TRAINING · PROGRAMME FOR · INNOVATIVE · TEACHING AND

PIT

1#Involution

1.1 Didactic commentary

Dominic Harion & Ann Kiefer

"With Big Data and the associated algorithms, we are witnessing an increased return of mathematics and scientific methods to the organisation of social matters", stated Felix Stalder on automated decision systems in our digital cultures (Stalder, 2017). Far from the importance bestowed upon them in computer science, algorithms are, firstly, in general terms, "a set of instructions for solving a problem or a type of problems and are made up of a finite number of well-defined units". Even instructions for use of a machine and recipes are ultimately just algorithms. Understanding what such algorithmicity means in our world and what it entails particularly when it is interlaced with the technology we use in our day-to-day lives, plays a role in a well-founded educational requirement, as do the skills to model, apply and modify it, if necessary. First of all, algorithms are neither specifically technological, technical nor even computational. However, they can no longer (and should not) be removed from our infrastructures. In these specific forms, they are responsible for complex processes.

When it comes to the systems of mapping and transport management in topography, algorithms play an integral part in social organisation processes. Public transport maps and transport applications are two specific examples of this. "The shortest path algorithm" on a graph, also known in specialist settings as the Breadth-first search algorithm or Dijkstra's algorithm (in cases where the graph's nodes have weighs) represents a case study similar to everyday life and, at the same time, it can be easily studied independently for use in class. As part of the module presented herein, a fun and collaborative approach was chosen using Involution©, a mathematical game or puzzle developed at the University of Luxembourg by Hugo Parlier and Bruno Teheux. Learning about algorithmic modelling using fun and competitive steps has been tried and discussed for a long time with the help of the Rubik's Cube (see, for example, Lakkaraju et al., 2022). It opens up learning pathways to the mathematics and computer science interface (see, for example, Joyner, 2008; Agosinelli et al., 2019). Although these cubes can be used to illustrate 3D models, Involution© allows us to reproduce and clarify algorithmic models in a 2D space. Therefore, this game is ideal for expanding interdisciplinary educational horizons in the computational sciences through the formation of mathematical models linked to real world problems: the solution to Involution© lies in finding the shortest path in an underground train system, a task that the pupils already know from the Digital Sciences course (Digital Sciences, 2021). Children and adolescents from different age groups have already explored the search for the shortest path and it has been used to embody formal abstract models (cf. Gibson, 2012). It can therefore be used with gradual levels of difficulty for differentiated learning.

From a teaching perspective, the module is based on the two principles of problem-based and cooperative learning. A template and basic instructions are given to pupils who are faced with a problem. It encourages pupils to use a targeted cognitive process in order to transform a given situation into a target situation, without providing an obvious method to solve the problem, while, at the same time, appealing to creative and critical thinking (cf. Mayer & Wittrock, 2006). This approach inspires pupils to change their opinion on STEM (Science, Technology, Engineering and Mathematics) subjects. For many young people, these subjects are about finding the "one and only solution" and doing so as quickly as possible, yet that does not reflect the reality of the scientific world. Generally speaking, STEM professionals do not have a clear idea of the direction to take to find a solution. This mindset is sometimes overlooked in STEM subjects at school. Games and puzzles like Involution© provide the perfect framework to develop pupil resilience. "Resilience is related to students' affective ability to deal with and be able to overcome obstacles and negative situations in the learning process, turning those negative situations into situations that support them." (Hutauruk & Priatna, 2017). Pupils who are not used to managing frustrating learning circumstances and failures see them as very negative. However, if pupils become used to it, then this experience of resilience will have extremely positive outcomes for the studies and future employment of young people (Hutauruk & Priatna, 2017).

Each pupil does not solve the task individually, but in pairs or small groups. It is designed to be competitive (but not in the sense of a competition), like a class rally. This fun and motivating approach also encourages pupils to model and articulate their personal solutions and offers opportunities for learning by teaching, as the pupils have to express their thought process and back it up with arguments. A configuration in which the groups are constituted homogenously based on the difficulty of the tasks could be considered, as could another configuration with heterogenous groups in which the pupils are coached by their classmates while they come up with solutions.

References:

Agostinelli, Forest, Mavalankar, Mihir, Khandelwal, Vedant, Tang, Hengtao, Wu, Dezhi, Berry, Barnett, Srivastava, Biplav, Sheth, Amit & Irvin. Matthew. (2021). Designing Children's New Learning Partner: Collaborative Artificial Intelligence for Learning to Solve the Rubik's Cube. In Interaction Design and Children (IDC '21). Association for Computing Machinery, New York, NY, USA, 610–614. https://doi.org/10.1145/3459990.34651750

Digital Sciences. (2021). Mon monde numérique et moi. Schüler Achse 1.

Gibson, Paul J. (2012). Teaching graph algorithms to children of all ages. In Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education (ITiCSE '12). Association for Computing Machinery, New York, NY, USA, 34–39. https://doi.org/10.1145/2325296.2325308

Hutauruk, Agusmanto J.B., & Priatna, Nanang. (2017). Mathematical Resilience of Mathematics Education Students. J. Phys.: Conf. Ser. 895 012067. https://iopscience.iop.org/article/10.1088/1742-6596/895/1/012067/pdf

Joyner, David (2008). Adventures in group theory: Rubik's Cube, Merlin's machine, and other mathematical toys (2nd ed.). Baltimore : John Hopkins University Press.

Lakkaraju, Kausik, Hassan, Thahimum, Khandelwal, Vedant, Singh, Prathamjeet, Bradley, Cassidy, Shah, Ronak, Agostinelli, Forest, Srivastava, Biplav, & Wu, Dezhi. (2022). ALLURE: A Multi-Modal Guided Environment for Helping Children Learn to Solve a Rubik's Cube with Automatic Solving and Interactive Explanations. (Preliminary Preprint). https://www.aaai.org/AAAI22Papers/DEMO-00182-LakkarajuK.pdf

Mayer, Richard E. & Wittrock, Merlin C. (2006). Problem Solving. In P. A. Alexander & P. H. Winne (Eds.). Handbook of Educational Psychology (2nd Ed), 287-303. New York : Routledge.

