Informatics for all students
A Computational Thinking Approach

Maciej M. Sysło
University of Wrocław, University of Toruń,
syslo@mat.umk.pl, http://mmsyslo.pl/
Introducing computational thinking:

- Computational thinking as an extension of algorithmic thinking
- Computational thinking as an approach to informatics (computer science) education of all school students.
- Computational thinking in other than informatics subjects (disciplines).

Instead of defining what CT is.
Contents

- School system in Poland and informatics education
- Informatics *versus* ICT
- Computer Science Education in crisis?
- Informatics education – shifts in approach
- Computational thinking (CT)
- Informatics for all high school students: Project Based Learning (PBL), CT, FL – a textbook
- Programming for all students
- Computational thinking in mathematics
The school system in Poland

Pre-school year

1st stage
Pre-school year 7-9

2nd stage

10-12

13-15

16-18

19-18

Tertiary education – University

Upper – high school

Secondary education

Lower – gimnazjum, middle school

Primary education

Informatics for all students, 1h
Informatics adv. – elective, 6h
Informatics (mostly as ICT) with elements of algorithmics

Computer lessons (ICT)

Maciej M. Sysło

Informatics as a mandatory subject has been in the curricula in Poland for last 25 years!!!
Informatics (CS) **versus** ICT

- Informatics (CS) is concerned with **designing** and **creating** informatics ‘products’ and ‘tools’, such as: algorithms, programs, application software, systems, methods, theorems, computers, …

- ICT – applications of CS (computing) – concentrates on how to **use** and **apply** informatics and other information technology tools in working with information; can be also creative
History: 1965 – …
computers in education

1965 … 1985 …

Informatics curricula and teaching – computer science – there was no information technology

beginning of 90’
moves in education:

- computer science → information technology
  i.e.: constructing computer solutions
  → using ready-made tools
  i.e.: computer science for some students
  → information technology for all

- recently: informatics for all – computational thinking
Computer science (education) – in crisis?

Q: Is computer science in crisis? a dying discipline?

A crisis in university computer science (in the US):

- the number of students enrolled in CS has fallen for several years: in 2007 dropped 49% from 2001/2002
- impact on degree „production”: the number of bachelor’s degrees fell 43% between 2003/04 and 2006/07

Similar figures for UK.

In Poland: declining interests in high school informatics,

On the other hand – there is still a demand for experts and specialists in computer use and applications: in the USA – in 2020: 1.5 mln computer scientists needed, but 0,5 graduates
Computer science education in crisis
some answers

A:

- students have tested *enough ICT* in their upbringing and they want something different at Uni level
- the traditional school and university curricula in computing are unattractive to present-day students
- students (not only) do not distinguish between using and studying (computer tools)
- opposed to a vocational qualification, the mission of uni is to develop understanding, rather than skills only

The lack of adequate CS education in high schools
UK: harmful ICT replaced by Comp Sci – 2012

‘Harmful’ ICT curriculum set to be dropped this September to make way for rigorous Computer Science

Education Secretary Michael Gove today announced he was scrapping the existing ICT curriculum.

In its place, he will introduce new courses of study in Computer Science. The move, which is being supported by industry experts including Ian Livingstone – co-founder of Games Workshop, would give schools the freedom to create their own ICT curriculum.
The True 21st Century Literacy is Programming
Informatics education – shifts in approach

- 60’ – 90’: algorithmic thinking: creating programs, algorithmics, programming – there was no ICT
- 90’ – ICT era: step back: basic computer literacy – the capability to use today’s technology
- beginning of 2000: fluency with ICT – the capability to use new technology as it evolves
- J. Wing, 2006: computational thinking – competencies built on the power and limits of computing: 3R + computational thinking

Shift: algorithmic thinking to computational thinking
informatics for informatics to informatics for all
Computational thinking (J. Wing, 2006) in informatics for all

Includes a range of mental tools for problem solving:

- reduction and decomposition of complex problems
- approximation, when exact solution is impossible
- recursion: inductive thinking
- representation and modeling of data or phenomena
- heuristic reasoning (thinking)

The influence on other disciplines – in mathematics: *the purpose of computing is insight not numbers*

