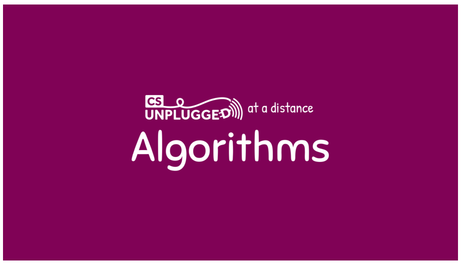
Algorithms (Speaker Notes)

Slide 1

No speaker notes for this slide.



CS UNPLUGGED at a distance

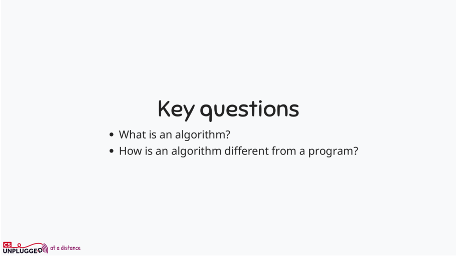
Algorithms

Slide 2

In today's session we will look at these two questions: what is an algorithm, and how is it different from a program.

Algorithms are sometimes compared to recipes, or are said to be steps for a solution, or something that works on a computer.

We are going to do an activity to help clarify exactly what an algorithm is.

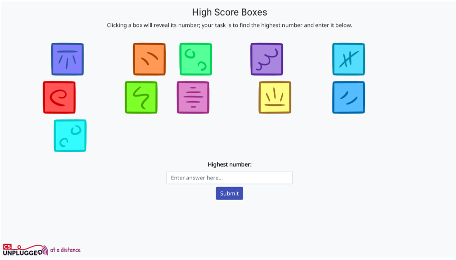


Key questions

- What is an algorithm?
- How is an algorithm different from a program?

CS UNPLUGGED at a distance

Slide 3



High Score Boxes

Clicking a box will reveal its number; your task is to find the highest number and enter it below.

In this challenge we will be working out which is the highest score from a group of game players. Each box shows a player's score.

When I click on this box [click on a random box] the score it shows is [say the score shown]. We only get to see what is in each box one at a time. I will go through all the boxes. Let's see if we can agree on what the highest score was.

Click on any box and read out the number revealed. Note down the first number to refer to later. Continue to click on each box and read out the numbers until all have been revealed.

Write in the chat what you think the highest score was.

Wait until there are at least a couple of replies, they will probably be the same.

I'll type our answer [the answer suggested in the chat] in here and see if we are right.

Type in the suggested number into the submission box.

We were right! A handy thing with this app is that each time you use it it will have a different high score.

Now, how did we work out what the high score was? I showed you 11 numbers, did anyone remember all of them and then work out which was the highest at the end?

Pause for responses from participants (either in chat or by their cameras).

I see a whole lot of shaking heads.

Can someone share with us how they worked out that high score?

Either let participants unmute (quickest), or have them write it in the chat. Someone will probably give an answer similar to the following:

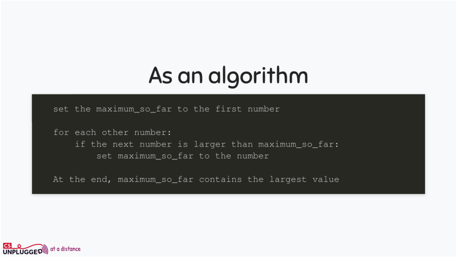
> That's right, we remembered the first number, then compared it to the next number. If the next number was bigger than the first number then we forgot the first number and remembered only the most recent number. If the new number was smaller, we ignored it and went on to the next number. We remembered the highest score so far, not all the numbers we had seen.
>
> How many numbers did we need to remember at a time for this activity? Type your idea in the chat.

Wait for a couple of replies.

> Two, the number that was in your head already, and the new number you are comparing it with. Now, what we have just described is an algorithm.

Slide 4

> This is a pseudo code. It's not a programming language, but it is trying to articulate what we did using a language between the English language and a computer language.



