COMPUTATIONAL THINKING AND ALGORITHMS

Slovenia, 2019

This material was made for

Erasmus+: OPEN YOUR DOORS TO DIGITAL AGE project

WARMUP

- What is computational thinking?
- What is an algorithm?
- What is a program?

PART 1: COMPUTATIONAL THINKING

INTRODUCTION (FOR TEACHERS)

- While studying chemistry, we acquire and expand the knowledge and understanding of certain scientific methods over time.
- While studying music, we improve our rhythm performance and sense of relations of sounds and tones.
- With studying computer science, however, by learning about its basic principles and computer language, we acquire a multitude of specific problem-solving skills (The Open University, 2015).
- Using the technology is only a small part of the skills we need in a world where technology is found in virtually every area of life.
- We will achieve this no only by knowing the answers to how and what, but also WHY. Computational thinking supports this kind of learning.
- In large companies such as Google and Microsoft Research **computational thinking is an** essential skill and technique for software development.

WHAT IT ISN‘T

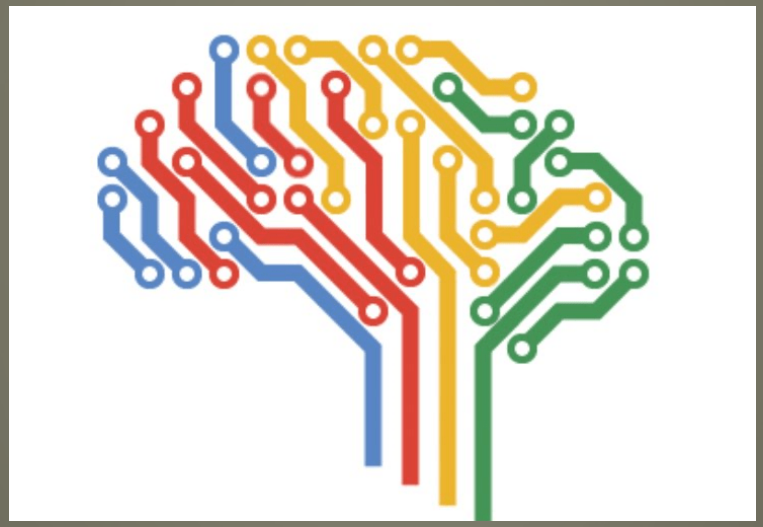
- Computational thinking isn‘t „thinking like a computer“
- Computational thinking isn‘t necessarily coding/programming.
- Computational thinking doesn‘t always require the use of computers.
- Technological literacy (ability to use and understand technology, e.g. computers, phones…) isn‘t required for computational thinking.
- Computational thinking isn‘t just coding a program, but thinking on multiple levels of abstraction.
- Computational thinking is used also by psychologists, doctors, mechanics, teachers…
- Computational thinking isn‘t just for computer science scientists – everyone should learn it.

WHAT IT IS

- Computational thinking can be done with or without computers.
- Computational thinking isn‘t only programming – but also problem solving.
- Developing computer thinking:
  - provides a deeper **understanding** of problems and their solutions
  - presents endless possibilities for creative problem solving,
  - develops the ability to work with others for a common goal or solution,
  - enhances already known problem solving techniques.

STEPS OF COMPUTATIONAL THINKING

1. **Decomposition**: breaking the main problem into many smaller and easier problems.
2. **Finding patterns:** Finding similarities between the main problem and similar problems for which we already know the solutions.
3. **Abstraction**: finding the most important parts and ignoring unnecessary parts (ordering the steps, sorting into categories, simplifying a complex idea).
4. **Algorithmic thinking (developing an algorithm)**
5. **Evaluation**: Deciding if the steps to the solution are correct and most appropriate. We evaluate speed, cost-effectiveness in terms of resources, ease of use.



VIDEO ABOUT COMPUTATIONAL THINKING

WE ARE BECOMING COMPUTATIONAL THINKERS

PART 2: ALGORITHM

HOW TO BRUSH YOUR TEETH?