Rogers, Hartley (1987). Theory of Recursive Funtions and Effective Computability. Massachusetts : C1957.

Stalder, Felix. (2017). Algorithmen, die wir brauchen. https://netzpolitik.org/2017/algorithmen-die-wir-brauchen/

1.2 Lesson planning

01 | Theme of the lesson in the overall structure of the axes

Module	Axes	Focus	Interdisciplinary ideas and link with other subjects
#Involution	Topic 1 My digital world and me!	• Games and algorithms • Algorithm of the shortest path	• Mathematics • Geography
#Climate Killer Internet	Topic 2 Understanding the internet: the World Wide Web and me	• Internet and climate • Judgement skills	• VIESO (Life and Society) • Geography • German • French
#Data Viz Superpowers	Topic 3 Do you speak Informatique?	• Different types of data visualisation • Manipulation of graphics	• Artistic education • Mathematics
#Discover Life on Mars with a Rover	Topic 5 Robots, partners for better or worse?	• Programmation en Scratch • Educational Robotics	• VIESO
#Pupils vs Machine	Topic 6 Is there a machine as intelligent as I am?	• Basic functioning of AI	• Mathematics • VIESO

Since the modules are independent from each other, it is not necessary to be acquainted with the previous modules to tackle this one.

This module was devised in partnership with Hugo Parlier and Bruno Teheux from the Mathematics Department of the University of Luxembourg.

02 | Conditions of unit

1. Target audience: 7e-5e (first to third year of classical and general secondary education)
2. Room: no specific premises required
3. Equipment required:
   - The Involution© game: the game can be borrowed from the IFEN's Centre of Pedagogical Documentation or the F.use (Future Spaces for Education) on the Belval Campus.
   - Computers or tablets can be used to read the instructions or you can simply print the instructions.
4. Time: 2 hours of teaching time

03 | Contextualisation of knowledge

The main objective of this lesson is to embody the notions of algorithm seen in class via Involution©: a game or puzzle developed by Hugo Parlier and Bruno Teheux, two mathematics researchers at the University of Luxembourg. Using this game, the pupils will familiarise themselves with mathematical and algorithmic concepts such as the shortest path algorithm and optimal and non-optimal strategies, all in a fun way. The game comprises a row of 10 black and white rings. A circular-shaped crank handle is placed on the game and the rings change position in a circular movement. The goal of the game is to change the position of the rings in order to get a given configuration based on the starting configuration.

04 | Didactic transposition

a. Learning objectives and target skills

Target skills in Topic 1: My Digital World and Me!

- KNOWLEDGE: a basic understanding of general computational and computer-related
  vocabulary, of human-computer communication (algorithmic notion) and of the abstraction principle when it comes to problem solving.
- KNOW-HOW: breaking down a problem in order to solve it, algorithm design and representation, simple algorithm production and application using maps and an operation sequence.
- SOFT SKILLS: an awareness of how computer science is involved in daily tasks based
  on the choice of tools, the information provided and the outcomes obtained.

Target skills in the Media Compass[1]

- Competence 2 – Communication and collaboration: 2.1. Working with others
- Competence 3 – Creating content: 3.3. Modelling, structuring and coding

[1]https://www.edumedia.lu/medienkompass/medienkompass/

b. Didactic justifCcation

The objective of this module is for pupils to gain an understanding of the notion of algorithms and how they work by way of a game. One of the module's distinctive features is that the pupils are not to use electronic devices, computers or tablets. These so-called "unplugged" activities are often used to introduce computational thinking in order to "escape from the technological complexity of learning and to discover the fundamentals of computer science" (INRIA, 2020). This makes it easily accessible and enables pupils to understand a key area of digital technology: firstly, digital cultures do not have anything to do with computers in terms of electrical machines. The distinguishing feature of algorithmicity, for example, which is at the heart of this module, can be described as forming formal mathematical and logical models, which can also be represented in a mechanical way. To quote the computer scientist Leslie Lamport, "The importance of thinking and writing before you code needs to be taught in undergraduate computer science courses and it's not" (Han, 2022).

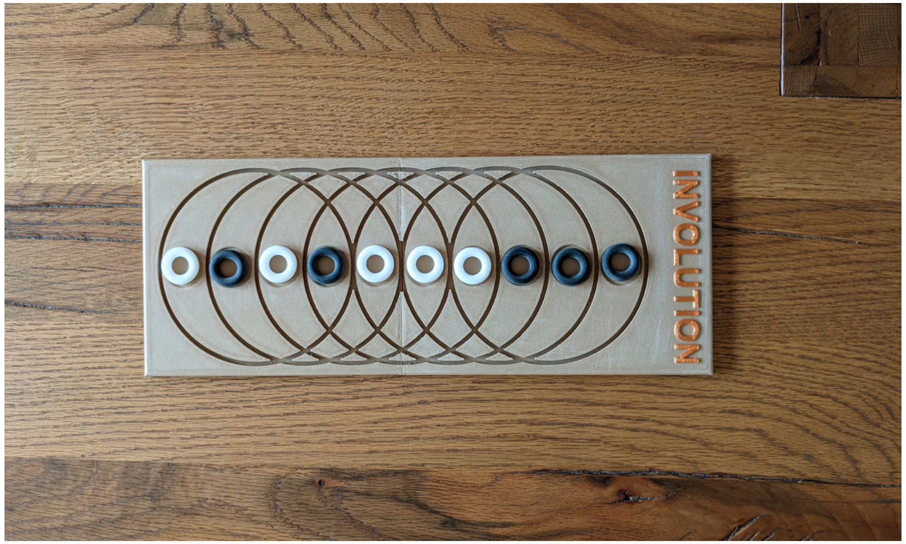
Thus, the #Involution module is based on the foundations of coding taught in primary schools and builds on them, at the crossroads of mathematics and computer science.

c. Didactic reduction

Involution© poses several "shortest path" type challenges at different levels and places them within the cooperative learning and problem-solving framework. The game stimulates a targeted cognitive process which can be actively developed based on the teaching material handed out. When it comes to the planning and implementation of the teaching unit, a strict time limit can be completely disregarded in favour of a more freely experimenting with Involution© and the instructions provided. Knowledge and skills are consolidated through learning by teaching as the pupils will present their solutions and the reasons for their choices to the class.

