POWERING POTENTIAL

L3 Data Essentials

- Module 4
- Statistics for Decision-Making
- 3 days Live Event learning

SESSION OVERVIEW

- Introductions and overview of the 3-day class-based learning.
- Re-cap Module 4 key concept digital learning content.
- Engage in practical activities to support module evidence collection.
- Provide support and guidance for the successful completion of Module 4.

THE KSB FRAMEWORK

Knowledge

K7 Algorithms and how they work using a step-by-step solution to a problem, or rules to follow to solve the problem and the potential to use automation.

K9 Basic statistical methods and simple data modelling to extract relevant data and normalise unstructured data.

K14 The significance of customer issues, problems, business value, brand awareness, cultural awareness/ diversity, accessibility, internal/external audience, level of technical knowledge, and profile in a business context.

SCHEDULE: DAY 1

AM

- Introduction to algorithms
- Activity: CupcakeBot
- Activity: Understanding algorithms
- Different types of algorithms
- Research activity: Types of algorithms

PM

- Activity: Implement an algorithm in Excel
- Review of statistic and algorithm practise activities
- Activity: Use of automated algorithms in business and everyday life
- Activity: Describe an organisational task as an algorithm

Day 1

WHAT IS AN ALGORITHM?

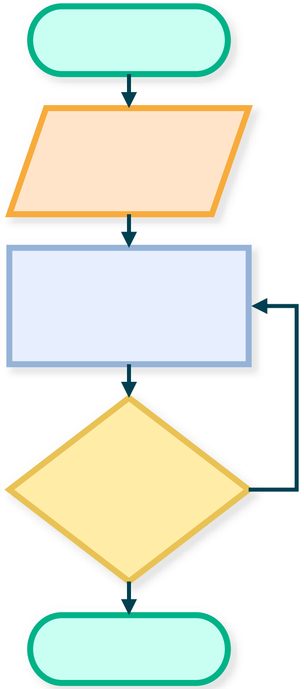
Simply put, an algorithm is a set of instructions or **rules** to be followed, **step-by-step**, to achieve a **recognisable result**.

Starting from defined inputs, the application of an algorithm will produce a required output, through a finite number of well-defined steps, with no space for discretionary interpretation, achieving the same results every time the algorithm is used.

WHAT IS AN ALGORITHM?

An algorithm is a procedure consisting of commands that must be carried out exactly and completely, with no human intervention, besides the algorithm creation.

The structure of an algorithm can always be represented by a **flowchart** in a **pseudo-code** environment, where the main inputs, actions, loops, and expected outcome are described and defined with no reference to any programming language.



ALGORITHMS VS. INSTRUCTIONS

Algorithms

- Exact, unbreakable commands.
- Typically given to a computer or machine.
- Not subject to interpretation or common sense.

Instructions

- Only ‘strong suggestions.’
- Typically given to an intelligent agent, such as a person.
- Subject to interpretation, adaptation, and common sense.

ALGORITHMS VS. INSTRUCTIONS

Algorithms are no better than instructions and instructions are no better than algorithms.

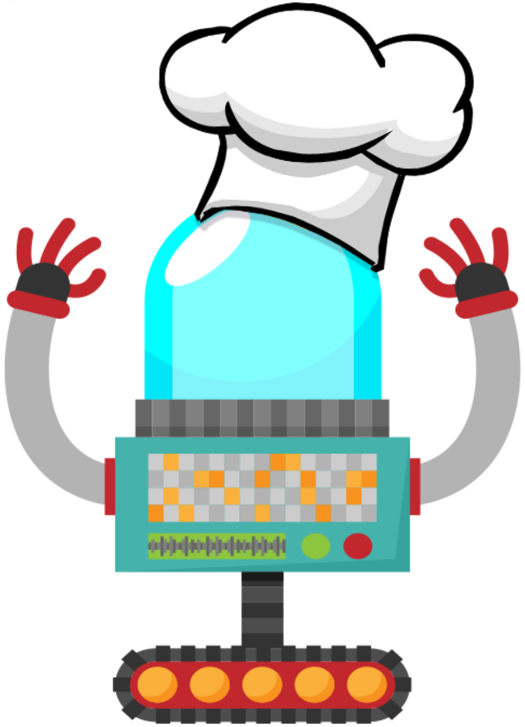
They just imply a different use of resources and tools to achieve a desired outcome and result.

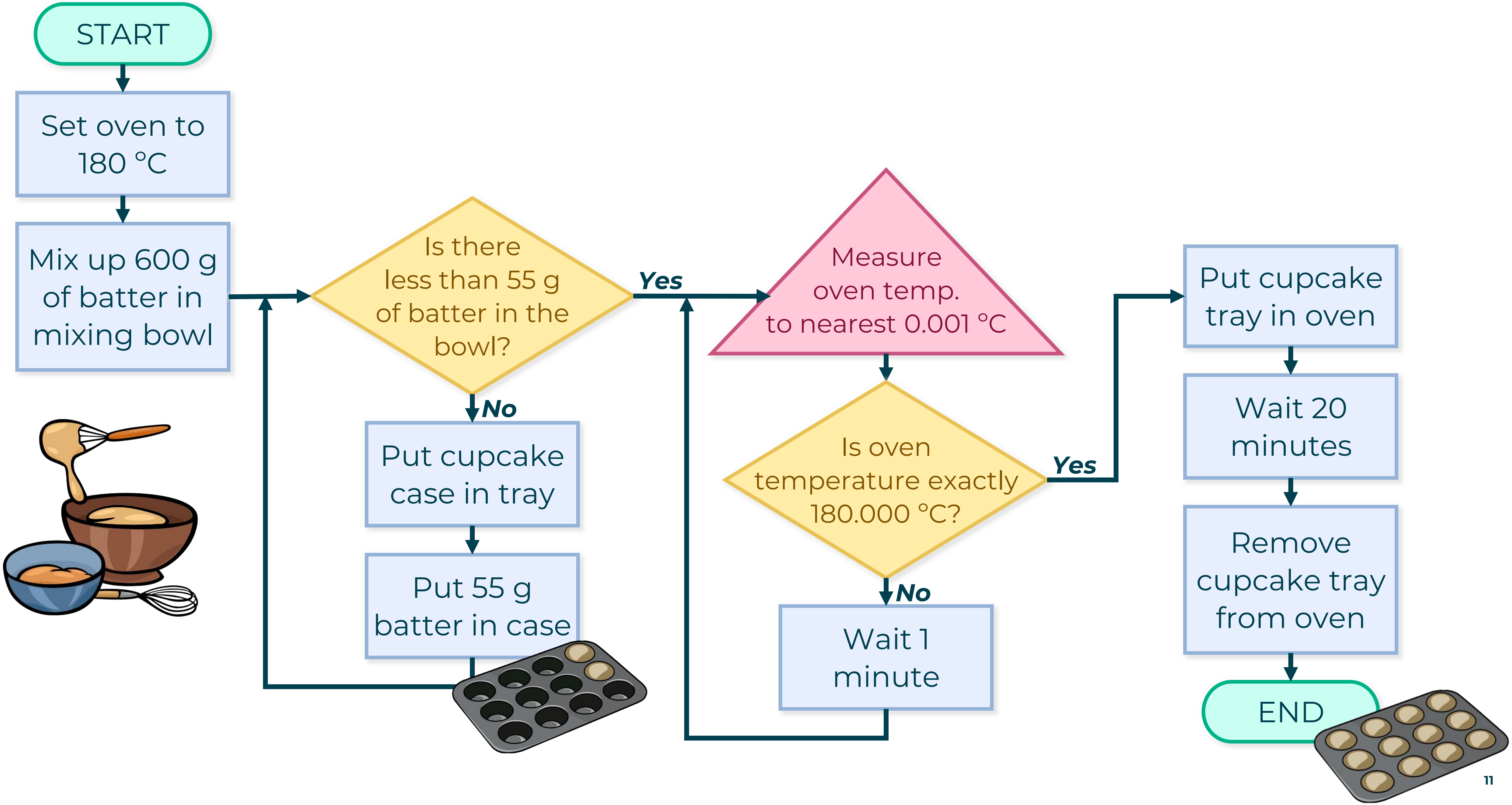
The man-machine interaction is continuously evolving leveraging on new technologies and tools that free up time and resources so to employ people in what are their irreplaceable added values

Algorithm example: CupcakeBot

CupcakeBot is an advanced cupcake-making robot.

In the quest to obtain perfect cupcakes every time, CupcakeBot's designers have programmed it with a **specially designed algorithm**. CupcakeBot must follow this algorithm exactly.





START

Set oven to 180 °C

Mix up 600 g of batter in mixing bowl

Is there less than 55 g of batter in the bowl?

No

Put cupcake case in tray

Put 55 g batter in case

Yes

Measure oven temp. to nearest 0.001 °C

Is oven temperature exactly 180.000 °C?

