Junior Computer Science

code-it.co.uk

Exchange Sort Investigation

Module Aim Pupils investigate all the different ways two cards can be compared and then test to see if these can be turned into working sorting algorithms

Program of Study use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs

Rationale

Many computer science lessons on sorting concentrate on explaining how different sorting methods work. This lesson encourages pupils to discover some of the more simpler sorting methods for themselves. Once the class has come up with a range of possible methods they can test these to see which ones will really sort numbers.

Resources Needed
Number cards enough for six cards between two pupils
Pupil Recording Sheet (one copy per pair of pupils)
One pencil per pair of pupils
Sort Instructions slides PDF
Pupil testing sheet (one copy per pair of pupils)

This lesson links to the slides on the Sort Instruction PDF

Introduction

Explain to the class that they are going to investigate sorting but first we need to know why sorting is important. Ask them to imagine how difficult it would be to find the word apple in a dictionary if the dictionary definitions were not sorted alphabetically. Ask them how they might go about this? Common answers might include randomly opening a page or starting at front and scanning page at a time through. Go to slide 3 and ask pupils to shout out what the largest number is (89) they will get that very quickly. Now go to slide 4 and ask the same question. Ask the same question for slide 5. Draw out from them that it is the amount of numbers that makes this increasingly difficult for us as humans. A computer following a sorting program/algorithm could find the largest number in that list in less than a second.

Finding methods to compare two cards

Explain that humans can quickly scan groups of things like our first six numbers but that we struggle when we need to compare bigger and bigger groups of things. A computer however can only see and compare two numbers/items at a time but it can make these comparisons very very quickly (slide 7). Our first task in inventing a sorting algorithm that a computer could be programmed to do would be to find the right patterns or method to compare two numbers at a time (slide 8). Separate pupils into pairs and give each pair six number cards. Ask them to deal these out face down in a straight line (slide 8).

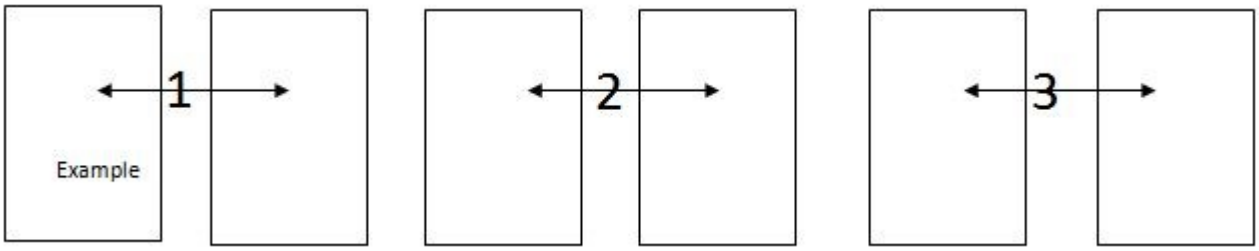
Junior Computer Science

code-it.co.uk

Exchange Sort Investigation P2

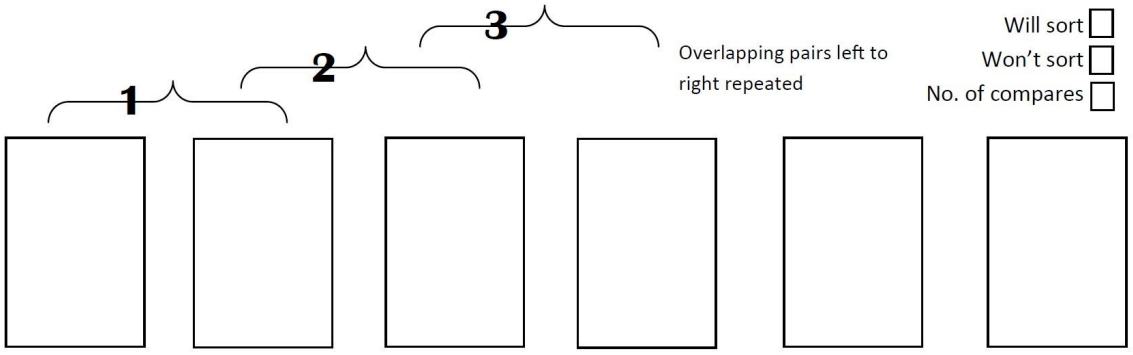
Give out the recording sheet and demonstrate how to record comparisons using lines. Use slides 9-14 to help them understand the task by demonstrating one method to comparing pairs of cards in pairs next to each other. It helps understanding to then demonstrate this method with the cards and point out the example annotation. Point out at the end that They can use same card more than once. They can do more than three compares.

Once pupils have had time to find lots of methods/patterns collate these as a class and see if the pupils have found the ones on the pupil testing sheet pdf



Testing their patterns

Ask pupils to shuffle the six cards and deal them face up in a line. If pupils have found most of the patterns themselves they can test these. If they haven't you can hand out the pupil testing sheet. Explain that they now need to test these patterns to see which ones will sort and which ones won't sort. It really helps to demonstrate this yourself using one pattern. Stress the need to repeat lots of times until it has sorted or not. Tell pupils to ignore the No. of compares. TEACHER NOTE In the future the number of compares will be important for finding out which methods are quickest but six cards are not a big enough sort sample to be useful.



Junior Computer Science

code-it.co.uk

Exchange Sort Investigation P3

Explain that we are going to use one simple rule (slide 15) if the number on the left is greater than the number on the right we are going to swap the cards. Allow pupils lots of time to test the patterns.

You may need to point out that they may need to repeat the patterns lots of times before the cards are sorted. If they are not sure if a pattern worked they can shuffle the cards and try that pattern again. It is possible that pupils will deal the cards out ready sorted. Deal them again if this happens.

Conclusion

Ask pupils to feed back their results pattern by pattern. What patterns sorted the cards and what patterns didn't? Explain that the ones that worked, apart from the random method, are all part of the exchange sort family of sorting algorithms. They are called bubble sorts because sometimes one number has to bubble to the end of the line a place at a time through lots of repeats. They are not the fastest family of sorting algorithms so are less used by computers these days.

Extension Activity

Can they sort each others heights using some of these methods and video them. You could also ask pupils to take one working method and write it out as an algorithm so that someone else could use it.

More Resources

Bubble Sort on Wikipedia

Efficiency

All the working algorithms discovered here are variations of an exchange sorts. Some are marginally more efficient but it would take a lot of testing with much higher ranges of numbers to notice this.

The exception is the random sort which would on average take longer. The most famous exchange sort is the bubble sort.