05 | Over the course of the lesson

In this module, the pupils play a game called Involution©. The game is made up of a tray with a crank handle. The tray has 10 holes that each contain one black or white ring.



A circle-shaped crank handle is placed on the game and the rings change position in a circular movement. The goal of the game is to change the position of the rings in order to reach a given configuration based on the starting configuration.

Involution 1

The game

The teacher briefly explains the principle of the game. The pupils are introduced to the game by simply playing it. They are split into pairs. Each group receives a box that contains a game tray (to be assembled) and challenge cards. Each card has two configurations. The challenge entails going from the configuration on the top to the configuration on the bottom by moving the crank handle. The pupils start by tackling the challenges in the given order. This activity is the first challenge in a long list of challenges in this module. All the challenges are provided in the teaching materials.

After this introduction, the pupils move on to the other challenges. The explanations below are for the same challenges. The module includes numerous challenges. The best-case scenario is that the pupils complete all the challenges. Starred challenges are more complicated (and more mathematical in their nature). They are designed for more eager pupils. Moreover, there are further optional challenges (as shown below in the detailed unit plan). Choosing not to take on these extra challenges will not impede the pupils from completing the module.

Challenge 2: After the initial stage, once the pupils have understood the principle of the game, they move on to challenge 2. The teacher divides the class in three groups; one group works on the first of three challenges; another works on the second and the final group on the third. The first challenge is slightly easier than the other two: please take this into account when dividing the class into groups and allocating the challenges.

Once the challenge has been solved, the pupils must agree among themselves, then communicate their solution to the rest of the class.

Challenge 3*: As the most complicated challenge, challenge 3 is optional. It should be taken on by eager pupils with a good level.

How to obtain all the confCgurations

This section helps pupils solve challenge 3. It concerns only the pupils who have tried it. Challenge 9 is one of the more demanding challenges in this section. We will leave it to the teacher to decide which pupils can/should take on the challenges in this section with or without challenge 9.

This section guides pupils towards solving challenge 3 using the mathematical principle of induction (without talking about the term itself or even mentioning it).

An optimal solution

Challenge 10: In challenge 10, the pupils look for a solution and count the number of movements they make to find that solution. They compare their solutions with each other.

In a moderated discussion, they come to the conclusion that one solution is better than another if it requires fewer movements.

A definition similar to the one below is formulated by the whole class:

A solution is optimal when it involves as few moves as possible.

The quest for an optimal solution

How do we find an optimal solution? Make the connection with the search for the shortest route in the New York underground, seen in Digital Sciences 1.

How to represent the game in graph form

This section starts with a more straightforward example: the Involution© game, but with only 5 rings and a reduced game tray. The 5 holes closest to the right must not be used. Therefore, there are only 2 possible positions in which to put the crank handle.

Challenge 11: The teacher divides the class in three groups. Within each group, the pupils are divided into pairs to play the game.

- The first group plays with 1 black ring (and 4 white rings).
- The second group plays with 2 black rings (and 3 white rings).
- The third group plays with 3 black rings (and 2 white rings).

The pupils try to solve their problem in pairs. Then they discuss it with their group and finally explain their outcome to the rest of the class.

Challenge 12*: This challenge is more complicated and is optional.

The pupils move on to the graph representation of the game (with the same game in a reduced format).

Challenge 13: For this challenge and those following it, the graph concept must be introduced or a brief reminder of it must be given. The pupils work in pairs, then they compare their results. They will find that the second and third cases are exactly the same. In a moderated discussion, the pupils will see that these two cases are symmetrical.

Challenge 14: Once the graph is constructed, challenge 12 becomes simpler. The pupils will see the answer clearly on the graphs.

In the next stage, the pupils repeat the game with 10 rings (5 black rings and 5 white rings).

Challenge 15*: This challenge is optional. The idea behind the challenge is to show the pupils that the counting exercises become complicated as the problem becomes more complex. Many pupils do not necessarily realise this because counting is very simple when only a few objects are involved.

The Involution© game's graph representation is given, but it is too large to look for the shortest path with the naked eye.

The quest for an algorithm

To make the quest for an algorithm easier, the pupils move to 6 rings (3 black and 3 white). The 6-ring Involution© is small enough to search for the shortest path manually (contrary to the 10-ring Involution© graph), but large enough for the solution not to be ridiculous (contrary to the 5-ring Involution© graph).

Challenge16 et 17: These challenges are used to put the pupils firmly on the track of an algorithm. These challenges are tackled in pairs. The outcomes are then discussed in class in the form of a moderated discussion.

Challenge 18: In this challenge, the algorithm is established manually. The pupils must write informal instructions. Point b) is more complex. It is intended for the most eager pupils only.

To draw the structural chart, the pupils can choose from drawing it from scratch or using the help already provided in the fields. The pupils must then put the fields in the right order and add arrows to form a structural chart.

Challenge 19: This challenge is only used to show the pupils that the general algorithm is exactly the same as those they have just done. This challenge is to be carried out directly in a moderated discussion.

Challenge 20: In a moderated discussion, the pupils realise that the algorithm they have just established is a shortest path algorithm. The same types of algorithms are used by GPS car navigators (see 1.6.02 The importance of graph theory in computational sciences).

06 Differentiation possibilities

For beginners

The module is completely feasible by dropping the starred challenges and sections. Encourage weaker pupils to do the module without these challenges.