[R.W.Hemming]

Applies to all other disciplines

Maciej M. Sysło

IBM, 1924
Computational thinking
old notions, extended meaning

Extended meaning of two notions:

- **a problem** – in a wider context, not necessarily algorithmic – occurs when one has to provide a solution based on what one has learned but is not told how to do it; here – provide a computer solution

- **programming** – giving a computer something to do, since computers only run programs; hence, we have the following ‘programs’: spreadsheet, data base, presentation, website, documents, … ; a program – not necessarily an effect of using a programming language

**Programming should not be confused with coding** – we have programming constructions independent of tools, programming methods, methodology
Changes in the curriculum (2008)

The main Informatics Topic: Problem solving and decision making with a computer by applying algorithmic approach, in particular students are expected to:

- discuss and analyze various problem situations;
- develop and formulate a specification;
- design a solution of a problem by choosing a method and computer tools;
- implement a solution in the form of a computer program (in a programming language, in application software);
- test and evaluate properties of a solution (complexity, correctness);
- present a solution to other students and discuss its applications to other problem situations.
Challenges

In September of 2012:

- How to switch from 2 hours/week of ICT to 1 hour/week of Informatics?
- New teachers are needed, unfortunately teachers of ICT continue to teach in informatics classes
- Training of ICT teachers to teach Informatics – is it possible?
- New textbooks needed
- In fact, new teaching methods needed
Project Based Learning

- various ways to define: what a project is;
- features of a project in education and PBL:
  - should motivate students to work actively
  - collaboration among students
  - opportunity for personalization of learning
  - leadership and other roles
  - „production” of final results and their presentation
  - opportunity to learn new tools
  - solving some real world problems
  - off class activities – flipped learning – a new culture of learning
  - …
We use computational thinking approach

Computational thinking is the thought process involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent (a computer).

Operational definition of CT:

- Formulating problems in a way that enables us to use a computer and other tools to help solve them.
- Logically organizing and analyzing data.
- Representing data through abstractions such as models and simulations.
- Automating solutions through algorithmic thinking (ordered steps).
- Identifying, analyzing, and implementing possible solutions to achieve the most efficient and effective combination of steps and resources.
- Generalizing and transferring this problem solving process to a wide variety of problems.
Responses to Challenges

- How to switch from 2 hours/week of ICT to 1 hour/week of Informatics?
  - Project Based Learning + Flipped Classroom (Learning) strategy
- New teachers are needed, unfortunately teachers of ICT continue to run informatics classes
  - Try to use computational thinking
- Training of ICT teachers to teach Informatics – is it possible?
  - Training in computational thinking
- New text books needed
  - Yes, we have published
- In fact, new teaching methods needed
  - CT, PBL, FC/L,

Flipped learning: a teacher explains the project and helps with problems; students work in and off the classroom
Projects, samples of topics

- **Computer environment**: My e-textbook.
- **Communication and information in the Internet**: a webpage for a project – web styles and templates, my web portfolio; graphics – banner
- **Text documents**: discussion with Umberto Eco – advanced editing of an extended document, a group project (discussion).
- **Information and data representation** – making a quiz
- **Data and their visualization** – use of Internet data, visualization, simulation. A collection of graphical plots of functions.
- **Collecting and analyzing data** – results of sport events, organizing an alumni meeting
- **Simple algorithmic calculations** – representations of numbers (binary, decimal, etc.). Working with election data – analysis of data, abstraction, algorithms, automation, simulation, sorting (bucket), designing and writing programs (members of teams)
Programming for all

- The Hour of **Code**
- **Robots**
- **Scratch** – Mitchel Resnick in **Toruń**
- **Computer** model – RAM
- Inne programy – [mmsyslo.pl](http://mmsyslo.pl)
Programming: RAM

Available at: http://mmsyslo.pl/Materialy/Oprogramowanie
CT: Data representation

- binary representation
- logarithmic length of numbers
- integer *versus* real numbers in computers

\[(111...1)_2 = 2^{k-1} + 2^{k-2} + ... + 2^2 + 2^1 + 2^0 = 2^k - 1\]