No

Wait 1 minute

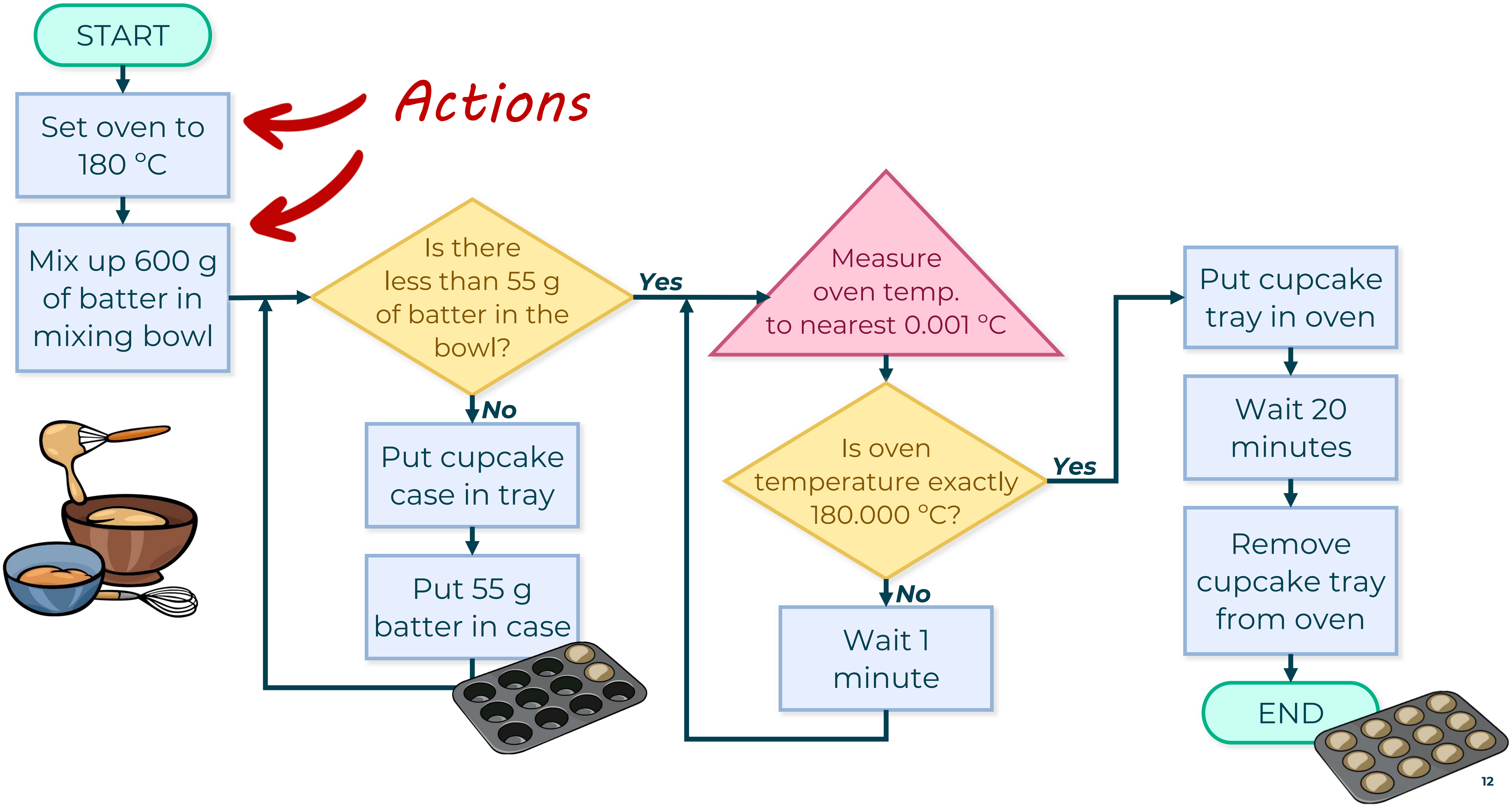
Yes

Put cupcake tray in oven

Wait 20 minutes

Remove cupcake tray from oven

END



START

Set oven to 180 °C

Mix up 600 g of batter in mixing bowl

Actions

Is there less than 55 g of batter in the bowl?

No

Put cupcake case in tray

Put 55 g batter in case

Yes

Measure oven temp. to nearest 0.001 °C

Is oven temperature exactly 180.000 °C?

No

Wait 1 minute

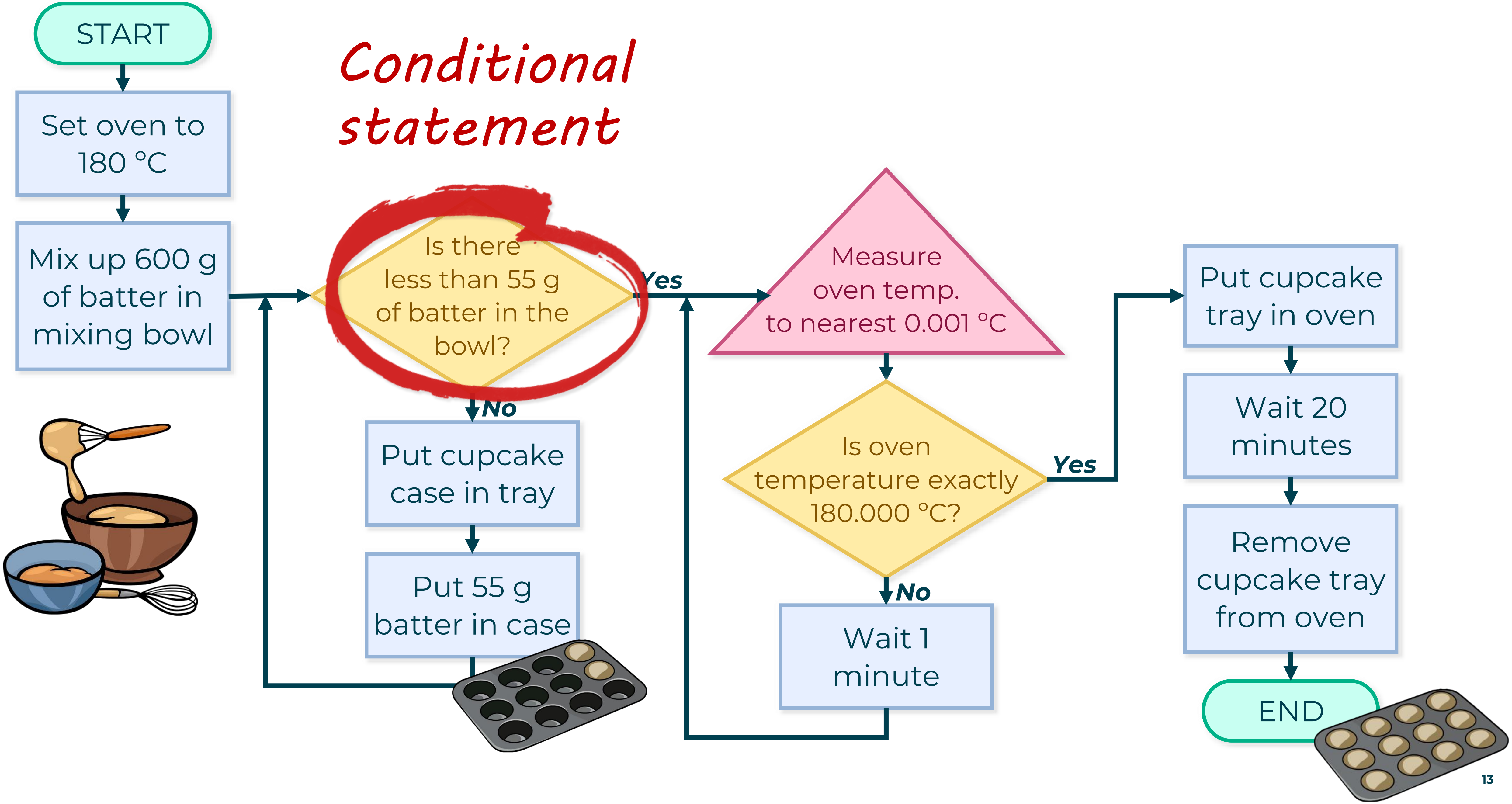
Yes

Put cupcake tray in oven

Wait 20 minutes

Remove cupcake tray from oven

END



Conditional statement

START

Set oven to 180 °C

Mix up 600 g of batter in mixing bowl

Is there less than 55 g of batter in the bowl?

Yes

No

Put cupcake case in tray

Put 55 g batter in case

Measure oven temp. to nearest 0.001 °C

Is oven temperature exactly 180.000 °C?

