Co-funded by the
Erasmus+ Programme
of the European Union

Trainers Support

Algorithmic Thinking

Algorithmic Thinking for Migrants Teachers Education

2021-1-EL01-KA210-ADU-000035033

LESSON #2

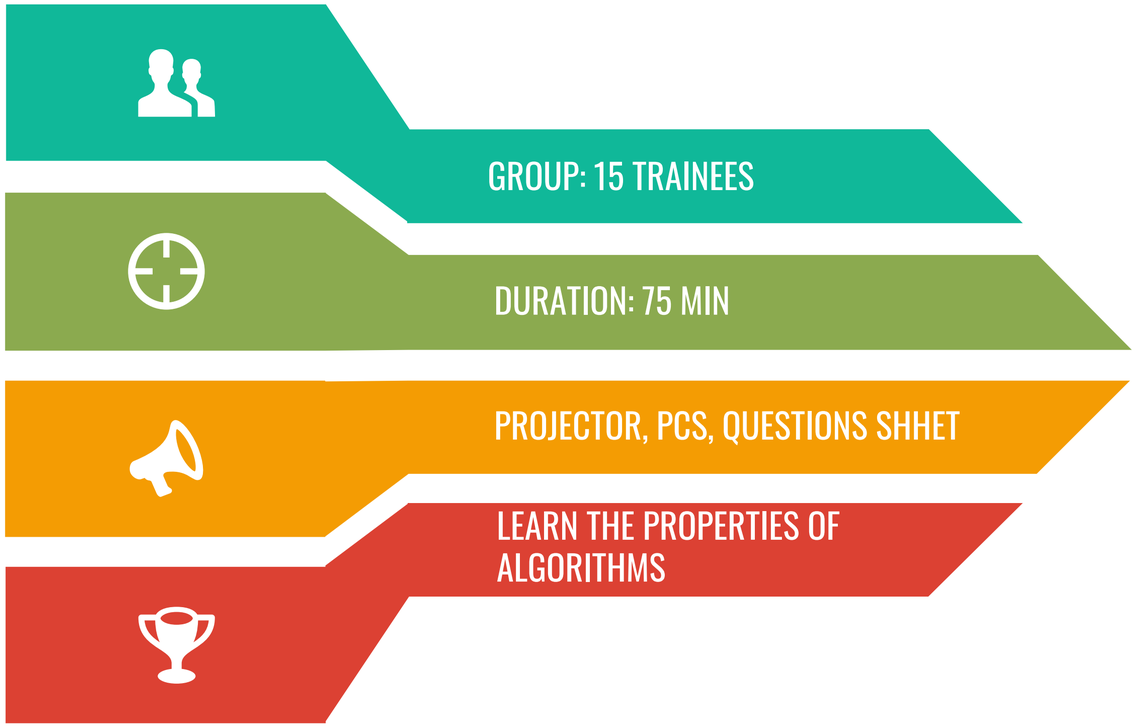
TITLE: ALGORITHMIC THINKING

Co-funded by the
Erasmus+ Programme
of the European Union

LESSON #2 - ALGORITHMIC THINKING

Erasmus+
Ανοίγει δρόμους, αλλάζει ζωές.

LESSON REQUIREMENTS



ΙΔΡΥΜΑ ΚΡΑΤΙΚΩΝ ΥΠΟΤΡΟΦΙΩΝ
ΙΚΥ
Co-funded by the Erasmus+ Programme of the European Union

LESSON #2 - ALGORITHMIC THINKING

WHAT IS AN ALGORITHM???

An algorithm is a set of **step-by-step procedures**, or a set of rules to follow, for completing a specific task or solving a particular problem. The word algorithm was first coined in the 9th century. Algorithms are all around us. Common examples include: the recipe for baking a cake, the method we use to solve a long division problem and the process of doing laundry.

