: to look up a number in printed phone directory

Sorting means work to be done.

Computers have huge amounts of data to sort.

Eg: my 25,000 emails; facebook pictures; Google data centres

Getting the computer to do it for us

Computers are good at doing repetitive tasks - very fast.

But we have to tell them what to do - in terms they can understand and steps they are able to perform

Eg: asking your maths teacher how to solve a problem and getting the response: replace the infinity with a variable, calculate the integral and then take the limit of the result as the variable tends to infinity.

The computer will only do what we tell it!

For sorting - what steps can a computer understand easily and perform efficiently?

Compare any 2 values at a time. Swap those values if we want.

Activity 1: selection sort

We're going to demonstrate one of these sorting activities...

Activity 1: selection sort

We're going to demonstrate one of these sorting activities... with your help!

Activity 1: selection sort

We're going to demonstrate one of these sorting activities... with your help!

Selection sort is an algorithm for sorting a list of items.

We'll demonstrate with numbers.

Inefficient – a lot of comparisons required.

Our experience shows it is a simple algorithm to demonstrate.

Activity 1: selection sort

The selection sort algorithm...

1. Start with an unsorted list of numbers.

Activity 1: selection sort

The selection sort algorithm...

1. Start with an unsorted list of numbers.

2. Find the minimum number in that list.

Activity 1: selection sort

The selection sort algorithm...

1. Start with an unsorted list of numbers.
2. Find the minimum number in that list.
3. Move that number to the sorted list.

Activity 1: selection sort

The selection sort algorithm...

1. Start with an unsorted list of numbers.
2. Find the minimum number in that list.
3. Move that number to the sorted list.
4. Repeat and perform selection sort on remaining unsorted list.

Activity 1: selection sort

The activity can be extended beyond the physical sorting.

Activity 1: selection sort

The activity can be extended beyond the physical sorting.

Developing flow charts that describe the steps in the algorithm (KS2)

Comparison of algorithms to explore efficiency (KS3)

Learning about different sorting algorithms and designing appropriate data structures (KS4)

Activity 1: selection sort

The activity can be extended beyond the physical sorting.

Developing flow charts that describe the steps in the algorithm (KS2)

Comparison of algorithms to explore efficiency (KS3)

Learning about different sorting algorithms and designing appropriate data structures (KS4)

Next step would be to explore selection sort using a programming language. Resources available to support these activities.

Other sorting algorithms

Quicksort algorithm

Students given cups filled with sand and a balance scales. Asked to sort cups using balance scales to simulate the computer comparing two items.

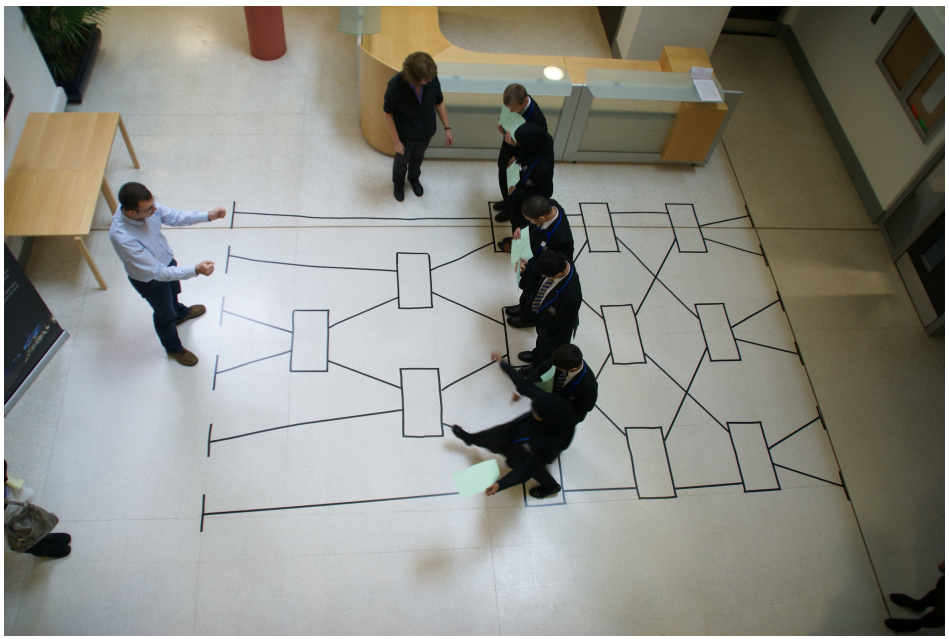
Students perform quicksort algorithm (more efficient that selection sort) and asked to record each comparison.

Other sorting algorithms

Network sort

Large sorting network marked on the floor using tape. Demonstrates how comparisons can be performed simultaneously.

Explore algorithm efficiency and how large systems work.



Resources for algorithms

Many examples for programming environments such as Scratch, Greenfoot and others.

Computer Science Unplugged – offers suggestions for practical activities
`www.csunplugged.org`

Scratch sorting algorithms – common sorting algorithms in Scratch
`http://code.google.com/p/scratch-unplugged/`

Many videos demonstrating the algorithms – for both students and teachers.

Further activities

This is just a quick idea of a few activities relating to sorting.

Different ones suitable to different abilities

Lots more sorting algorithms to explore!

Lots more issues of comparitive efficiency (and other aspects affecting choice of algorithm).

More opportunities for linking to maths

Data structures (eg: heap sort)

Uses of sorting - exploration of current uses and needs.

Algorithms for other interesting tasks

Designing algorithms and programming.

Over to you

Discussion in groups

- Where is your school currently at with KS3 computer science? Where do things stand on a scale of "ICT as usual" to "full implementation of CAS curriculum"? How much teaching time is allocated? Are things changing?
- Are there parts of the CAS curriculum which you feel would be more difficult to deliver or to develop resources for? If so, which?
- What non-programming resources and support would you find useful for KS3 teaching?

Please choose someone to note the main points of discussion from your group and report back briefly at the end.