Yes

No

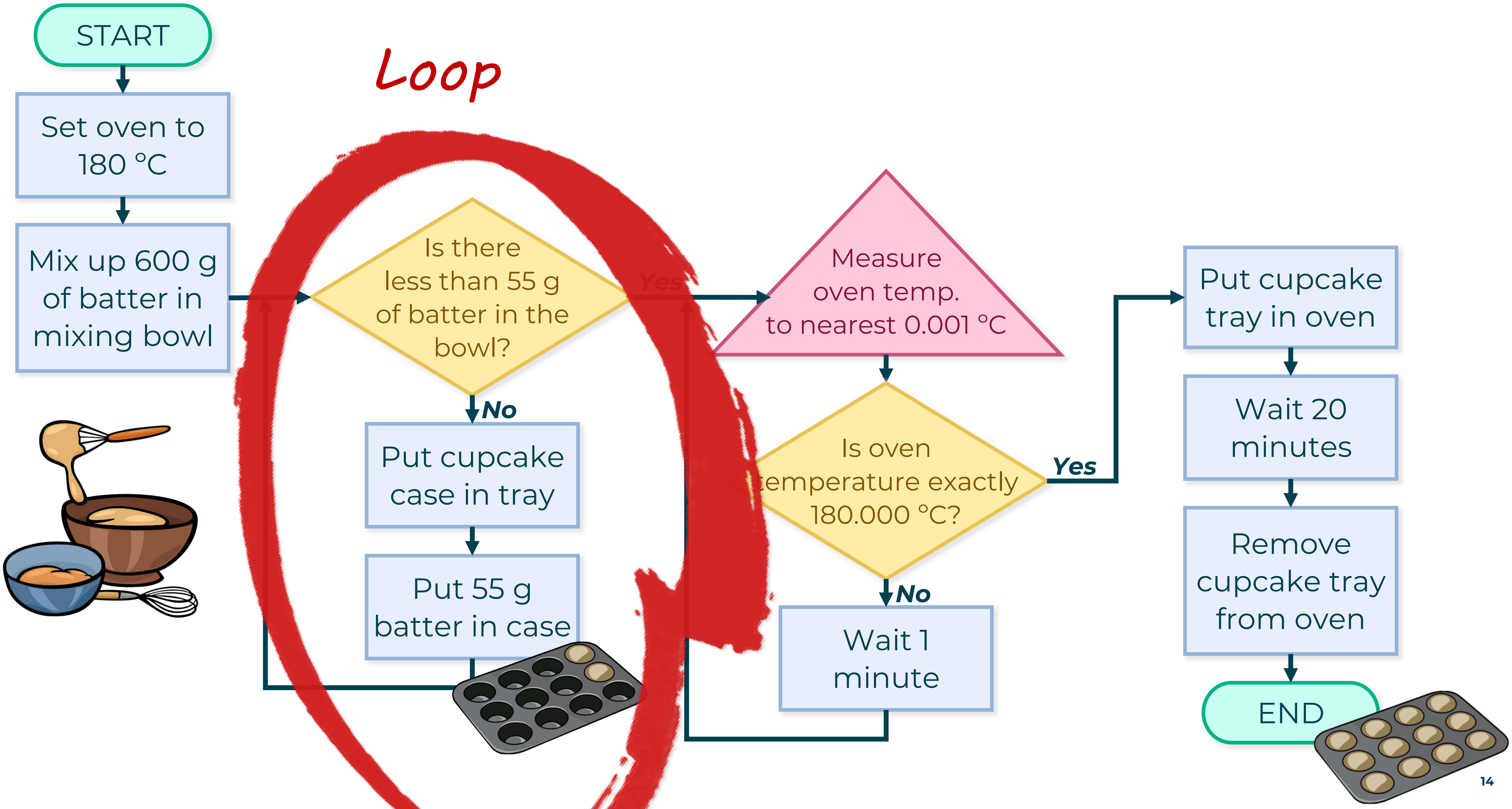
Wait 1 minute

Put cupcake tray in oven

Wait 20 minutes

Remove cupcake tray from oven

END



Loop

START

Set oven to 180 °C

Mix up 600 g of batter in mixing bowl

Is there less than 55 g of batter in the bowl?

No

Put cupcake case in tray

Put 55 g batter in case

Yes

Measure oven temp. to nearest 0.001 °C

Is oven temperature exactly 180.000 °C?

No

Wait 1 minute

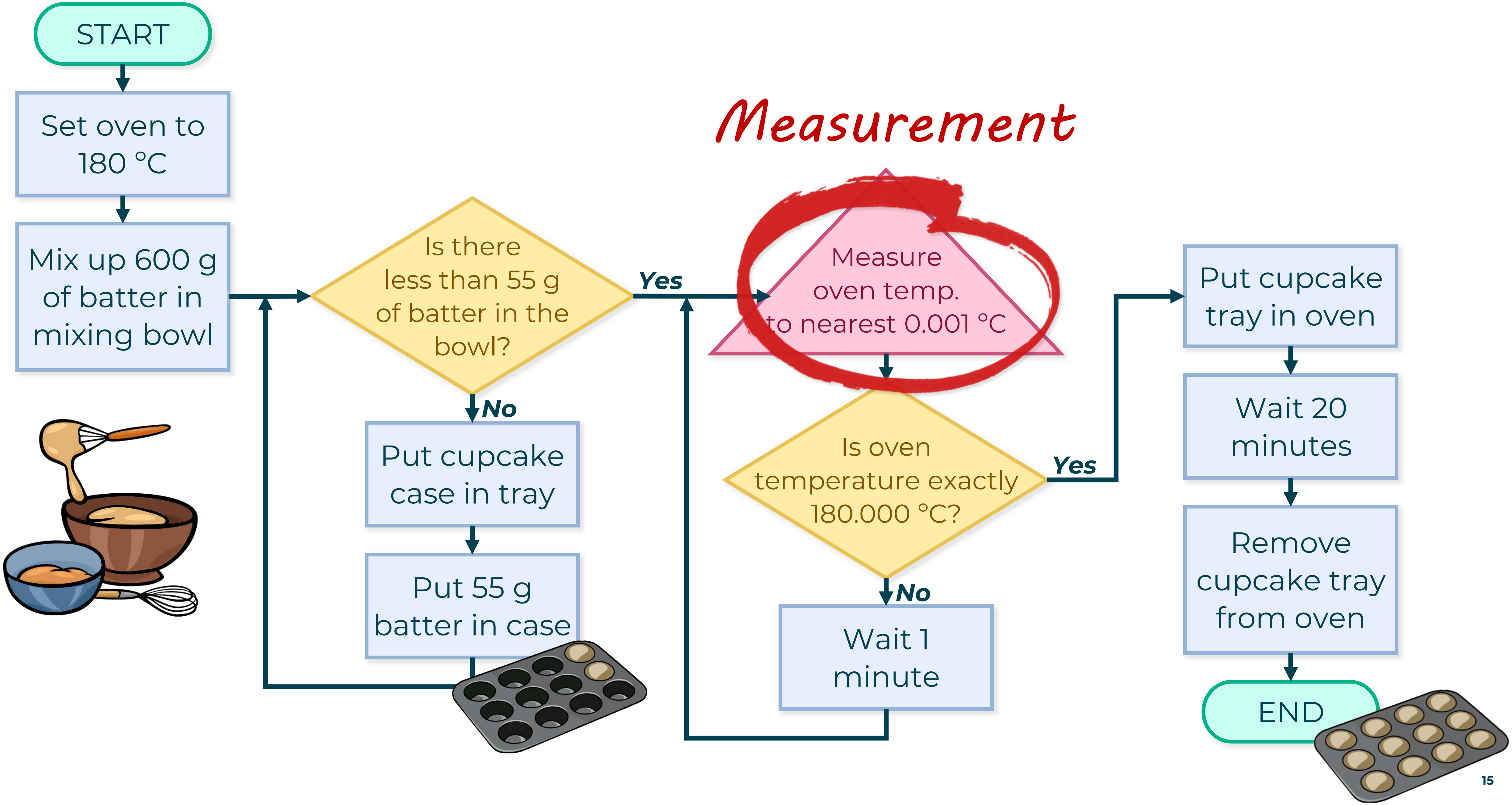
Yes

Put cupcake tray in oven

Wait 20 minutes

Remove cupcake tray from oven

END



Measurement

START

Set oven to 180 °C

Mix up 600 g of batter in mixing bowl

Is there less than 55 g of batter in the bowl?

No

Put cupcake case in tray

Put 55 g batter in case

Yes

Measure oven temp. to nearest 0.001 °C

Is oven temperature exactly 180.000 °C?

