Classroom resources

- Chalk

⊕ Learning outcomes

Students will be able to:

- Explain how a Sorting Network parallel algorithm works.
  Computational Thinking: Algorithmic Thinking
- Identify which number comes before or after in a given range of numbers.
  Mathematics: Numeracy
- Organise objects from smallest in size to largest in size.
  Mathematics: Numeracy

Key questions

What are examples of tasks get finished sooner if more people help with them? What are examp
finished sooner if more people help with them?

Potential answers could include:

Tasks such as tidying the classroom, picking up rubbish, or reshelving library books may come u
multiple helpers. Things that don't go faster might include delivering a note to the office (10 pe
get it there 10 times faster), or washing dishes if there is only one sink (two people are faster th
probably can't speed it up).

Lesson starter

⊕ See teaching this in action

Some other videos showing different situations using Sorting Networks:

- Video 1
- Video 2

Use the Sorting Network template to draw a 6 person Sorting Network on a paved surface outs
alternatives include using masking/painters tape on a carpet or wooden floor, tape on a tarpaul
grass). Note that the Sorting Network needn't use different colours or line thicknesses, but if su
this can help students remember which way to go. It should be large enough that two students
rectangles; the more spread out it is, the more effective the exercise is. In a very confined situat
top using game counters instead of students moving around, but this is much less engaging.

Show the students the Sorting Network drawn on the ground, and tell them "This chalk comput
let's investigate what it does."

Mathematical links

Supports students understanding of ordering any range of numbers, from ordering single dig
decimals, or numbers in their millions.

Lesson activities

1. Organise students into groups of six. Only one team will use the network at a time.
2. The current team should stand on the circles at the "input" end of the Sorting Network.
3. Give each of the six students a card to hold (initially use the set of cards containing the nu
   should be given to the students out of order). These cards are the inputs into this cool cha
   kind of computer that can process several operations at the same time).
4. Get the first two students to follow the lines from their circles until they meet at a box (the

the group repeat the task and check each comparison. If it doesn't work a second time, bring i
that each square has made the right decision which person is to go to the left and the right. E
when they meet at a square helps to avoid someone heading off before they have made a dec

If a student races to the end ahead of everyone else because they already know where their n
numbers are sorted (this happens quite often!) then some students are going to be left stuck i
they don't have someone to compare numbers with. This is a good opportunity to remind stu
follow the instructions they are given precisely to make sure they achieve the correct result; it
teamwork!

Applying what we have just learnt

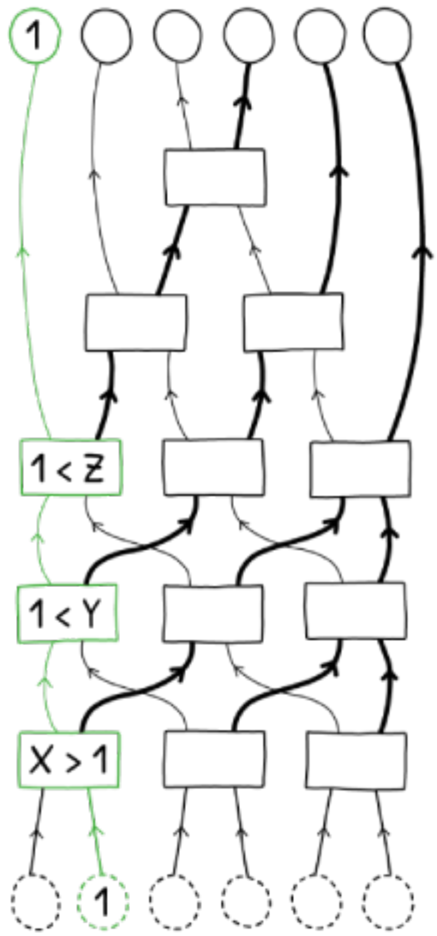
This technique with parallel instructions can't run directly in the kind of computer system that s
about, as simpler systems can only compare one pair of values at a time, while this one is comp
at the same time. But although this algorithm hasn't been written to work on a conventional sy
to be observed, a parallel algorithm, and this can be implemented with specialised hardware ar
creating parallel algorithms is to have as many things happening at the same time as possible,
faster. However, it's not always easy to break a problem up so that separate parts can happen a
comparison depends on the results of another. The diagram that we used above happens to be
for sorting 6 values.

How do we know this Sorting Network is reliable and works every time?

The outcome we want to achieve is that the numbers come out in the correct order with the sm
most box and the second smallest number finishing next to it, right through to the largest num
If we want to make sure it works for all possible inputs, then we would need to try it for every o
in - it turns out that there are 720 (6x5x4x3x2x1) different orderings that 6 items can start in, so
sorting more than 6 items, there are way too many different orderings to try out, so we must m
it works. Here are some elements of such a proof:

Let's disregard the numbers for now and look at the Sorting Network from the point of view of t
smallest number was in node 1, what path would it take and does it end up in the leftmost nod

Now repeat this by asking if the smallest number was in node 2, what path would it take and do
node at the end?



Repeat this until you've tested all 6 nodes. If the smallest number ends up in the leftmost node
that's part-way to being sure that the structure always works.

You can repeat this with the largest number - no matter where it starts, it will always end up in t

Doing this for the other four values (e.g. the second to largest) isn't quite as simple, but comput
that they will also always end up in their correct position.

Lesson reflection

- Was there ever a situation where the cards weren't sorted in the right order? What had ha
was it corrected?
- Can you trace the pathways for the lowest number if it was placed in any position? What a

Seeing the Computational Thinking connections

Throughout the lessons there are links to computational thinking. Below we've noted some ger
content.

Teaching computational thinking through CSUnplugged activities supports students to learn ho
what are the important details they need to solve this problem, break it down into small logical
create a process which solves the problem, and then evaluate this process. These skills are trans
area, but are particularly relevant to developing digital systems and solving problems using the

These Computational Thinking concepts are all connected to each other and support each othe

The Sorting Network used in these activities is itself an abstract representation of how Sorting
hardware and software. It represents the core functionality of a Sorting Network, whilst hiding
how the hardware and circuitry works.

Examples of what you could look for:

Can students make the connection between the lines and nodes on this graph and the way co
information by making comparisons? Can students understand that this representation can be
parallel processing computer would work?

Decomposition

The whole process of sorting in this activity is decomposed into a very simple operation: comp
operation alone is very simplistic, but when it is performed many many times it can be used to
larger task.

Examples of what you could look for:

Can students see how to design a Sorting Network to sort just 2 values? (It would just be a sin

Generalising and patterns

In this lesson students only worked with one type of information, numbers, so there wasn't mu
more prominent in the next lesson.

Evaluation

For this Sorting Network there can be up to three comparisons happening at once, and the len
how long it would take to complete all these comparisons. Although 12 comparisons need to b
the network, the network can be completed in the time it takes to an individual node to make

Examples of what you could look for:

Can students identify the longest path that any number would have to go through to get to th
numbers need to make 5 comparisons). Can students explain that if every comparison were to
sorting would be finished in 5x2 seconds, and not 12x2 seconds?

Logic

The smallest value will always take the left path at any comparison, and from every starting po
the left branch will lead to that node, the smallest value will therefore always end up in the lef