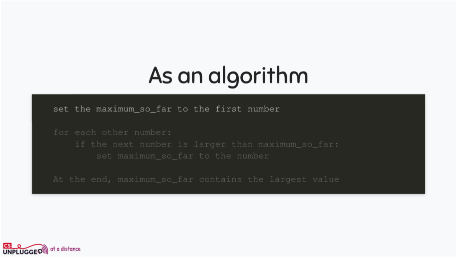
As an algorithm

```
set the maximum_so_far to the first number

for each other number:
    if the next number is larger than maximum_so_far:
        set maximum_so_far to the number

At the end, maximum_so_far contains the largest value
```

Slide 5

The first thing we did was find out the first number. The first number was [say recorded first number]. We had nothing to compare it to, so it was our biggest number so far. What shall we call it for our program? When we were talking about it we called it the 'number in our head' which is fine, or would we call it the 'biggest number'? Except that it might not be the biggest number. Often when you are doing this with students you will come up with the 'biggest so far' as the name for this number. Coming up with a useful name for this is very important as it helps to articulate the program and make it easy to understand. It will be a variable in a program at some point. You could just call it something like 'a', but this could get confusing, not only for yourself as you go on writing the program, but also for anyone else who reads your program.

This is where programming uses literacy skills: how could you describe the role of that number in your head using just 2 or 3 words so that someone else could look at it and know exactly what you mean? Computer programs are not just written for the user, but also for the next programmer who may have to fix or adapt it.

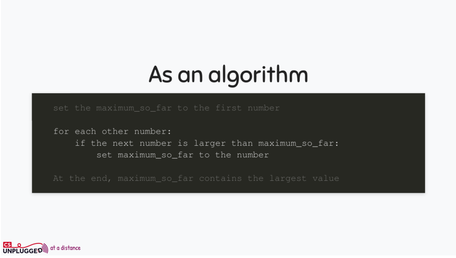


As an algorithm

```
set the maximum_so_far to the first number

for each other number:
    if the next number is larger than maximum_so_far:
        set maximum_so_far to the number

At the end, maximum_so_far contains the largest value
```

Slide 6

The next lines in our code, for all the other numbers I read out, if any of them is larger than the maximum so far, then the 'maximum so far' is changed to that new number.

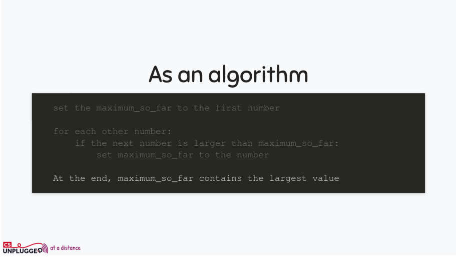


As an algorithm

```
set the maximum_so_far to the first number

for each other number:
    if the next number is larger than maximum_so_far:
        set maximum_so_far to the number

At the end, maximum_so_far contains the largest value
```

Slide 7

At the end, the one in your head, the value of 'maximum so far' will be the maximum value.



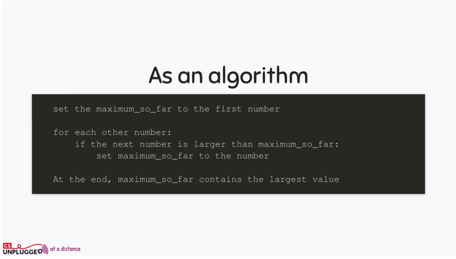
As an algorithm

```
set the maximum_so_far to the first number

for each other number:
    if the next number is larger than maximum_so_far:
        set maximum_so_far to the number

At the end, maximum_so_far contains the largest value
```

Slide 8

This program could have been recorded as a flow chart, or in English without all the indentation and other formatting, whatever format you use, it has captured the idea of what you do. This is the algorithm - it's not a computer program yet, but it's a specific way of getting the result. We were all doing this in our heads, but now we want to turn it into a computer program.

When writing a program, start with the problem you want to solve, then turn it into a general algorithm of what you want it to do so you have an idea of what will work. Then we turn it into an actual program. This is a good process for students, writing out a general algorithm first, before starting to program.



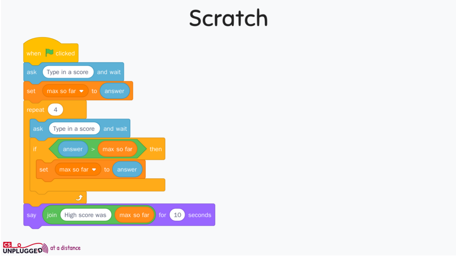
As an algorithm

```
set the maximum_so_far to the first number

for each other number:
    if the next number is larger than maximum_so_far:
        set maximum_so_far to the number

At the end, maximum_so_far contains the largest value
```

CS UNPLUGGED at a distance

Slide 9

Here is a Scratch program for this algorithm. It starts by asking for a score. It stores that first number as a variable.