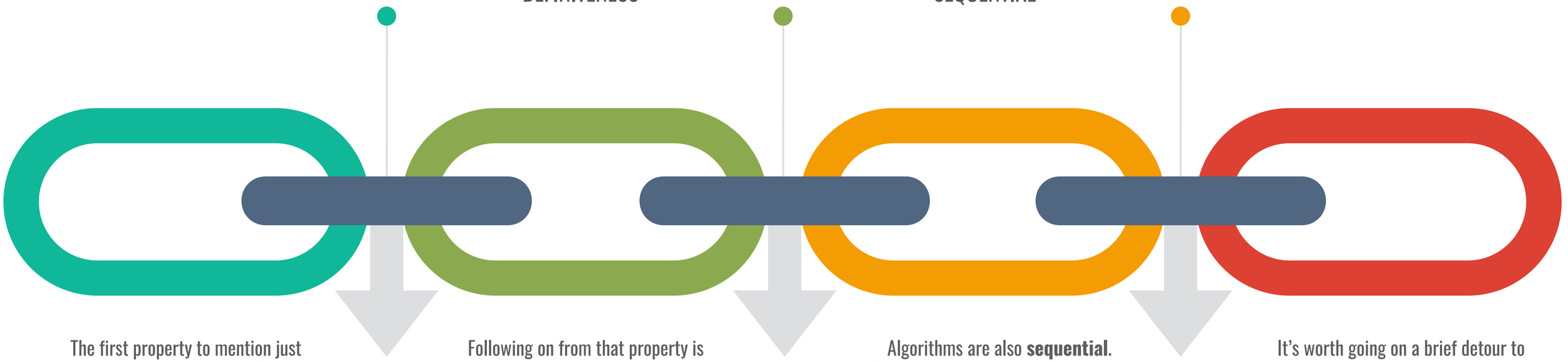
Here's what baking a cake might look like, written out as a list of instructions, just like an algorithm:

1. Preheat the oven
2. Gather the ingredients
3. Measure out the ingredients
4. Mix together the ingredients to make the batter
5. Grease a pan
6. Pour the batter into the pan
7. Put the pan in the oven
8. Set a timer
9. When the timer goes off, take the pan out of the oven

LESSON #2 - ALGORITHMIC THINKING

ALGORITHM PROPERTIES

COLLECTION OF INDIVIDUAL STEPS



The first property to mention just restates something said earlier: an algorithm is a **collection of individual steps**. A recipe fits this analogy quite simply, filled as it is with steps like: 'pre-heat the oven to 180 degrees Celsius' or 'add two tablespoons of sugar to the bowl'.

DEFINITENESS

Following on from that property is **definiteness**, meaning that every step must be precisely defined. Each step in an algorithm can have one and only one meaning, otherwise it is ambiguous. Similarly, chefs never write things like *'some sugar'* or *'cook it for a while'*.

SEQUENTIAL

Algorithms are also **sequential**. The steps that make up the process must be carried out in the order specified. Failing to do this means that the result of executing the algorithm is likely incorrect. Like a recipe, we must respect the sequence when running through an algorithm for it to have any meaningful result.

DETOUR: STATE IN ALGORITHMS

It's worth going on a brief detour to examine why sequence is so important to algorithms. It's all to do with state, by which simply mean the current values of all the things the algorithm is keeping track of. Clearly sequencing the steps of an algorithm ensures the that state always changes in the same way whenever the algorithm is executed.

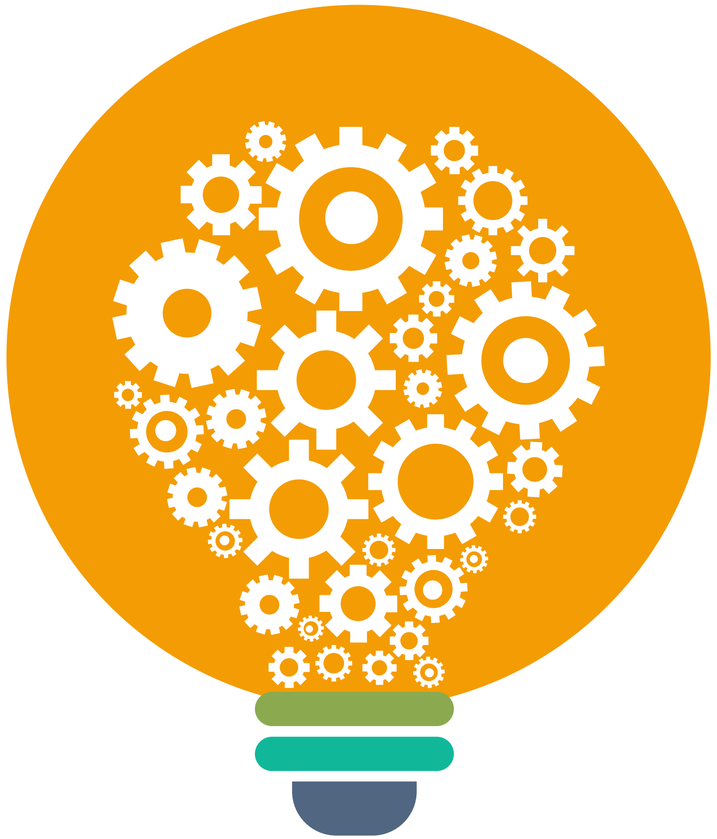
ΙΔΡΥΜΑ ΚΡΑΤΙΚΩΝ ΥΠΟΤΡΟΦΙΩΝ
ΙΚΥ
Co-funded by the Erasmus+ Programme of the European Union