It is easy to see that when we add 1 to this binary number we get \(2^k\), the next power of 2, hence we get the last equality above. On the other hand, the smallest integer, which needs \(k\) bits, has only 1 on its most significant position, therefore equals \(2^{k-1}\). Therefore we have the following inequalities:

\[2^{k-1} - 1 < n \leq 2^k - 1.\]

Now, adding 1 to all sides of these inequalities and taking \(\log_2\) of all sides we get the inequalities:

\[k - 1 < \log_2 (n + 1) \leq k.\]

Since the number of bits \(k\) is an integer number, we have:

\[k = \lceil \log_2 (n + 1) \rceil,\]
CT: Recursion and logarithm

Compute (RSA):

\[ x^{12345678912345678912345678912345} \]

School mathematics:

\[ x^n = x \cdot x \cdot x \cdot \ldots \cdot x \]

\( n - 1 \) multiplications: 12345678912345678912345678912344

Supercomputer \( 10^{15} = 1\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ 000\ \text{oper/sek} \)

It will take: \( 3 \times 10^8 \) years!
CT: Recursion and logarithm

Power(x,n) \quad \{ \, x^n \, \}

if n=1 then Power:=x
else if n - even then
    Power:=Power (x,n/2)^2 \quad \{ x^n = (x^{n/2})^2 \}
else Power:=Power(x,n-1)*x \quad \{ x^n = (x^{n-1}) \cdot x \}

Number of operations:

- number of bits in the binary representation of n – \log_2 n
- plus
- number of 1’s in the binary representation of n – \log_2 n

Total: at most 2*\log_2 n

For

x^{12345678901234567890123456789012345...}

only .... 200 multiplications

<table>
<thead>
<tr>
<th>m</th>
<th>\log_2 m</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^4</td>
<td>10</td>
</tr>
<tr>
<td>10^6</td>
<td>20</td>
</tr>
<tr>
<td>10^9</td>
<td>30</td>
</tr>
<tr>
<td>10^{12}</td>
<td>40</td>
</tr>
<tr>
<td>10^{20}</td>
<td>66,5</td>
</tr>
<tr>
<td>10^{50}</td>
<td>166</td>
</tr>
<tr>
<td>10^{100}</td>
<td>332,2</td>
</tr>
</tbody>
</table>

Maciej M. Sysło
CT: Approximation

\[ \sqrt{a} : \quad a \quad a/x \]

\[ x \]

\[ x' \leftarrow \frac{(x + a/x)}{2} \]

\[ a \]

\[ x' \]
CT: Approximation

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Calculation of a square root of $a$ =</td>
<td>$2,000000000000000000000000$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Exact value of a square root of $a$ =</td>
<td>$1,414213562373100000000$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Initial approximation of $x$ =</td>
<td>$2,000000000000000000000000$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Iteration</td>
<td>Successive approximation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Initial</td>
<td>$2,000000000000000000000000$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>$1,500000000000000000000000$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>$1,416666666666666667000000$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>$1,414215686274510000000000$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>$1,414213562374690000000000$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>$1,414213562373090000000000$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>$1,414213562373090000000000$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>$1,414213562373090000000000$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>$1,414213562373090000000000$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9</td>
<td>$1,414213562373090000000000$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>$1,414213562373090000000000$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>=PIERWIASTEK($D$2)</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Krok algorytmu</th>
<th>Wartość przybliżenia</th>
</tr>
</thead>
<tbody>
<tr>
<td>Początkowy</td>
<td>=D6</td>
</tr>
<tr>
<td>1</td>
<td>=(D10+$D$2/D10)/2</td>
</tr>
<tr>
<td>=C11+1</td>
<td>=(D11+$D$2/D11)/2</td>
</tr>
<tr>
<td>=C12+1</td>
<td>=(D12+$D$2/D12)/2</td>
</tr>
<tr>
<td>=C13+1</td>
<td>=(D13+$D$2/D13)/2</td>
</tr>
<tr>
<td>=C14+1</td>
<td>=(D14+$D$2/D14)/2</td>
</tr>
<tr>
<td>=C15+1</td>
<td>=(D15+$D$2/D15)/2</td>
</tr>
<tr>
<td>=C16+1</td>
<td>=(D16+$D$2/D16)/2</td>
</tr>
<tr>
<td>=C17+1</td>
<td>=(D17+$D$2/D17)/2</td>
</tr>
<tr>
<td>=C18+1</td>
<td>=(D18+$D$2/D18)/2</td>
</tr>
<tr>
<td>=C19+1</td>
<td>=(D19+$D$2/D19)/2</td>
</tr>
</tbody>
</table>
CT: Reduction

We know, how to find min/max among $n$ numbers – $(n - 1)$ oper.