No

Wait 1 minute

Yes

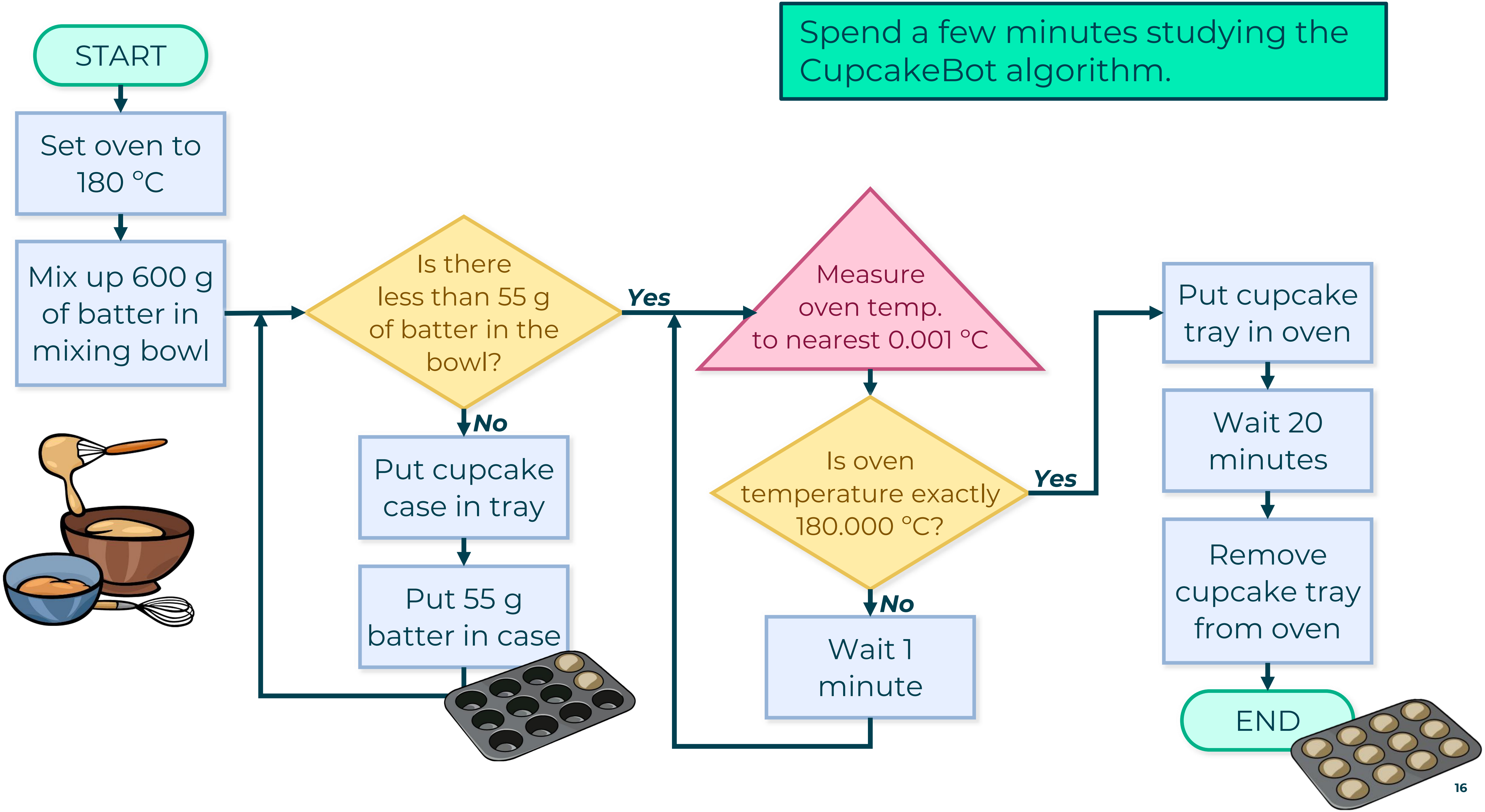
Put cupcake tray in oven

Wait 20 minutes

Remove cupcake tray from oven

END

Spend a few minutes studying the CupcakeBot algorithm.



START

Set oven to 180 °C

Mix up 600 g of batter in mixing bowl

Is there less than 55 g of batter in the bowl?

Yes

No

Put cupcake case in tray

Put 55 g batter in case

Measure oven temp. to nearest 0.001 °C

Is oven temperature exactly 180.000 °C?

Yes

No

Wait 1 minute

Put cupcake tray in oven

Wait 20 minutes

Remove cupcake tray from oven

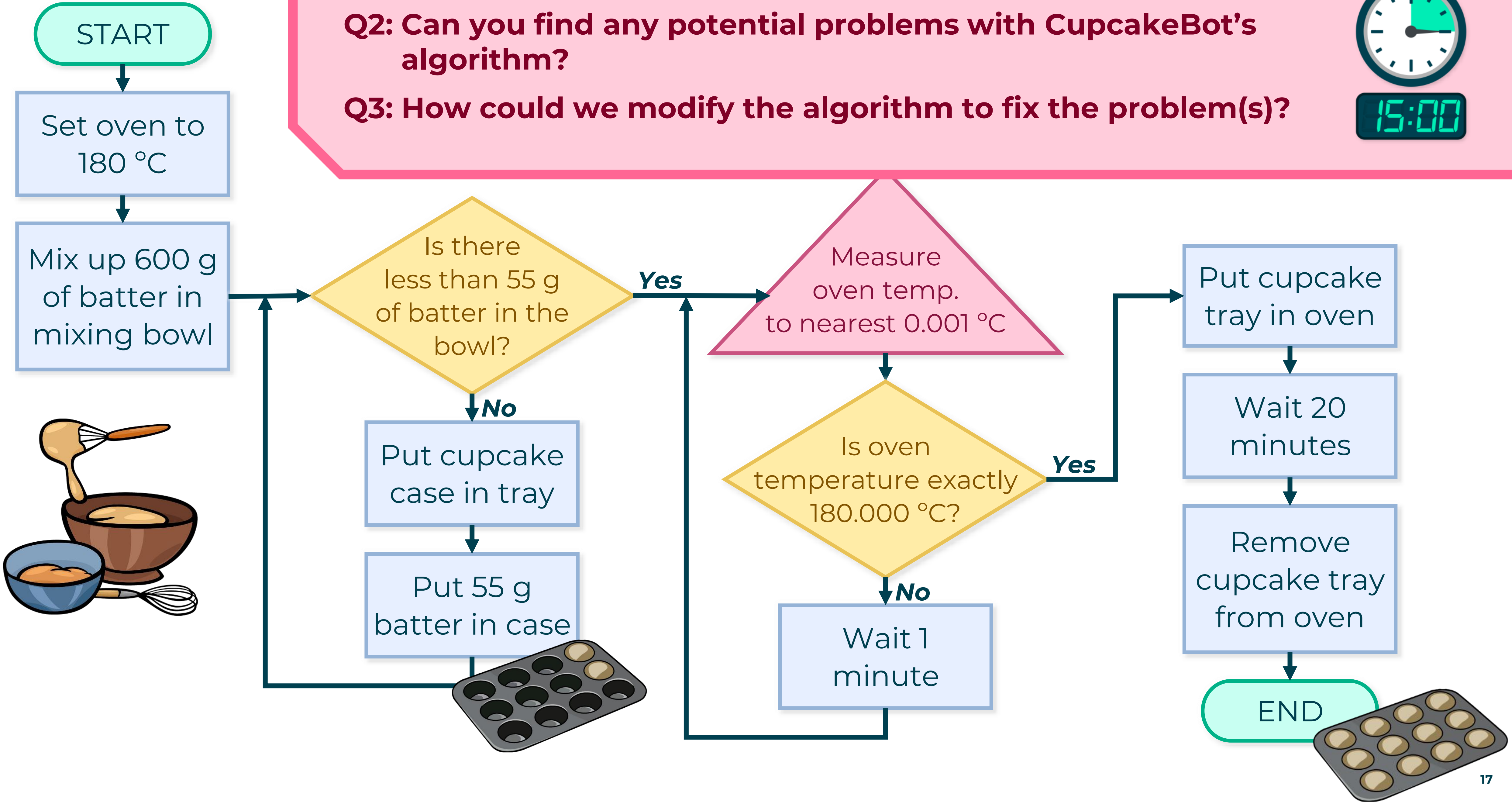
END

Q1: How many cupcakes will be baked?

Q2: Can you find any potential problems with CupcakeBot's algorithm?

Q3: How could we modify the algorithm to fix the problem(s)?

15:00



START

Set oven to 180 °C

Mix up 600 g of batter in mixing bowl

Is there less than 55 g of batter in the bowl?

No

Put cupcake case in tray

Put 55 g batter in case

Yes

Measure oven temp. to nearest 0.001 °C

Is oven temperature exactly 180.000 °C?

No

Wait 1 minute

Yes

Put cupcake tray in oven

Wait 20 minutes

Remove cupcake tray from oven

END

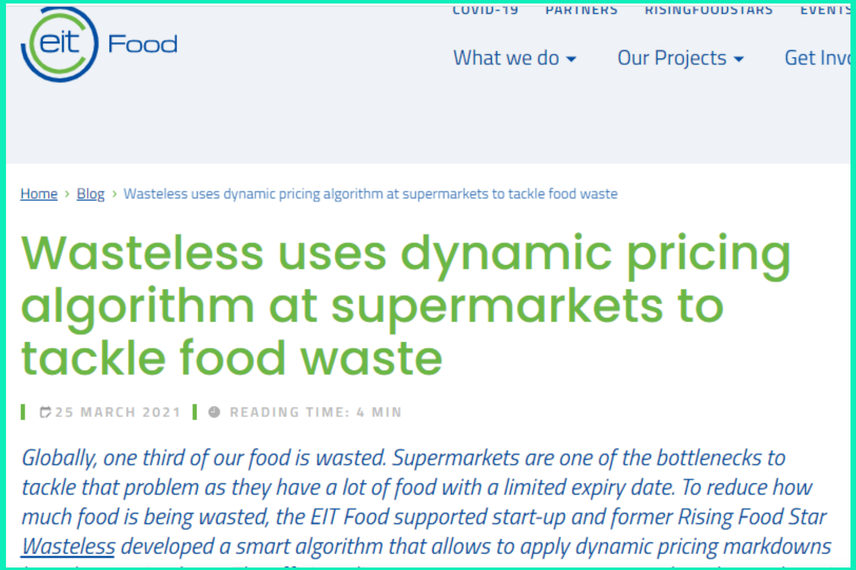
CUPCAKEBOT

Q1: How many cupcakes will be baked?

Q2: Can you find any potential problems with CupcakeBot’s algorithm?

Q3: How could we modify the algorithm to fix the problem(s)?