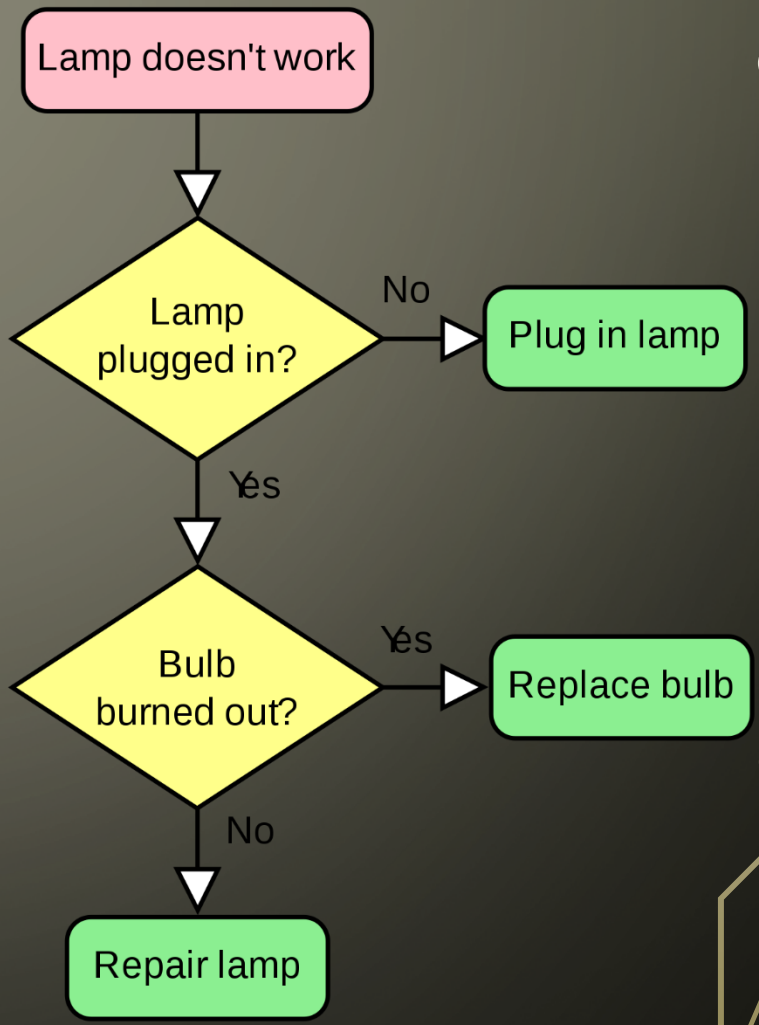
HOW TO SOLVE A RUBIK‘S CUBE?

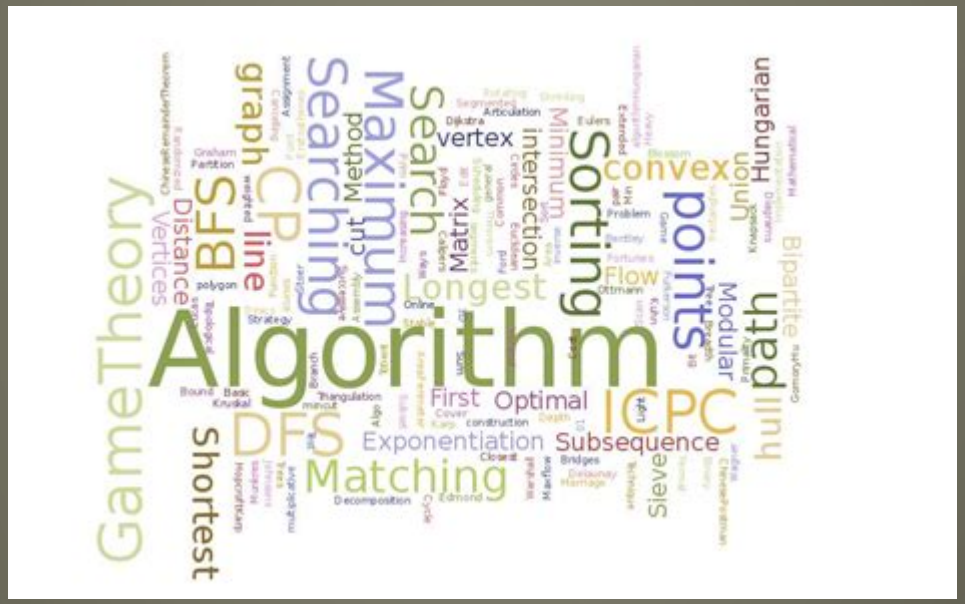
AN ALTORITHM

- An **algorithm** is a list of steps to follow in order to solve a problem.
- How detailed the steps are depends on who is going to follow the algorithm.
- If the algorithm is for a computer, then we are making it for a **computer program**.
- Algorithms need to have their steps in the **right order**.
- Usually the computer reads the steps from top to bottom. Sometimes it can repeat them many times. Sometimes it can follow the steps only if there is the right condition.

FLOWCHARTS

- A flowchart is a diagram that graphically shows how an algorithm works, step-by-step.
- We use different boxes with arrows.





[VIDEO ABOUT ALGORITHMS 1]
[VIDEO ABOUT ALGORITHMS 2]

SOURCES

- Computational thinking, http://ikt-projekti.uni-lj.si/RacunalniskoRazmisljanje.html
- Computational thinking, https://teachyourkidscode.com/what-is-computational-thinking/
- Algorithm, https://en.wikipedia.org/wiki/Algorithm