For more eager pupils

Include the starred challenges (and the starred section) in exercises for more eager pupils. These challenges are slightly more mathematical in their nature and include logical reasoning. There is also a two-starred challenge. It is only intended for pupils with a good level and a decent grasp of the subject.

In challenges 2 and 11, the teacher can divide the pupils into three groups so that the pupils with the weakest level are given the most straightforward case to solve.

Challenge 18, that is to say the main challenge, where the algorithm is established, has three levels of difficulty:

1. Establishing an algorithm that does only count the minimum number of movements and where the structure chart fields are already given.
2. Same as above but without the structure chart fields.
3. Establishing an algorithm that counts the minimum number of movements and the exact sequence of these movements.

07 Further criteria to be met as part of the lesson series

a. **The Luxembourgish context:** the Involution© game was invented by Hugo Parlier and Bruno Teheux, two mathematics researchers from the University of Luxembourg. In its current form, the game can be linked to the contents in the *Luxembourg Media Compass* and it corresponds to the key topic I in *Digital Sciences*.
b. **Differentiation:** several of the module's challenges are optional. The starred challenges are more complicated than the others and can be used (or not) depending on the level of the class and/or pupils.
c. **Media Compass:** see the learning objectives set out by the *Media Compass* in the teaching analysis section in this document.
d. **The 4Cs model: communication, collaboration, creativity and critical thinking:** the 4Cs model is applied in a range of ways through the different social forms and teaching activities.
e. **Relation to current research:** lhe challenges in Involution© are part of a family of reconfiguration problems (combinatorial reconfiguration), an area of research currently being studied in maths and computer science. The shortest path algorithms are more than 50 years old, but they remain the subject of research for the purpose of optimising shortest path applications.
f. **Relation to research in Luxembourg:** the Involution© game was developed by two researchers from the University of Luxembourg. In the podcast in section 1.7 A word from the scientists, Hugo Parlier explains how he develop games by using the results of his research.

References:
Institut national de recherche en sciences et technologies du numérique (INRIA). (2020). *Éducation et Numérique : enjeux et défis.* Livre Blanc N 04. https://hal.inria.fr/hal-03051329v2/document
Han, Sheon. (2022). How to Write Software With Mathematical Perfection. Quantamagazine.
https://www.quantamagazine.org/computing-expert-says-programmers-need-more-math-20220517

Theme of the lesson: Discovering algorithms through a game

Learning objectives and skills to be developed during the lesson: The pupils understand the basics of algorithmic problem solving and these skills are put to test in a real game. In so doing, they have developed cooperative problem-solving strategies and thought about them metacognitively based on independent documentation.

OPTIONAL AND DISCRETIONARY challenges: In the following overview, a distinction is made between the optional and discretionary tasks. Depending on the speed at which the pupils work through the different stages, it is possible to spontaneously decide whether or not certain tasks should be omitted or added.

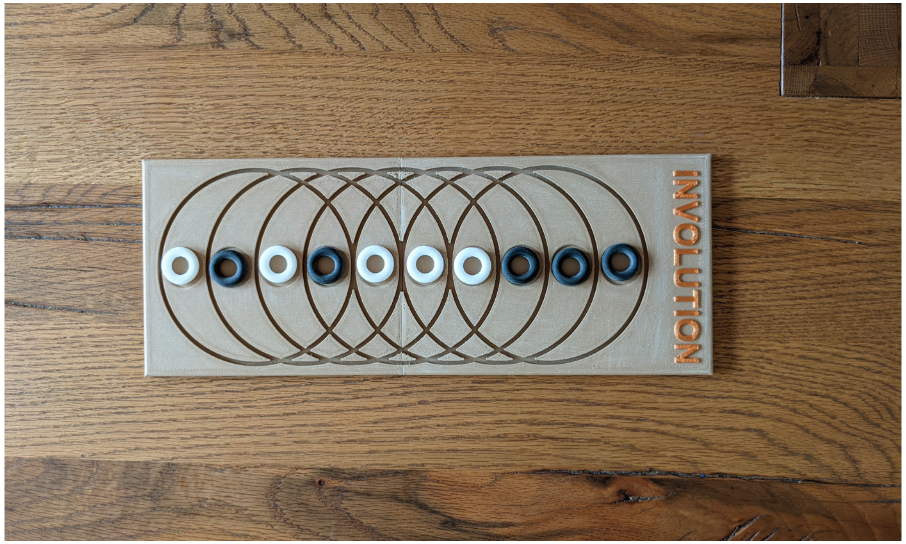
(Possible) evaluation: Several ways of evaluating the pupils are proposed in section 1.5 Evaluation ideas with different levels of difficulty.

Duration	Phases	Focus	Social forms/Methods	Materials and media	Learning processes
10min	Introduction	**Learning how the Involution© game works Challenge 1**	• Groups of two • Classroom lecture • Work in pairs	• Involution© • Worksheets with instructions and challenges (print-outs or on tablet)	The pupils ...**understand** the concept and how the Involution© game works.
20min	Involution©	**[OPTIONAL] Elaboration Challenge 2 & 3**	• Work in pairs	• Involution© • Worksheets with instructions and challenges (print-outs or on tablet)	The pupils ... **come up** with a solution to the proposed challenges. ... **reflect** on their strategies and solutions.
20min	Involution© An optimal solution	**Elaboration Challenge 10**	• Work in pairs	• Involution© • Worksheets with instructions and challenges (print-outs or on tablet)	The pupils ... **come up** with a solution to the proposed challenges. ... **reflect** on their strategies and solutions.
25min	Involution© The quest for an optimal solution	**Elaboration Challenges 11-15 [OPTIONAL: Challenges 12, 14 and 15]**	• Work in pairs	• Involution© • Worksheets with instructions and challenges (print-outs or on tablet)	The pupils ... **come up** with a solution to the proposed challenges. ... **reflect** on their strategies and solutions.
25min	Involution© The quest for an algorithm	**Elaboration Challenges 16-20**	• Work in pairs	• Involution© • Worksheets with instructions and challenges (print-outs or on tablet)	The pupils ... **come up** with a solution to the proposed challenges. ... **reflect** on their strategies and solutions.