LESSON #2 - ALGORITHMIC THINKING

TYPES OF ALGORITHMS

Algorithms are classified based on the concepts that they use to accomplish a task. While there are many types of algorithms, the most fundamental types are:

- **Divide and conquer algorithms** – divide the problem into smaller subproblems of the same type; solve those smaller problems, and combine those solutions to solve the original problem.
- **Brute force algorithms** – try all possible solutions until a satisfactory solution is found.
- **Randomized algorithms** – use a random number at least once during the computation to find a solution to the problem.
- **Greedy algorithms** – find an optimal solution at the local level with the intent of finding an optimal solution for the whole problem.
- **Recursive algorithms** – solve the lowest and simplest version of a problem to then solve increasingly larger versions of the problem until the solution to the original problem is found.
- **Backtracking algorithms** – divide the problem into subproblems, each which can be attempted to be solved; however, if the desired solution is not reached, move backwards in the problem until a path is found that moves it forward.



LESSON #2 - ALGORITHMIC THINKING

Erasmus+
Ανοίγει δρόμους, αλλάζει ζωές.

SORTING ALGORITHMS

A **sorting algorithm** is an algorithm that puts elements of a list in a certain order. Sorting is often an important first step in algorithms that solves more complex problems. There are a large number of sorting algorithms, each with their own benefits and costs. Besides, we will focus on some of the more famous sorting algorithms.

- **Linear sort:** Find the smallest element in the list to be sorted, add it to a new list, and remove it from the original list. Repeat this until the original list is empty.
- **Bubble sort:** Compare the first two elements in the list, and if the first is greater than the second, swap them. Repeat this with every pair of adjacent elements in the list. Then, repeat this process until the list is fully sorted.
- **Insertion sort:** Compare each element in the list to all the prior elements until a smaller element is found. Swap these two elements. Repeat this process until the list is fully sorted.

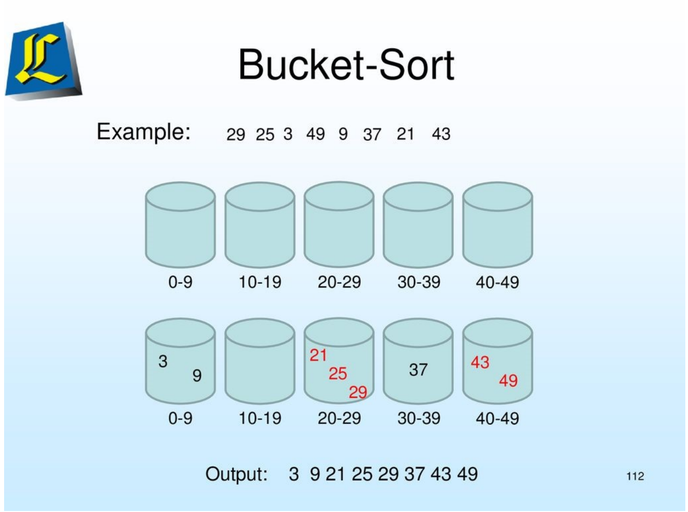
ΙΔΡΥΜΑ ΚΡΑΤΙΚΩΝ ΥΠΟΤΡΟΦΙΩΝ IKY
Co-funded by the Erasmus+ Programme of the European Union

LESSON #2 - ALGORITHMIC THINKING

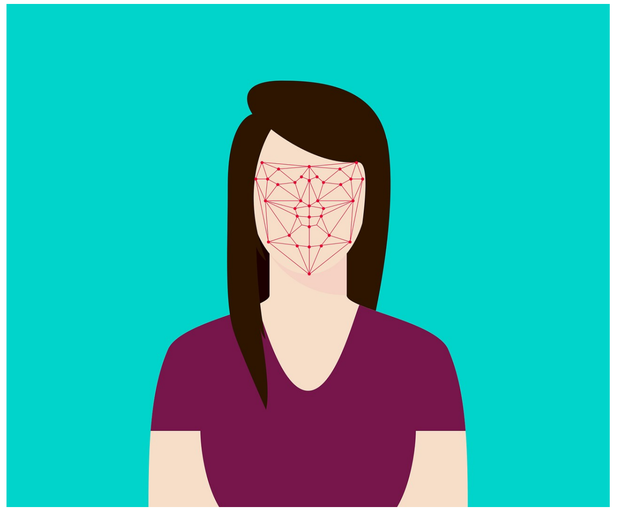
Erasmus+
Ανοίγει δρόμους, αλλάζει ζωές.

