Teaching algorithmics for Informatics Olympiads: The French Method

Arthur Charguéraud, Mathias Hiron

www.france-ioi.org

Presentation by Mathias Hiron

The French Method:

How we train students on www.france-ioi.org and at our training camps.

The priority: improving student's problem solving skills.

Accumulation of knowledge of classic algorithms is secondary.

The means:

- **A curriculum based on guided discovery learning**

  Students learn by solving series of problems instead of reading lectures.

- **Teaching a problem solving method**

  Students can solve harder problems if they apply these techniques

Classic curriculums, focused on knowledge

Most curriculums are based on:

- **lectures:** to teach new algorithms
- **examples:** to show how to apply them
- **application exercises:** to practice using them.

This works well to transmit knowledge, and understanding of that knowledge.

Our view:

improving skills to solve difficult problems is what really matters in algorithmics.

The aim of a teaching curriculum should be to improve those skills.

A curriculum focused on solving skills

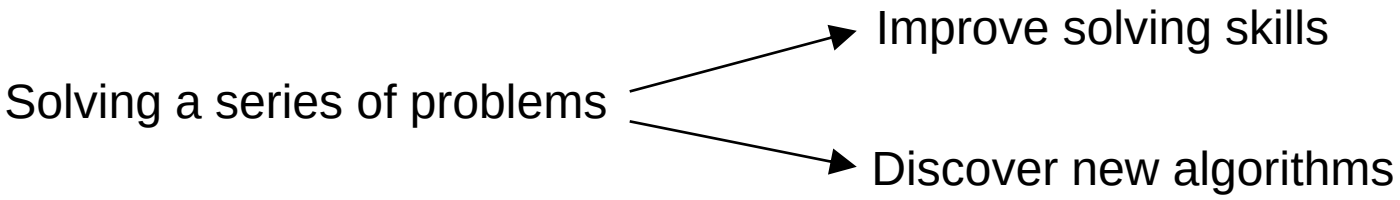
- **Series of problems:** to discover new algorithms in a given domain
  In each problem, a student discovers a new concept, then uses it in the following problems.

- **Hints:** to avoid getting stuck and teach steps of the method
  Help can be provided either in the form of :
  - a **textual hint**
  - a **new intermediate problem**

- **Detailed solutions:** to show solving techniques and reinforce knowledge

Explaining the expected algorithm is not enough.

A possible thought process leading to the solution is also given.



The problem solving method

Objective: to teach students how to get good ideas

Can we teach students what makes experienced people better ?

Origin: collecting hints

To help students stuck on a problem, we often provide reusable hints.

organizing these hints ⟶ first versions of the method

Improvement: reflecting on the thought process

After solving a task:

- How did I get the right ideas ?
- When did I waste time ?
- What can I do better next time ?

⟶ Update the method accordingly

Finding good ideas of algorithms
Some steps of the method

1) Read the task and Write a simple, synthetic reformulation of the problem.

2) List the dimensions of the problem, and their ranges of values

3) Find the best graphical representations for the task

4) Create useful examples of input and solve them carefully and entirely by hand.

5) Look for a simple working (usually slow) solution, and try optimize it.

6) Simplify the problem, solve each simplified version, and combine the solutions.

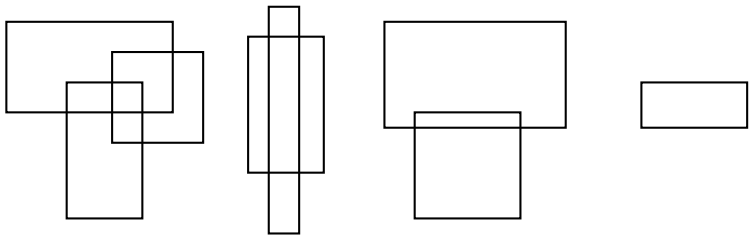
7) Try to look at the task from different points of view (orders, algorithms, …)

Step 2 : List the dimensions, and their ranges of values

Helps to see some properties of the task

Will be used by other steps of the method.

Example :"*Given 10,000 rectangles, give the maximum number of rectangles that have the same surface.*"



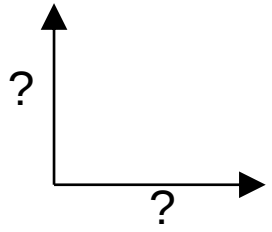
Dimension	Range of values
Input dimensions	
id of a rectangle	[0..9999]
x_1	[0..100,000]
y_1	[0..100,000]
x_2	[0..100,000]
y_2	[0..100,000]
Output dimensions	
Number of rectangles of a given surface	[1..10,000]
Implicit dimensions	
Width of a rectangle	[0..100,000]
Height of a rectangle	[0..100,000]
Surface of a rectangle	[0..10^{10}] (overflow!)