1.3 Teaching materials

01 | The game

In this module, we are going to play a game called Involution©. The game includes a tray with a crank handle. The tray has 10 holes that each contain 1 ring (of one colour).



The rings can be moved as shown in the following video:

Involution 1

Challenge 1

Get into pairs and play Involution©: try to solve the problems on the cards.

Challenge 2

Starting with the configuration on the left, can you get to the configuration on the right?

○○○○○●●●●●	●○●○●○●○●○
○○○○○●●●●●	●●●●○●○○○○
○○○○○●●●●●	●●●○●●○○○○

If so, explain your movements. If not, why not?

Challenge 3*

Can all the ring configurations be obtained? If so, how? If not, which of them can't be obtained? *Justify your answer.*

02 | How to obtain all the confCgurations

For a clearer understanding, let's now try to play the same game but with 9 white rings and one black one.

The key question is as follows.

Challenge 4*

Is it possible to obtain the configuration below from any other configuration? *Justify your answer.*

●○○○○○○○○○

Let's now play with two black rings.

Challenge 5*

Is it possible to obtain the configuration below from any other configuration? Use your solution from challenge 4 and justify your answer.

●●○○○○○○○○

Let's now play with three black rings

Challenge 6*

Use the solution from challenge 5 to see if you can reach the following configuration

●●●○○○○○○○

starting from any other configuration.

Finally, let's play with four black rings.

Challenge 7*

Use the solution from challenge 6 to see if you can obtain the following configuration

starting from any other configuration.

●●●●○○○○○○

Challenge 8*

Review challenge 3*.

Challenge 9**

If we play the same game, but with 12 rings (6 black and 6 white), is it possible to obtain all the configurations? What about with 100 rings (50 black and 50 white)?

03 | An optimal solution

Let's repeat the game with 10 rings.

Challenge 10

Starting with the following configuration

find the following configuration

○○○○○●●●●●

and count the number of movements.

●●●●●○○○○○

Compare your solutions with the rest of your group. Which is the best? Define a criterion to evaluate why one solution is better than another.

After discussing it, try to complete the following definition:

A solution is optimal when/if...

04 | The quest for an optimal solution

Let's now build a visual representation of our game in order to find an optimal solution. As a reminder, one of the first algorithms we came across in Digital Sciences 1 was an algorithm to find a route on the New York underground. Just as the underground map is a graphic representation of the real network, we will make a graphic representation of the game: the network nodes are the different configurations of the game (in the case of the underground, the nodes were the stations) and two configurations are linked by a line if you can go from one configuration to another in one movement (in the case of the underground, two stations are connected by a line if you can go directly from one station to the other via the underground).

Here's an example: let's play Involution©, but with only 5 rings and a smaller tray (take the same tray, but do not use the 5 holes on the right). You are only allowed to make two different movements with the crank handle.

Challenge 11

The class is divided into 3 groups.

- The first group plays with 1 black ring (and 4 white rings).
- The second group plays with 2 black rings (and 3 white rings).
- The third group plays with 3 black rings (and 2 white rings).

In groups of two, draw all the possible configurations of your version of Involution©.

Together, *discuss and explain* how many configurations of your version of Involution© are possible. Then present your solution and your explanation to the whole class.

Challenge 12*

Decide if, in your case, it is possible to obtain any configuration whatsoever and explain your response.

We will now construct the graphic representation.

Challenge13

Give a name to each configuration (for example C1, C2, C3, etc). For each configuration found in challenge 11, draw a node and write its name beside it. Then, connect the nodes with a line if it is possible to go from one configuration to the other with just one move of the crank handle. Compare your graphs.

Challenge 14

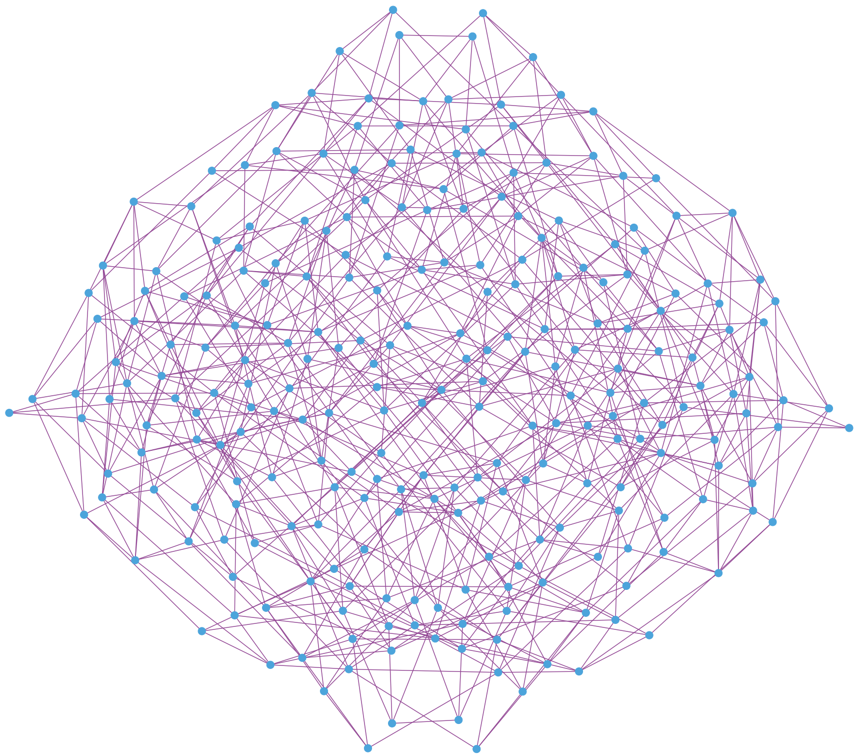
Go back to challenge 12 (by using the graph you have just drawn).