Activity: Understanding algorithms



Home › Blog › Wasteless uses dynamic pricing algorithm at supermarkets to tackle food waste

Wasteless uses dynamic pricing algorithm at supermarkets to tackle food waste

25 MARCH 2021 | READING TIME: 4 MIN

Globally, one third of our food is wasted. Supermarkets are one of the bottlenecks to tackle that problem as they have a lot of food with a limited expiry date. To reduce how much food is being wasted, the EIT Food supported start-up and former Rising Food Star Wasteless developed a smart algorithm that allows to apply dynamic pricing markdowns

https://www.eitfood.eu/blog/post/start-up-wasteless-tackles-food-waste-at-supermarkets-with-dynamic-pricing-algorithm

- Read through the online article in breakout sessions.
- Identify the key points in the article in terms of how algorithms are used to tackle food waste.
- Discuss your findings in break-out sessions and feedback to the group.
- Time allocated: 30 minutes.

Types of Algorithms

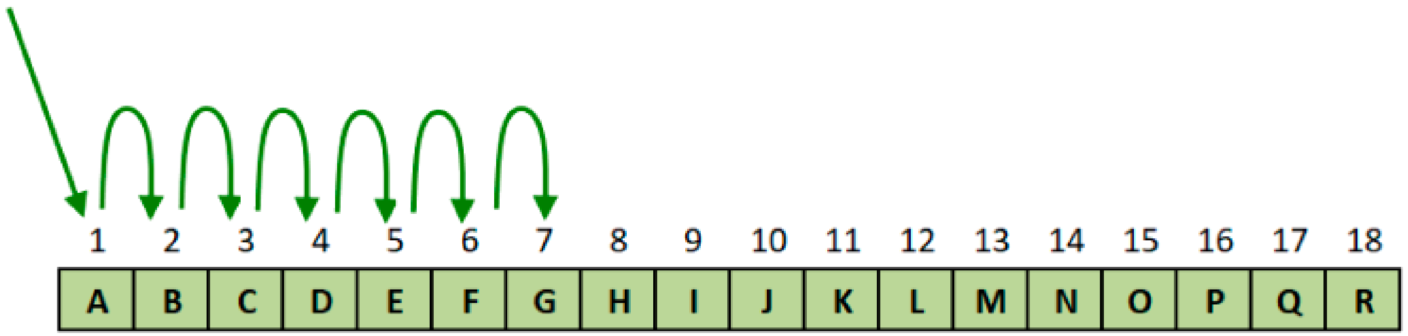
There are many types of algorithms that are used in different business environments.

Sorting and **Searching** are two of the most common algorithms used in program design.

Sort and Search algorithms

Algorithm yype	Description
Linear Search	This algorithm looks through a list of items from start to finish to find the search item.
Binary Search	Binary Search algorithms cut an ordered list in half before checking in which half the search item does not appear. The algorithm then splits the other half in to two, and the repeats until all you have left is the search item or an empty list.
Bubble Sort	Bubble Sort is a simple algorithm that looks at the first two items in a list to see which is greater and swaps them if they are not in the correct order. It then looks at the second and third items and does the same. The algorithm then runs through the whole list multiple times until every item is correctly ordered.
Merge Sort	This kind of sorting algorithm splits the list into multiple lists containing only one element. These are then merged and sorted back into one list.