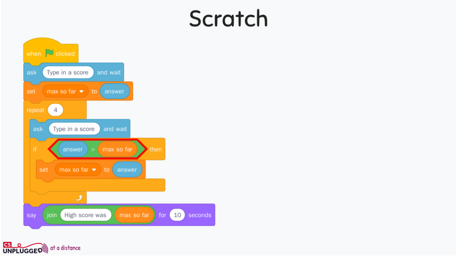


Scratch

```
when 🏁 clicked
ask (Type in a score) and wait
set [max so far ▾] to (answer)
repeat (4)
  ask (Type in a score) and wait
  if <(answer) > (max so far)> then
    set [max so far ▾] to (answer)
say (join (High score was) (max so far)) for (10) seconds
```

Slide 10

Here's the block that checks if the new number is higher than the previous maximum.



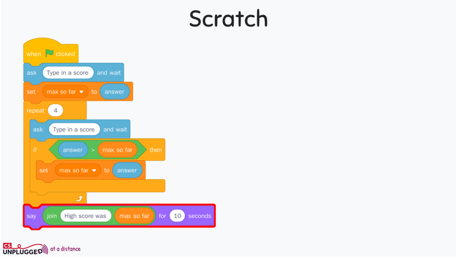
Scratch

```
when 🏁 clicked
ask (Type in a score) and wait
set [max so far ▾] to (answer)
repeat (4)
  ask (Type in a score) and wait
  if <(answer) > (max so far)> then
    set [max so far ▾] to (answer)
say (join (High score was) (max so far)) for (10) seconds
```

Slide 11

At the end we say what the high score was.

To keep this program simple, it only handles numbers. It would require extra blocks to test the inputs for invalid inputs like letters.



Scratch

```
when 🏁 clicked
ask (Type in a score) and wait
set [max so far ▾] to (answer)
repeat (4)
  ask (Type in a score) and wait
  if <(answer) > (max so far)> then
    set [max so far ▾] to (answer)
say (join (High score was) (max so far)) for (10) seconds
```

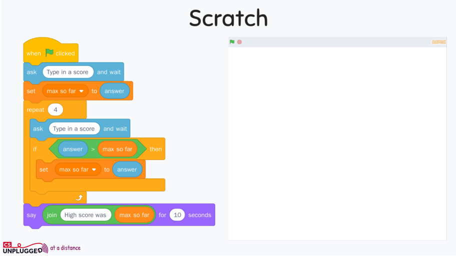
Slide 12

We can try it out here. You can see that this program is displaying the variable. I'm going to try a few different numbers.

Click on the green flag in the top, left hand corner of the inserted Scratch screen, then type 5 random numbers, saying each out loud as you go and pressing `Enter` between each one, into the box at the bottom. Pause after the fifth entry to read what the cat says. It will say the highest number you entered. Point out that the orange 'max so far' variable shown under the green flag only changes when the last number is bigger than the previous 'max so far'.

When you write your own program remember to test it. Check what happens if the first number is the biggest number, if the last one is bigger etc.

Of course, this isn't the only algorithm that will find out the biggest score. Another one could remember all the numbers, sort them into number order smallest to biggest, and take the last number as the biggest number. We can compare algorithms and work out which is more efficient. Hint, it doesn't involve ordering all the numbers.

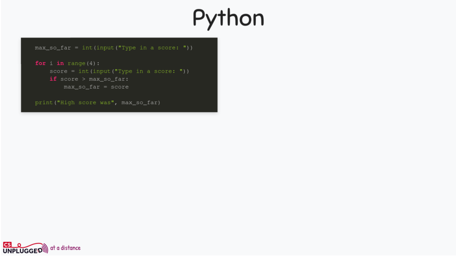


Scratch

```
when ⚑ clicked
ask (Type in a score) and wait
set [max so far ▾] to (answer)
repeat (4)
    ask (Type in a score) and wait
    if <(answer) > (max so far)> then
        set [max so far ▾] to (answer)
    ↲
say (join (High score was) (max so far)) for (10) seconds
```

Slide 13

Here is the same algorithm in Python.

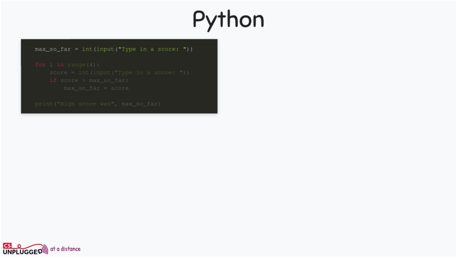


Python

```python
max_so_far = int(input("Type in a score: "))

for i in range(4):
    score = int(input("Type in a score: "))
    if score > max_so_far:
        max_so_far = score

print("High score was", max_so_far)
```

Slide 14

It stores the first number as a variable.