Now let's return to the initial game with 10 rings.

Challenge 15*

Can you calculate the number of different configurations or at least give an upper limit?

In total, there are 252 different configurations. This gives a graph with 252 vertices, which looks like this:



The 10-ring Involution© graph is too large to see the shortest path with the naked eye. That's why we need an algorithm.

05 | The quest for an algorithm

Challenge 16

Go back to your game and play with 6 rings now (3 black and 3 white) and a smaller tray (take the same tray, but do not use the 4 holes to the right). Find all the possible configurations and draw the graph.

Challenge 17

Use the graph found in challenge 16 to determine an optimal solution for obtaining the configuration below

starting with the following configuration:

The Big Bang Theory Sheldon's Friendship Alogrithm

Challenge 18

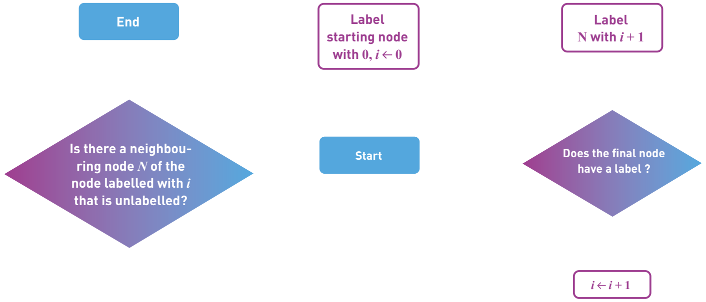
Based on the previous challenge, look for an algorithm that gives:

1. The minimum number of moves to go from one configuration to another (once the graph for the game is given).

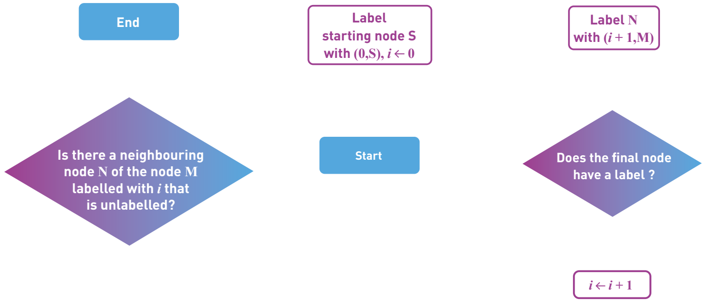
2. An optimal solution for going from one configuration to another (once the graph of the game is given).

Write some informal instructions for your algorithm and draw its flow chart.

Help 1 :



Help 2:

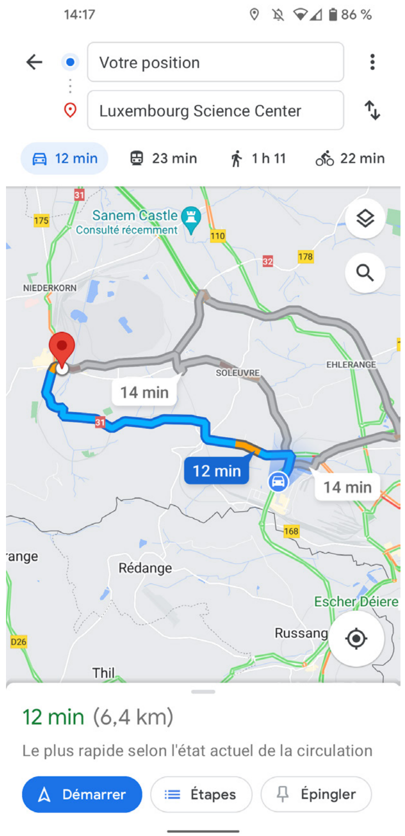


Challenge 19

Apply the algorithm to the Involution© game with 10 rings.

Challenge 20

Look at the image below:



What app is this? Look for a link between this app and the search for an optimal solution in the Involution© game.

1.4 Interdisciplinary ideas

Mathematics

This module targets the four skills relating to the mathematical processes listed in the documents *Compétences disciplinaires attendues à la fin de la classe de 6e et à la fin de la classe de 4e*
(Discipline-specific skills expected at the end of 6e and at the end of 4e) and *Compétences disciplinaires attendues à la fin des classes de 7G-6G-5G* (Discipline-specific skills expected at the end of 7eG, 6eG and 5eG) issued by the Ministry of Education. These four skills are as follows:

1. Problem solving: As explained in the Discipline-specific skills document, solving mathematical problems "is characterised, on one hand, by the implementation of general strategies, and, on the other, by the implementation of specific strategies". This is exactly what the pupils are led to do in this module: they start by discovering the game through testing a few examples, then they follow the strategy given in the module. This module is a wonderful example that demonstrates that you can practice problem solving by "working actively on problems and then reflecting on problem-solving methods and strategies".
2. Modelling: In this module, the pupils are first led to simplify the problem, then model it in graph form. This process accurately corresponds to the description of the skill to be modelled: "It is a matter of first simplifying the real situation and then mathematising it, i.e. describing it with mathematical tools".
3. Present your arguments: This skill is described as follows: "Mathematical argumentation starts with exploring situations, looking for structures and relationships and by formulating conjectures about the mathematical relationships." This is exactly what happens in challenges 3 to 8 in 1.3 Teaching materials. In the first exercise, the pupils are asked to formulate a conjecture, then in the following exercises, they are led to prove their conjecture.
4. Communicate: The exercises proposed in this module are not the typical exercises you would usually find on a mathematics course, but they need reasoning and argumentation to be proven. The pupils must, therefore, "explain mathematical content in an appropriate way using everyday language and mathematical language".

To be more precise, this module enables work to be done on a skill in the chapter on literal calculation in the new provisional mathematics curriculum for the 6eC. This skill reads as follows: decode a literal expression into an ordered series of calculation instructions and understand how to formalise this algorithm.