LINEAR SEARCH

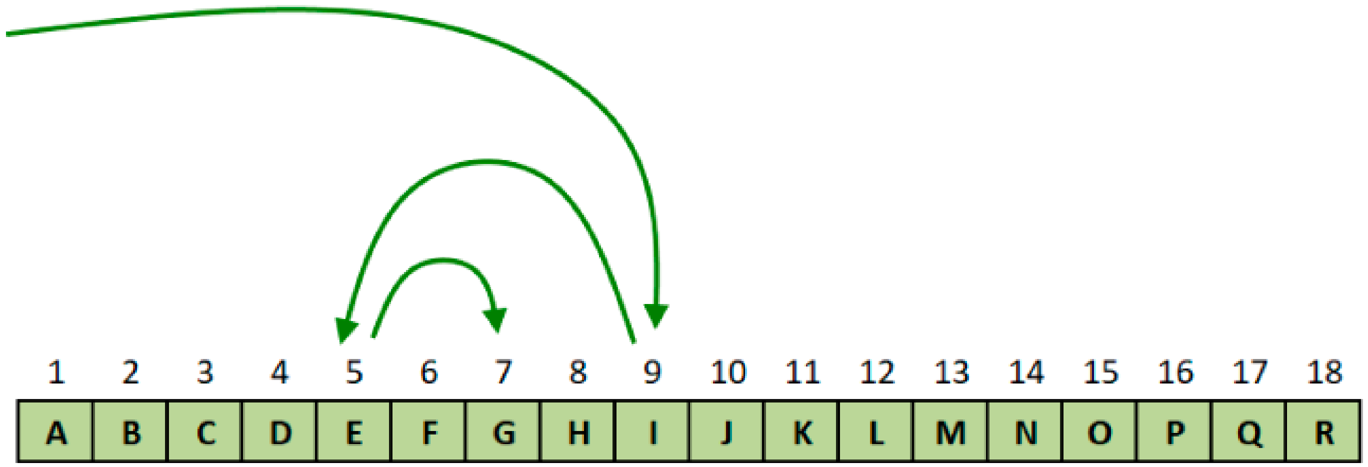


1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R

Linear Search - Find 'G' in sorted list A-R

Source: tutorialhorizon.com

BINARY SEARCH

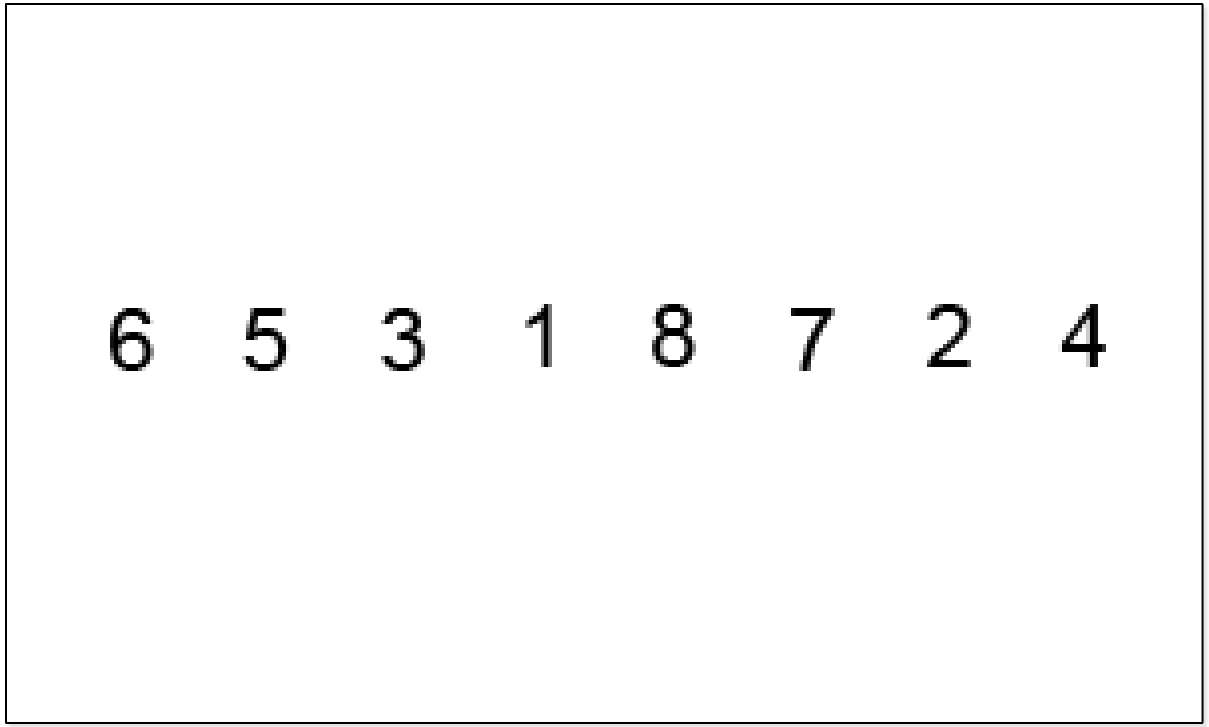


1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R

Binary Search - Find 'G' in sorted list A-R

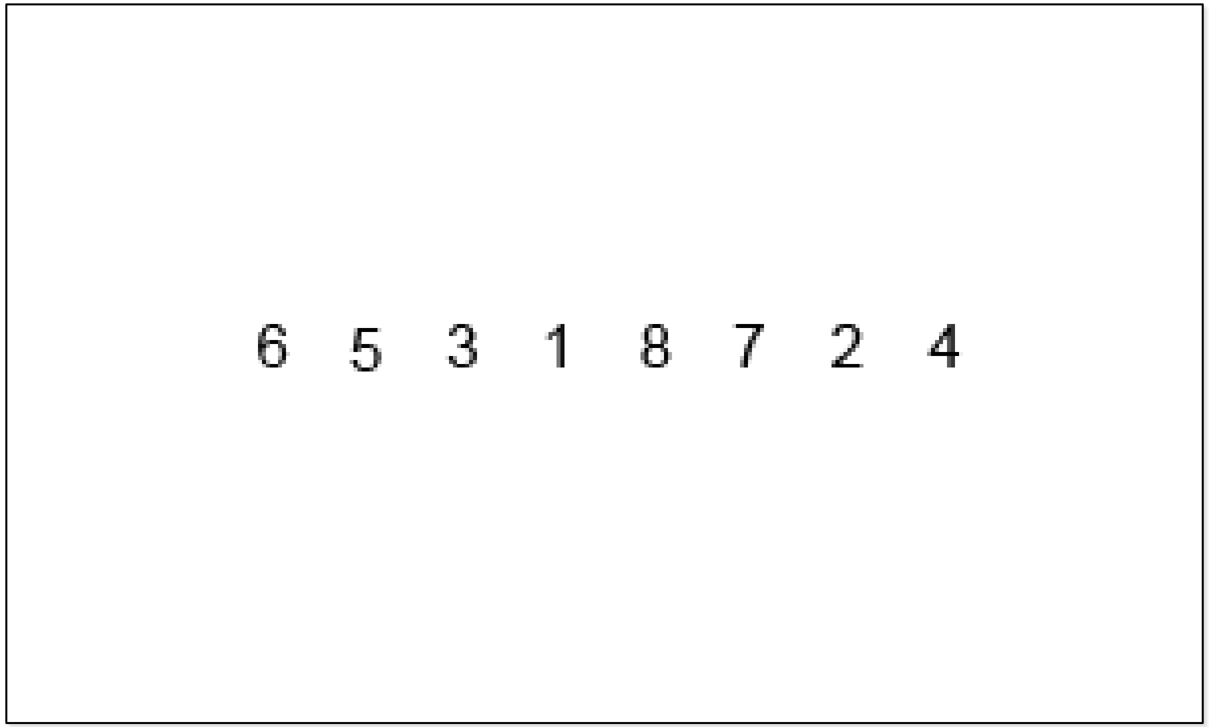
Source: tutorialhorizon.com

BUBBLE SORT



Source: Wikipedia

MERGE SORT



Source: Wikipedia

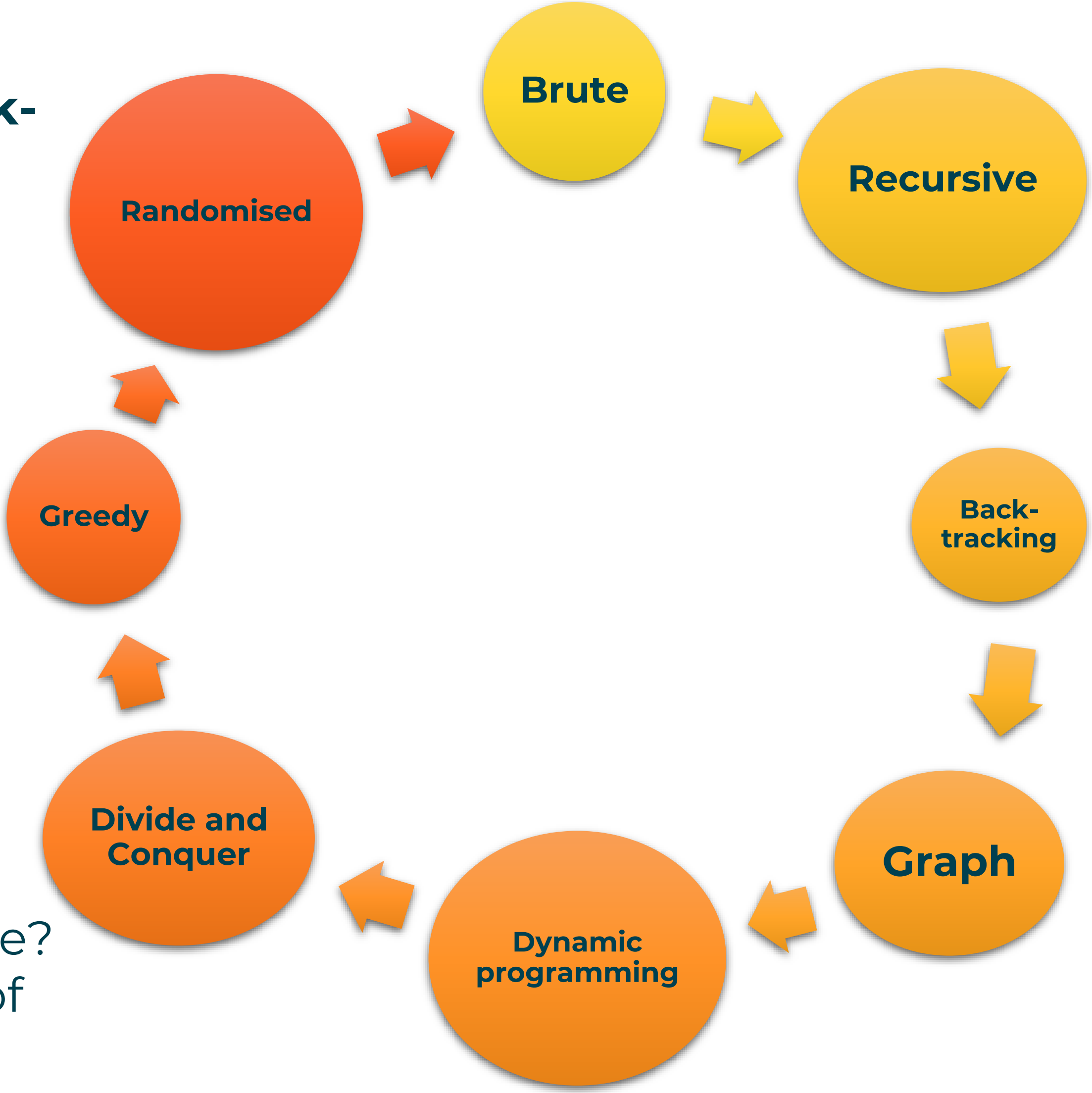
Types of algorithms

In your online learning content for Module 4, you were introduced to different types of algorithms: **Greedy, Back-tracking, Brute, Randomised, Dynamic Programming, Recursive, Divide and Conquer, Graph**

These algorithm types are not all mutually exclusive, and some are subsets of others. For example, some graph algorithms are greedy, and all divide-and-conquer algorithms are recursive.

Task

1. The class is split in break-out sessions to review and analyse the different algorithms, and report back to class a brief description of the same.
2. Discuss as a class what are the benefits of each algorithm and the main differences between the same? Each group will also have to provide some examples of practical use of those algorithms, particularly and ideally within their working organisations.



GREEDY ALGORITHMS

- Greedy algorithms select the option that maximises an immediate gain, without considering what could be the optimal long-term solution.

- A greedy algorithm is an approach to solving problems by making the locally optimal choice at each step, with the possibility, but not the certainty, of also finding a global optimum solution.

- A simple but intuitive algorithm that is used in optimisation problems to provide a quick, although potentially sub-optimal, solution.

- The greedy algorithm makes the optimal choice at each step as it attempts to find the overall optimal way to solve the entire problem.

Randomised algorithms

- Make random choices or select randomly generated numbers to decide what to do next within their logic.

Brute algorithms

- Brute algorithms are among the most common types of algorithms.
- A solution is devised by exploring and trying all possible different solutions.
- Simple but poorly efficient. It guarantees to find a solution if exists.

GRAPH ALGORITHMS

- Graph algorithms operate on a network of connected data (nodes) and are used to explore the relationships existing between data and the complexity of their interaction.
- A graph consists of a finite set of vertices or nodes and a set of edges connecting these vertices.
- These algorithms are based on the attempt to understand the relationships between data and are useful to visualise existing relationships between agents operating within a complex system
- Every time there is a sample of objects, where the behaviour of a single object influences and is influenced by the behaviour of the other objects, a graph algorithm might be an appropriate way to describe the dynamics describing such sample.
- Graphs have become a powerful tool of modelling data in real-world scenarios such as social media networks, web pages and links, locations and routes in GPS, natural sciences, and increasingly also financial markets.

DYNAMIC PROGRAMMING ALGORITHM

- Find a solution to the complex problem based on the solutions found for more simple sub-problems.
- Every sub-problem is solved only once.
- Dynamic programming algorithms are advanced algorithms where past choices and output are remembered and applied to future situations and determine the next output.

RECURSIVE ALGORITHMS

- They work based on reduction.

- If the instance of a problem cannot be solved directly, you can reduce it to simpler instances of the same, which will be then combined to address the main initial problem.

- An example of using a recursive algorithm could be found to calculate the sum of an array of numbers. In business, this could relate to calculating sales figures over a set period across different product ranges.

DIVIDE AND CONQUER ALGORITHM

- This algorithm separates the main problem into sub-problems, solving them individually.

- The solutions to the sub-problems are then combined to define the solution to the entire problem.

- Dynamic Programming algorithms and Recursive algorithms are both forms of Divide and Conquer.