EXAMPLES OF REAL-WORLD ALGORITHMS

SORTING PAPERS #1



FACIAL RECOGNITION #2



TRAFFIC LIGHTS #4

GOOGLE SEARCH #3

BUS SCHEDULES #5

ΙΔΡΥΜΑ ΚΡΑΤΙΚΩΝ ΥΠΟΤΡΟΦΙΩΝ IKY
Co-funded by the Erasmus+ Programme of the European Union

LESSON #2 - ALGORITHMIC THINKING

Erasmus+
Ανοίγει δρόμους, αλλάζει ζωές.

ACTIVITY #2.1

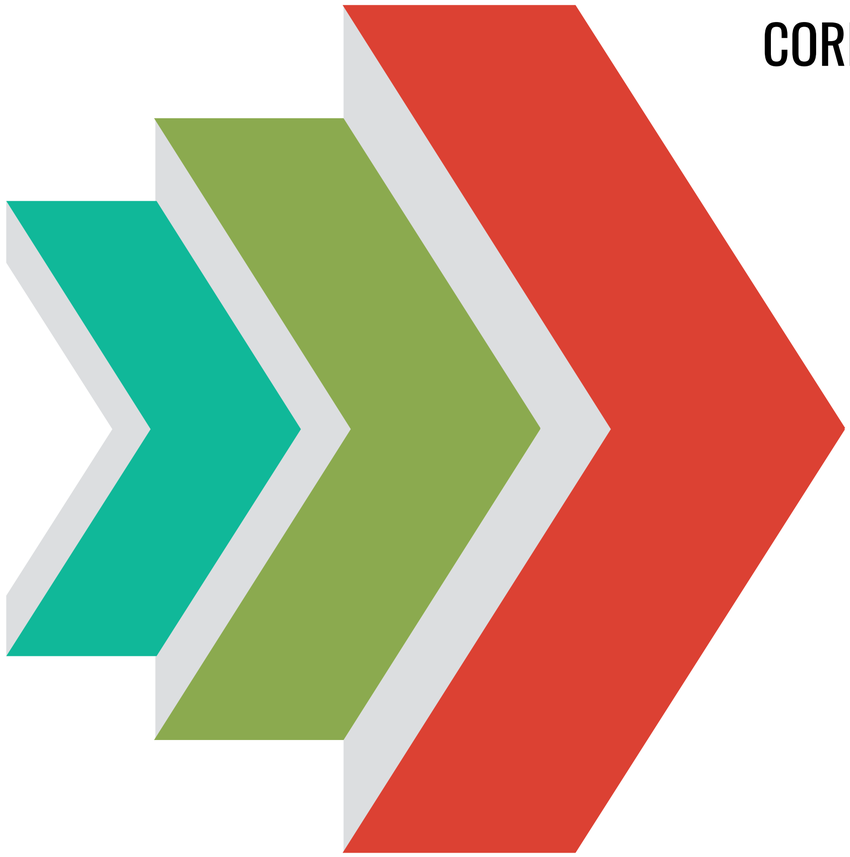
The trainer shares a sheet with a problem to be solved. The problem is about a shepherd who is planning to pass through a river by using his boat the below cargos:

- a sheep
- a wolf
- a grass

The transportation should be completed in safety since the wolf might eat the sheep or the sheep might eat the grass. The boat capacity is small so can only pass on and only one cargo of the above.

Trainers, should design the algorithm.

The discussion follows last 10 minutes.



CORE SKILLS DEVELOPED

- Problem solving skills
- Reasoning skills
- Environmental skills
- Technical skills
- Time management skills

TIMING

30 min

REQUIRED TOOLS

PC, projector, sheet

ΙΔΡΥΜΑ ΚΡΑΤΙΚΩΝ ΥΠΟΤΡΟΦΙΩΝ IKY
Co-funded by the Erasmus+ Programme of the European Union

REFERENCES

BEECHER, KARL. 2017. COMPUTATIONAL THINKING: A BEGINNER'S GUIDE TO PROBLEM-SOLVING AND PROGRAMMING. SWINDON, ENGLAND: BCS: THE CHARTERED INSTITUTE FOR IT.

KNUTH, D. (1997) THE ART OF COMPUTER PROGRAMMING, VOLUME 1: FUNDAMENTAL ALGORITHMS. BOSTON, MA, USA: ADDISON-WESLEY

PANE, J. F. ET AL. (2001) STUDYING THE LANGUAGE AND STRUCTURE IN NON-PROGRAMMER'S SOLUTIONS TO PROGRAMMING PROBLEMS. INTERNATIONAL JOURNAL OF HUMAN-COMPUTER STUDIES, 54 (2). 237.

PEA, R. ET AL. (1987) THE BUGGY PATH TO THE DEVELOPMENT OF PROGRAMMING EXPERTISE. FOCUS ON LEARNING PROBLEMS IN MATHEMATICS, 9 (1). 5.