Step 3 : Look for the best graphic representations.

"*Don't stick to your first idea of a graphical representation : try different ways to draw examples*"

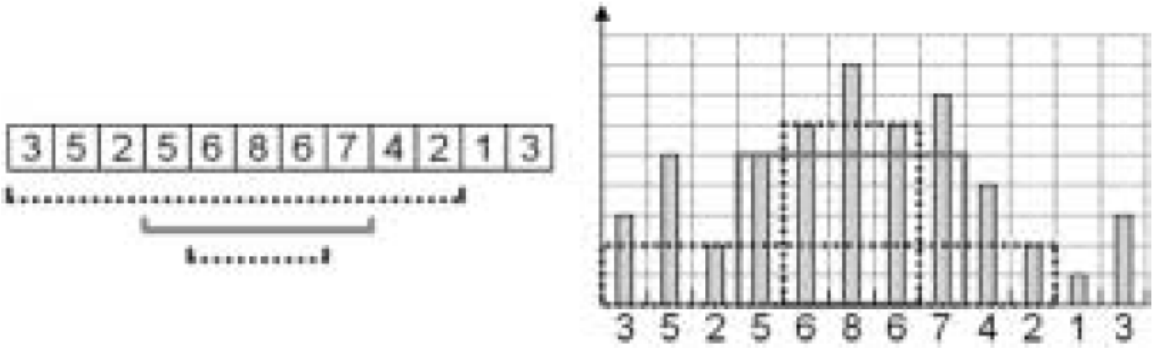
Two dimensions on a piece of paper
=> Two dimensions of the task presented in a very clear ordered way.

Selecting the 2 dimensions of the problem
to map to the axis is essential.



Example : "*Given a sequence of positive integers, find an interval of positions that maximizes:*

length of the interval x smallest value in the interval



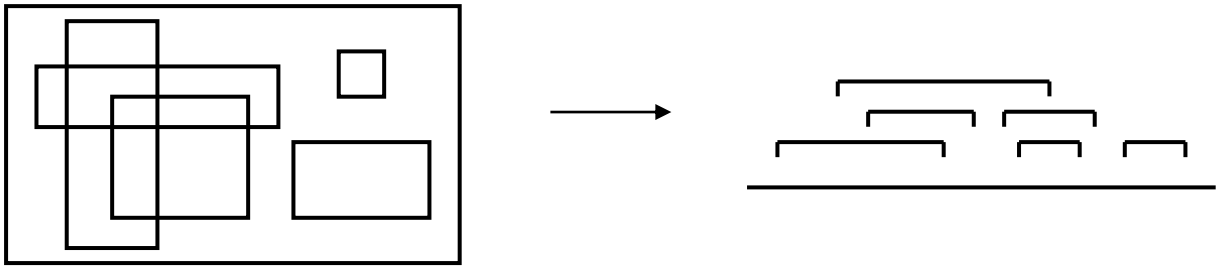
Step 7 : Solve simplified versions of the task

Any algorithm or clever observation required in a simplified solution will most likely be a part of the complete solution.

1) **For each dimension** of the task, try to **simplify the problem** by either:
   (a) removing the dimension completely.
   (b) setting the value for this dimension to a constant
   (c) reducing the range of possible values for that dimension.

2) For each simplification, **restate the corresponding problem clearly** and try to solve it as if it were a completely independent problem.

3) Try to **combine the solutions** of the different simplified versions to get some ideas for the complete problem.



Main steps of the Method : Finding an Idea

Go through these steps until an idea is found :

1) Read the task and Write a simple, synthetic reformulation of the problem.
2) List the dimensions of the problem, and their ranges of values
3) Find the best graphical representations for the task
4) Create useful examples of input and solve them carefully and entirely by hand.
5) Look for a simple working (usually slow) solution, and try optimize it.
6) Simplify the problem, solve each simplified version, and combine the solutions.
7) Try to look at the task from different points of view (orders, algorithms, …)

Apply recursivly on each encountered simplified problem and sub-problem.

Conclusion

The full method includes :

- The set of 7 steps we presented, to find good ideas of algorithms.
- A set of 7 steps to apply on each idea
- Detailed techniques to apply for each of these 14 steps.
- Specific techniques for various domains of algorithmics.

These are taught little by little through various elements of the curriculum:

- As hints given for each task
- By example, within the detailed solutions provided when a task is solved
- Through synthesized dedicated documents.

Having the students apply the method correctly can sometimes be a challenge, but the effort has proven worthwhile.