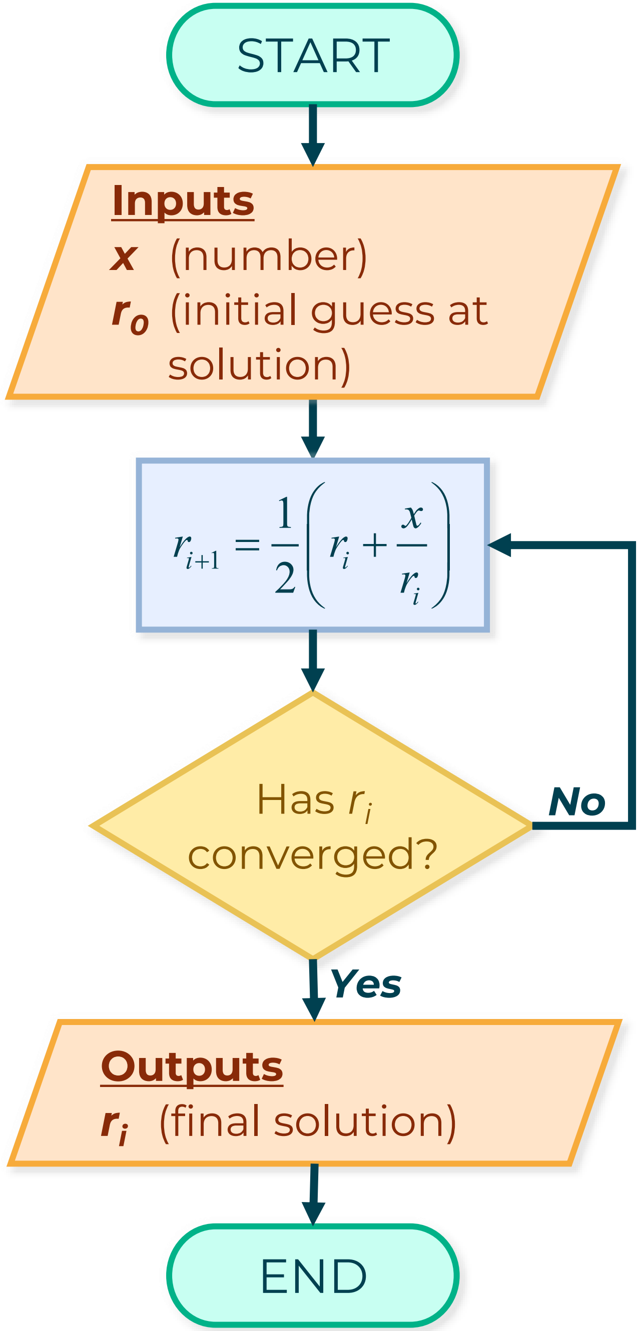
BACK TRACKING

- They are a modified form of Brute algorithms and work on a backtrack basis.

- Any time the algorithm is in a position where is not able to advance in finding a solution, it goes back one or more steps, revisiting the previous decisions taken, and trying a possible different path.

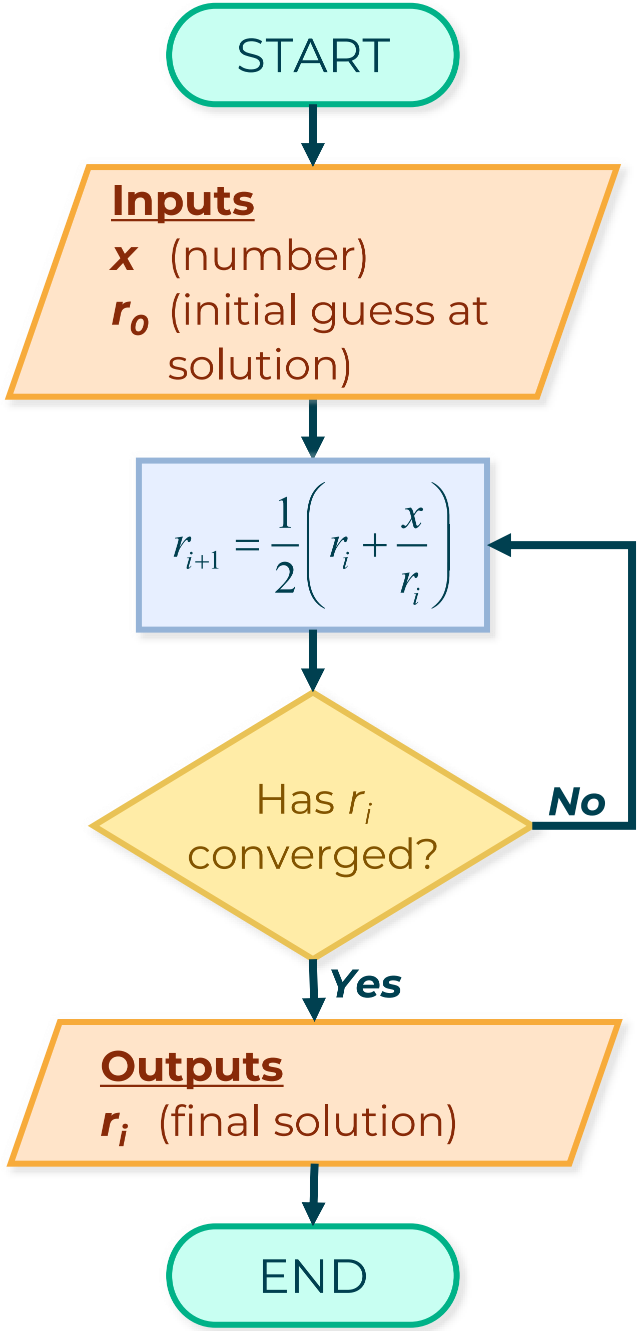
- A typical example of brute algorithm relates to finding your way out from a maze. Anytime you reach a dead end, you are forced to go back to the previous crossroads, and explore alternative paths to find the exit, until you eventually succeed.

EXAMPLE ALGORITHM


START
Inputs
x (number)
$\boldsymbol{r_0}$ (initial guess at solution)
$r_{i+1} = \frac{1}{2}\left(r_i + \frac{x}{r_i}\right)$
Has r_i converged?
No
Yes
Outputs
$\boldsymbol{r_i}$ (final solution)
END

Allocated activity time: 30 minutes

EXAMPLE ALGORITHM



START

Inputs

x (number)

***r*₀** (initial guess at solution)

$$r_{i+1} = \frac{1}{2}\left(r_i + \frac{x}{r_i}\right)$$

Has r_i converged?

No

Yes

Outputs

***r*ᵢ** (final solution)

END

Activity

Implement this algorithm in Excel.

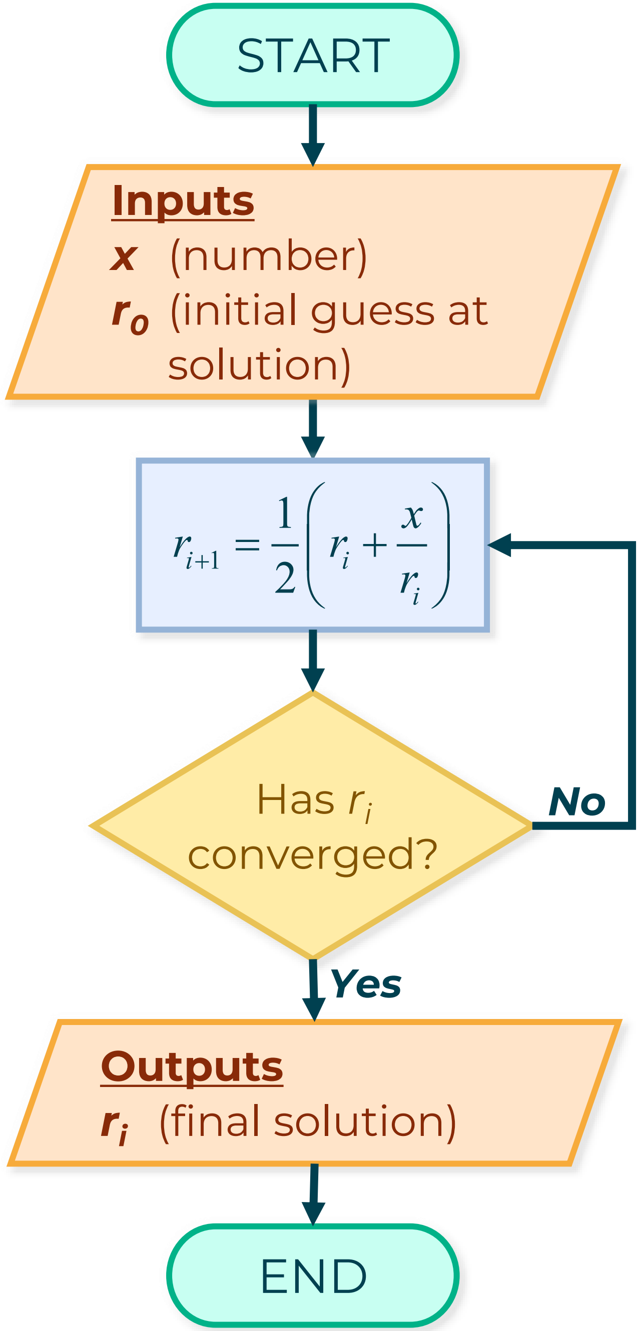
1. Write a formula to perform the blue box calculation.
2. Copy the formula down the rows to implement the loop.

Start with ***x*** **= 100**.

Experiment with inputs *x* and r_0.

45:00

EXAMPLE ALGORITHM


START
Inputs
x (number)
r_0 (initial guess at solution)
$r_{i+1} = \frac{1}{2}\left(r_i + \frac{x}{r_i}\right)$
Has r_i converged?
No
Yes
Outputs
r_i (final solution)
END

Question 1

What does this algorithm do?

Question 2

Does the initial guess input, r_0, have any effect on the final solution?

Question 3

How many iterations are required for **complete** convergence?

Question 4

Are there any values of r_0 or x that will cause the algorithm to produce an error or fail to converge?

Statistics and algorithms practice activities - Review

As part of your on-line learning, you were introduced to some statistics and algorithms practice activities.