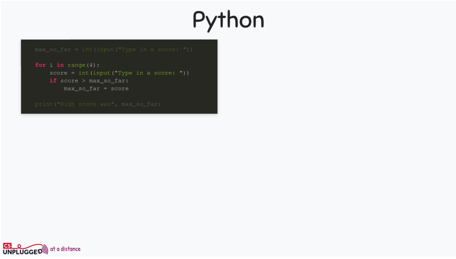


Python

```
max_so_far = int(input("Type in a score: "))

for i in range(4):
    score = int(input("Type in a score: "))
    if score > max_so_far:
        max_so_far = score

print("High score was", max_so_far)
```

Slide 15

Then compares each number to the highest so far.

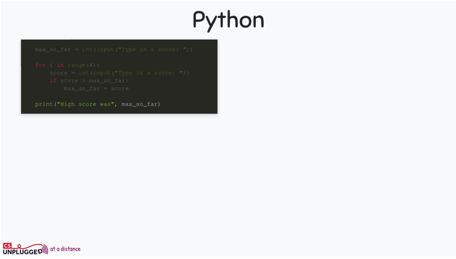


Python

```
max_so_far = int(input("Type in a score: "))

for i in range(4):
    score = int(input("Type in a score: "))
    if score > max_so_far:
        max_so_far = score

print("High score was", max_so_far)
```

Slide 16

Then it tells the user the highest number.



Python

```
max_so_far = int(input("Type in a score: "))

for i in range(4):
    score = int(input("Type in a score: "))
    if score > max_so_far:
        max_so_far = score

print("High score was", max_so_far)
```

Slide 17

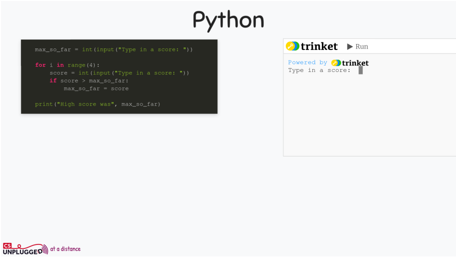
Click the ▶ Run button at the top of the interactive if the "Type in a score:" prompt doesn't show.

Now lets try it out.

It's the same algorithm in a different language. Remember, even when using the same language different algorithms will work.

Click within the interactive box. Type in a score, say it out loud, and press the `Enter` key. Repeat this 4 more times. Point out that the program has displayed the highest score.

You can click the ▶ Run button to rerun the program if you wish.



Python

```
max_so_far = int(input("Type in a score: "))

for i in range(4):
    score = int(input("Type in a score: "))
    if score > max_so_far:
        max_so_far = score

print("High score was", max_so_far)
```

```
Powered by trinket
Type in a score:
```

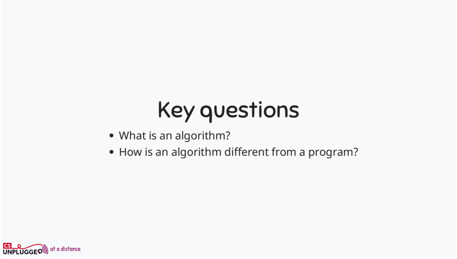
Slide 18

Back to the key questions.

What is an algorithm? An algorithm is a process for achieving an outcome that can be implemented on a computer.

How is an algorithm different from a program? A program will work on a computer, an algorithm is the idea for how it works.

In our example, the algorithm was the idea of comparing each value with the largest so far, and possibly updating it. The two programs both implemented this algorithm on a computer. From this activity we can see that it is often helpful for students to articulate the process (the algorithm) to solve a problem before they write it as a program.



Key questions

- What is an algorithm?
- How is an algorithm different from a program?

Slide 19

Here are some supporting resources for you. I'll paste these links in the chat.

```
Computer Science Field Guide on Algorithms
https://www.csfieldguide.org.nz/en/chapters/
algorithms/

High score interactive
https://www.csfieldguide.org.nz/en/interactives/high-
score-boxes/
```

Supporting Resources

- Computer Science Field Guide on Algorithms
  https://www.csfieldguide.org.nz/en/chapters/algorithms/
- High score interactive
  https://www.csfieldguide.org.nz/en/interactives/high-score-boxes/