• min and max at the same time
• selection sort

**Problem.** Let set $A$ contain a large amount of integers, e.g. 10 millions. Verify if each triple of numbers from $A$ can be the lengths of sides of a triangle [4].

It is obvious that one has to test if each triple of numbers $a, b, c$ satisfies the triangle condition, i.e. three inequalities:

$$a + b > c; \quad a + c > b; \quad b + c > a$$

$$min_1 := \infty; \quad min_2 := \infty; \quad max := - \infty;$$

while there is new data $x$ in $A$ do begin

if $x < min_2$ then

if $x < min_1$ then begin $min_2 := min_1; \quad min_1 := x$ end

else $min_2 := x$

else if $max < x$ then $max := x$

end
Recursive thinking in life situations

Dance

\[
\text{Dance;}
\]
\[
\text{if no music then STOP}
\]
\[
\text{else}
\]
\[
\text{make a step;}
\]
\[
\text{Dance}
\]
CT: Heuristics

- Change making problem
- Knapsack problems
- Shortest path (cycle) problems
Bóbr 2013:Junior

Najkrótsze drogi

Bóbr lubi się bawić w parku. Jego dom (S) i park (G) są połączone układem mostów zbudowanych z belek o tej samej długości, jak poniżej:

Nielety skrzyżowania oznaczone symbolem X zostały zniszczone.

Ile jest najkrótszych dróg z jego domu do parku?

- 12
- 14
- 16
- 18
TRAFFIC

Here is a map of a system of roads that links the suburbs within a city. The map shows the travel time in minutes at 7:00 am on each section of road. You can add a road to your route by clicking on it. Clicking on a road highlights the road and adds the time to the Total Time box.

You can remove a road from your route by clicking on it again. You can use the RESET button to remove all roads from your route.

Question 1: TRAFFIC CP007Q01
Pepe is at Sakharov and wants to travel to Emerald. He wants

- 20 minutes
- 21 minutes
- 24 minutes
- 28 minutes
Problem: Find a plan for a visit of the President of Poland in all regions (Travelling Salesman Problem - TSP)

Solution:

\[15 \times 14 \times 13 \times 12 \times 11 \times \ldots \times 2 \times 1 = 15!\]

In the USA: \[48 \times 47 \times 46 \times \ldots \times 2 \times 1 = 48!\]
For a supercomputer, 1 PFlops – $10^{15}$ operations/second

$15! = 1307674368000/10^{15}$ sek. = 0.01 sek.

$48! = 1,241391559253607267086228047373*10^{61}/10^{15} = 3*10^{38}$ years
TSP – a greedy approach

Optimal road

bad choice of a greedy method
What next?

Road map for 2014-2015:

- proposition of changes to the National Core Curriculum on informatics (computer science) including programming for all students in K-12 – to be accepted by the Ministry
- introduction of teachers preparation standards in informatics (computer science), then used by teachers and by teachers preparation institutions
- new centers for teachers preparation (blended learning)
- systematic evaluation of teachers (in schools) and teachers preparation institutions (universities, in-service courses)
- pilot projects with teachers, students, schools – public and non-public institutions
WCCE – learning while we are connected

Key speakers:

- **George Siemens** – connectivism
- **Mitchel Resnick** – Scratch
- **Włodzisław Duch (UMK)** – mind and education
- **Flipped school** – by a couple of teachers from the US
- **Vice president of Intela, John A. Davies** – mobile learning
- **Pearson representative**

Recordings of talks will be available in a week on the conference website: [http://wcce2013.umk.pl/](http://wcce2013.umk.pl/)
Thank you for your attention
syslo@ii.uni.wroc.pl