The Involution© game can also be used to illustrate central symmetry and rotation concepts. The game's crank handle movement is nothing more than central symmetry or 180-degree rotations. The geometric transformations are taught in 5e and 4e in Luxembourg and often pose problems for pupils: axial symmetry is much more natural to them and they find visualising these new transformations rather difficult. Involution© is another method of visualisation.

In lower technical secondary school, the pupils have already started to familiarise themselves with probabilities. Involution© enables the pupils to count and enumerate (how many different configurations exist?) and to calculate probabilities in many different ways.

Finally, this module prepares the ground for proof by induction. In challenges 4 to 8 of 1.3. Teaching materials, the pupils are led, stage by stage, to do proof by induction (without mentioning the mathematical term itself).

Geography

Using navigation algorithms (such as Google Maps), we can make the connection with the maps used by navigation apps. Directions and maps feature in the geography curriculum for the 7eC and 7eG.

1.5 Evaluation ideas

The group discussion on the methods of solving the different challenges using Involution© can be complemented by an evaluation. It contributes to the consolidation of the knowledge and skills acquired.

Create a solution book for Involution© in 3 colours

Let's play Involution© with 3 colours: 2 black rings, 2 white rings and 1 hole without a ring (the hole without the ring plays the role of the third colour). The pupils are responsible for writing a booklet containing optimal solutions for a number of challenges (determined by the teacher). The choice of presentation is up to the pupils – they can be simple descriptions of the processes, drawings, purely visual representations or a video produced by the pupils themselves.

To find the optimal solutions, the pupils must build the graph corresponding to Involution© with 3 colours and look for the shortest path in this graph (careful – this path may not always exist!).

The completed solutions booklet can then be exchanged among the pupils and checked for accuracy against the game. The pupils can then assess each other and give feedback.

It is also possible to do larger and more complex projects. Here are three different ones: the first is a more computer-science based one (and requires programming), the second one falls more into the area of social sciences and creativity and requires writing skills and the third is intended for pupils who are more interested in mathematics.

Programming Involution©

The goal is to write a programme (on Scratch or Python), using the algorithm established in challenges 15 and 16 to solve the Involution© game.

There are several levels of difficulty:

Write a programme that takes as the input two configurations of the Involution© game and gives as the output the minimum number of movements to go from one configuration to another.

Write a programme that takes as the input one configuration of the Involution© game and gives as the output a sequence of moves (not necessarily the sequence corresponding to the optimal solution), allowing to go from this configuration to the configuration where all the white rings are on one side and all the black rings are on the other.

Write a programme that takes as the input one configuration of the Involution© game and gives as the output the optimal sequence of moves allowing to go from this configuration to the configuration where all the white rings are on one side and all the black rings are on the other.

Invent a game for two players

The pupils should think about of a two-player version of Involution©. They must invent a collaborative version and a competitive version of the game. They are then invited to write clear rules of play. The choice of presentation is up to the pupils (small booklet, drawings, digital media, etc.).

And with 3 colours?

Let's play Involution© with 3 colours: n black rings, n white rings and 1 hole without a ring (the hole without a ring plays the role of the third colour). Here n can take any value greater than or equal to 2. The question the pupils have to answer is the following: from which value of n on can we obtain all the configurations starting from any other configuration?

1.6 More on this topic

01 | The BFS and Dijkstra algorithms

The algorithm that the pupils have just explored in this module is the shortest path algorithm. The technical name of this algorithm is Breadth-First Search Algorithm or simply the BFS algorithm. Invented in 1945, it allows us to search a graph by first exploring a source node, then its successors, then the unexplored successors of the successors, etc. The Breadth-First Search algorithm is used to calculate the distances of all the nodes from the source node on a graph. The BFS works by using a queue in which it takes the first vertex and places its unexplored neighbours last. The already explored nodes are marked in order to avoid exploring the same node several times. This algorithm only works on unweighted graphs, i.e., graphs on which the nodes are connected by edges, all of which are unweighted. It is indeed the case for the graphs in this module: two nodes (here configurations) are connected by an edge if one can go from one configuration to another with a single crank handle movement. On the other hand, the BFS works for directed and undirected graphs. A directed graph is a graph in which the edges between two nodes are directed, i.e., they go from one node to another (Diestrel, 2018), or have a sense of direction. As the Involution© game is symmetrical (if one can go from configuration A to configuration B with a single movement, one can go from B to A with the reverse movement, which is indeed exactly the same movement), the graphs explored in this module are undirected. The Computer Science Department of Cornell University published a series of video introductions to the BFS which last 15 minutes in total (Gries, n. d.).

Some graphs can also be weighted. The road network graph we saw in this module is a weighted graph: the distance between the two junctions varies from one situation to another. The graph must take this into account by assigning a weight to the edges. The shortest path problem is therefore no longer necessarily the path that crosses the fewest nodes and the BFS does not work on weighted graphs. The Dijkstra's algorithm is a well-known algorithm that gives the shortest path for weighted and unweighted graphs, both directed and undirected. It bears the name of its inventor, Edsger Dijkstra, and was published for the first time in 1959 (Dijkstra, 1959). The Dijkstra's algorithm works as follows: the algorithm starts at a source node and calculates the distances between this node and all other nodes on the graph. It then chooses the closest node to the source node. This node is added into a queue along with the source node, then all the distances are re-evaluated: if the distance between the source node and another node is shorter by passing through the new node, then it is replaced by the shortest distance. The algorithm continues as such until it has crossed all the nodes, which enables minimum distances between the source node and all the other nodes to be established. The Computer Science Department of Cornell University has also published an introduction to the Dijkstra's algorithm (Gries, n. d.).