Activity 1 objective

- To use Excel to show how basic statistical techniques are performed as algorithms, and to show how automating algorithms can speed up workflow by drawing parallels between Excel Formulas and subroutines in programming languages.

Activity 2 objective

- Use statistical methods in Excel within a real case scenario to analyse a dataset and draw conclusions from the same.

ACTIVITY REVIEW

Use of automated algorithms in business and everyday life

Task

- In breakout sessions, spend 10 minutes refreshing your memory with the tasks that were completed in the two activities.
- As a follow up on Activity 1, describe the formula of average and standard deviation as an algorithm (in plain English using bullet points, ideally though a flow chart).
- As a follow up on Activity 2, describe as an algorithm, the conditional formatting rule which identifies the values above the average in the dataset. In the description, include the pseudo code developed as Activity's 1 follow up (i.e., formula of average) as a defined function.
- Prepare to feedback your findings to the class.
- Activity time: up to 45 minutes.

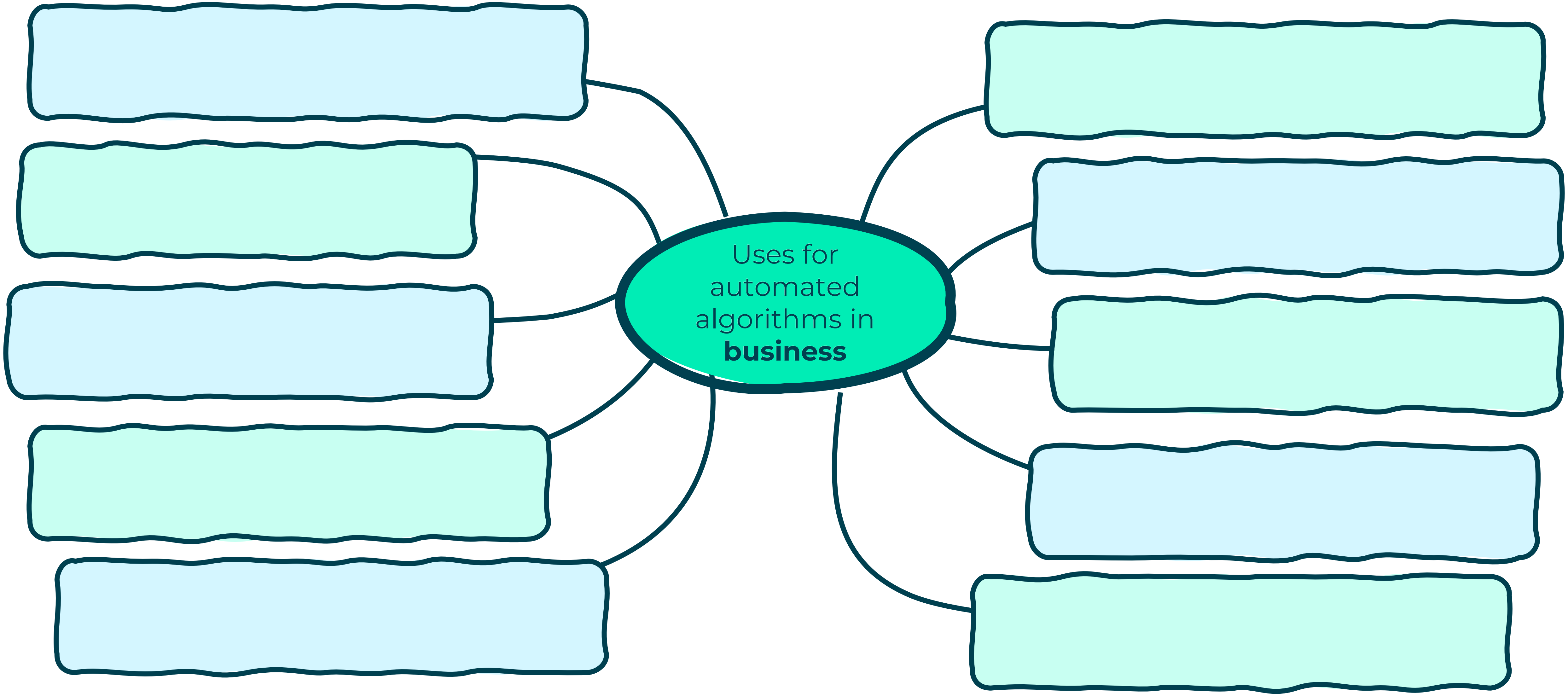
ACTIVITY DESCRIPTION

Use of automated algorithms in business and everyday life

Task

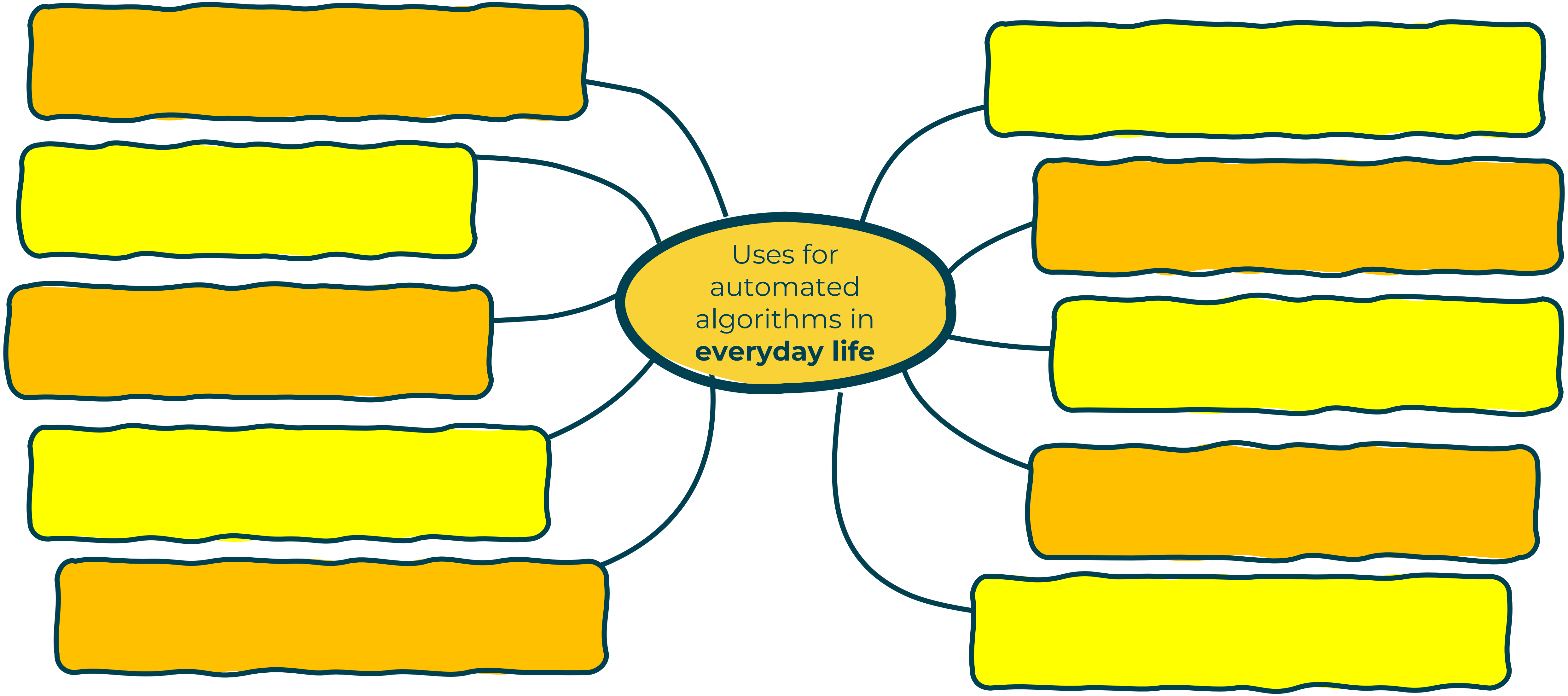
- In breakout sessions, discuss the ways in which automated algorithms are used in business and everyday life (for example, to improve the efficiency on solving tasks and organise processes).
- Discuss the advantages and disadvantages of using automated algorithms and produce a table of evidence.
- Identify how algorithms and automated algorithms are used, or could be used, in your work environment and organisation.
- Prepare to feedback your findings to the class.
- Activity time: 40 minutes.

TASK: ‘In breakout sessions, discuss ways in which **automated algorithms** are used in **business**. If possible, try and relate the discussion and examples to your current organisations.’



Uses for automated algorithms in **business**

TASK: 'In breakout sessions, discuss ways in which **automated algorithms** are used in **everyday life**.'



Uses for automated algorithms in **everyday life**

AUTOMATED ALGORITHMS

Pros

Cons

FINAL STRETCH GOAL ACTIVITY

Describe an organisational task within your organisation as an algorithm using a flow chart

Task

Following up on the discussion you had in the breakout sessions and in class about the importance of automated algorithms in business, try and describe an organisational task or process, within your organisation, as an algorithm using a flow chart.

Talk about tools that would help automating this process, ultimately leading to the creation of a fully automated algorithm in implementing the same.

You might draw the flowchart using visio.com, diagrams.net, or just pen and paper, scanning the drawing afterward.

Allocated time: 30 minutes.