02 | The importance of graph theory in computational science

On several occasions, the mathematician J. H. C. Whitehead stated that "combinatorics is the slums of topology" (Cameron, 2011). Combinatorics is a mathematical discipline which is part of graph theory. At the start of the 20th century, this view of graph theory became widespread among other mathematicians, but it has significantly, or rather completely, changed today. Graph theory is an important tool for studying the links between complex systems. As soon as a system can be represented diagrammatically in the form of nodes and edges, graph theory can be applied. These systems can be very diverse: road networks, urban data network, etc. The applications in the computational world are now endless. Some very important examples (Flovik, 2020) are:

- The spread of the COVID-19 virus throughout communities
- The ranking of webpages in search engines (such as Google)
- Network security
- GPS navigators
- And many others

Let's go back to GPS navigators (such as Google Maps, Waze, etc.), which are also involved in this module. GPS navigation systems are part of the graph theory applications which have a direct impact on our daily lives: we often rely on these systems to get to a place via the shortest (or quickest) route. A road network can be represented by a graph on which the nodes are junctions and buildings, and whose edges are the sections of the roads that connect them. As some roads are one-way, the graph is directed (unlike the Involution© graph which is undirected). The navigation system calculates an optimal route between two points of the network, based on the criteria defined by the user (it optimises either the length of the journey, the time of the journey or favours motorways, etc.). To illustrate the idea, imagine that the system is seeking to minimise the length of journeys. To model the problem, we must add information to the directed graph which models the road network. Each section of the road is assigned with a weight that represents exactly the length of that section. Therefore, navigators use directed and weighted graphs. The shortest path algorithm that navigators use is based on the Dijkstra's algorithm (Teheux, 2019).

Interestingly, GPS navigation systems implement the Dijkstra's algorithm in a reverse way. More precisely, if the goal of the user is to get from node A to node B, the navigation system applies the Dijkstra's algorithm to the resulting graph by reversing the edges of the road network graph with node B as the source node. The outcome is thus the shortest path from B to A. However, you only have to reverse this path to get the desired path. The idea behind this inversion is an economy of calculations. It is not uncommon for us, as drivers, to leave the route recommended by the navigator (we take a wrong turn, a road is closed to traffic, we change our plan at the last minute, etc.). If the algorithm had already performed the search for the shortest path in the right direction, it should start again (as we find ourselves in a place that was unforeseen in the algorithm's calculations) and all the calculations made before would have been lost. On the other hand, starting at the end, the algorithm has already calculated the shortest path from any point (within a reasonable radius) to the end point. Consequently, even if we change the itinerary, the algorithm is able to quickly provide us with an alternative (Teheux, 2019).

Although pure graph theory is an abstract and mathematical discipline, it shows that it has plenty of concrete and interesting applications in computer science. In any case, as world-renowned computer scientist Leslie Lamport states "If you really want to do things right, you need to write your algorithm in the terms of mathematics." (Han, 2022).

References:
Cameron, Peter J. (2011). Aftermath. Preprint. https://arxiv.org/pdf/1111.4050.pdf
Diestel, Reinhard. (2018). *Graph theory (5th Ed.).* Graduate Texts in Mathematics, 173. Berlin : Springer.
Dijkstra, Edsger W. (1959). A note on two problems in connexion with graphs. *Numer. Math.*, 1, 269 – 271.
https://doi.org/10.1007/BF01386390
Flovik, Vegard. (2020). What is Graph Theory, and why should you care. From graph theory to path optimization. *Towards Data Science.* https://towardsdatascience.com/what-is-graph-theory-and-why-should-you-care-28d6a715a5c2
Gries, David. (n.d.). *Depth-first search and breadth-first search.*
https://www.cs.cornell.edu/courses/JavaAndDS/dfs/dfs01.html
Gries, David. (n.d.). *The shortest-path algorithm.*
https://www.cs.cornell.edu/courses/JavaAndDS/shortestPath/shortestPath.html
Han, Sheon. (2022). How to Write Software With Mathematical Perfection. *Quantamagazine.*
https://www.quantamagazine.org/computing-expert-says-programmers-need-more-math-20220517
Teheux, Bruno. (2019). À la recherche des chemins les plus courts. *Losange*, N 46, 45-54.

1.7 A word from the scientists: podcast with Dr Hugo Parlier and conversation with Dr Bruno Teheux

Hugo Parlier is a professor at the Mathematics Department of the University of Luxembourg. His research centres on different topics of geometry, topology and combinatorics. He studied mathematics at the Swiss Federal Institute of Technology Lausanne where he received his PhD in 2004. After his postdoctoral stays in Madrid, Toronto and Geneva, he became an associate professor in Fribourg, with the support of the Swiss National Science Foundation.
In 2017, he joined the Mathematics Department of the University of Luxembourg.

Bruno Teheux is a permanent researcher at the Mathematics Department of the University of Luxembourg and conducts research into mathematical logic. He received his PhD from the University of Liège in Belgium in 2009. After working for two years at Animath in Paris, he joined the University of Luxembourg in 2012.

These two mathematicians love sharing their passion for mathematics through activities aimed at the wider public. They are the minds behind The Simplicity of Complexity and Recreate: shapes from the collective Imagination, which were presented at the Expo 2020 in Dubai. They have also participated in The Sound of Data as part of Esch2022, the European Capital of Culture.

In addition to these joint projects, Bruno Teheux has edited the pilot volume of scientific comics written by PhD pupils. As for Hugo Parlier, he has co-written an interactive mathematics book for the iPad called Mathema. This book gave rise to the puzzles for iPad or iPhone called Quadratis.

Interview with Bruno Teheux and Ann Kiefer

Unfortunately, Hugo Parlier was unable to take part in our interview, which is why we would like to point you in the direction of the wonderful podcast "Mäin Element" in which he features. This podcast was recorded by Lëtzebuerger Journal in partnership with the National Research Foundation (FNR).

Podcast Mäin Element with Hugo Parlier

ISSN 2738-9383

PIT

PROGRAMME FOR INNOVATIVE TEACHING AND TRAINING