



Jak zainteresować ucznia matematyką

Maciej M. Sysło
UWr, UMK, WWSI
syslo@ii.uni.wroc.pl

NA RATUNEK UCZĄCYM SIĘ MATEMATYKI W SZKOŁACH JAK MOGLIBYŚMY SIĘ UCZYĆ

Maciej M. Sysło

syslo@ii.uni.wroc.pl <http://mmsyslo.pl>

- Raport NIKu:** malejące z latami słupki osiągnięć uczniów –
konkluzja: **zawiesić maturę z matematyki**
- Diagnoza:** ucząc wszystkich uczniów tego samego, nawet
w różny sposób, ale egzaminując z tego samego,
tych słupków nie ruszymy i ich nie nauczymy
- Izolacja mat.** uczymy klasyki matematyki dla klasyki, **brak spojrzenia**
przed siebie, **wokół uczniów**, na ich zainteresowania
matematyka i informatyka: małżeństwo czy rozwód?

Uczeń – priorytetem edukacji

Ustawa o Oświacie: szkoła winna zapewnić każdemu uczniowi warunki niezbędne do jego rozwoju

Każdy człowiek, każdy uczeń jest inny
lub inaczej, każdy mózg jest inaczej okablowany

stąd, głównym priorytetem edukacji powinna być:
indywidualizacja (personalizacja)

Jak spowodować, by uczeń mógł po latach powiedzieć:

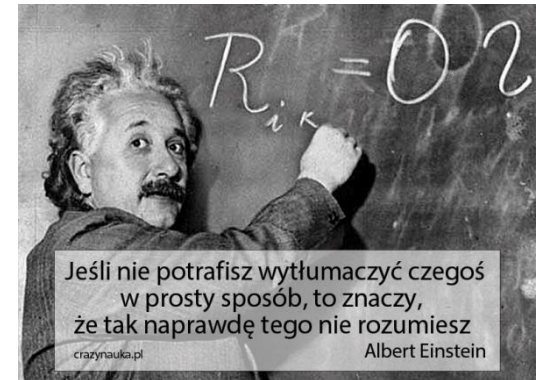
I have never let my schooling
Interfere with my education Nigdy nie dopuściłem, by chodzenie do
szkoły kolidowało z moim (wy)kształceniem

[Mark Twain, XIX w.]

Uwaga: Personalizacja nie jest pomysłem dzisiejszej technologii

Wielcy o edukacji

Everybody is a GENIUS
but if you judge a fish
by its ability to climb a tree
it will live its whole life
believing that it is STUPID



W edukacji: I'm a very big believer in:

equal opportunity

as opposed to equal outcome

równe szanse, w przeciwieństwie

do jednakowych rezultatów (do wyrównywania szans)



Sir Ken Robinson:

indywidualna podstawa programowa
dla każdego ucznia, ale ...
przyjdą politycy i wyrównają



Propozycje

- **podstawa programowa** jako **ogólny standard** kształcenia, np. „uczeń rozwiązuje nierówności”, ale już bez „nie trudniejsze/latwiejsze niż...”
- **programy nauczania „różnych matematyk”**, np. codzienna, algebra, odkrywanie matematyki, dociekliwość matematyczna, geometria
- **mariaż z informatyką**, źródłem zastosowań i wsparcia, czyli **motywacji**
- uwzględnienie elementów teorii uczenia się: **konstrukcjonizmu (Papert)**, **spiralnego rozwoju (Bruner)**, **myślenia komputacyjnego (Papert, Wing)**
- **egzamin/matura**, jeśli już, to dostosowane do nauczanego zakresu

Nauczyciele – najważniejsza technologia w edukacji

- studia nauczycielskie: matematyka i informatyka – **to powiązanie przyciągnie przyszłych nauczycieli matematyki**
- **ale ...** uprawnienia pedagogiczne i dydaktyczne **nie dodatkiem** do studiów matematycznych czy informatycznych
- nacisk na pracę z **uczniem** o różnorodnym „okablowaniu mózgu”

Dlaczego nauczanie matematyki chce się obejść bez informatyki?

Myślenie komputacyjne w kształceniu matematycznym

Sytuacja staje się coraz poważniejsza dla dalszych efektów kształcenia matematycznego. Coraz trudniej odizolować cele nauczania matematyki od sposobów rozwiązywania problemów z różnych dziedzin, w coraz większym stopniu związanych z myśleniem komputacyjnym i programowaniem. **Utrzymywanie dalej aktualnego zakresu i sposobu nauczania matematyki oraz zignorowanie możliwości, jakie się wyłaniają, powoduje już teraz, a nasili się jeszcze bardziej w najbliższej przyszłości, izolację szkolnej matematyki od tego, co uczniowie poznają na innych przedmiotach, w szczególności na informatyce, a także poza szkołą, w środowisku nowych technologii.** Ten zastój w rozwoju zakresu kształcenia matematycznego wstrzymuje także dalszy profesjonalny rozwój nauczycieli matematyki, którzy mogliby i powinni wspierać opisane zmiany, będąc jednocześnie ich ambasadorami.

Motto

Celem obliczeń nie są liczby, a zrozumienie

The purpose of computing is insight not numbers

[R.W. Hemming, 1959]

Note: w 1959 roku obliczenia były głównie na papierze

Traktujemy tutaj computing, nie tylko jako działania wokół komputerów, ale jako **procesy myślowe – myślenie komputacyjne**

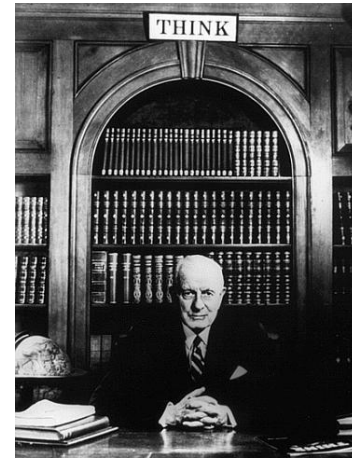
Celem programowania nie są programy, ale abstrakcja

The purpose of programming is abstraction not programs

Mądrością staje się **symbioza** tego,
w czym **mózg** jest najlepszy,
z tym, co **komputer** potrafi wykonać nawet lepiej

[Mark Prensky, 2013]

Ta symbioza wymaga jednak **mózgu**, myślenia,
a więc będzie tutaj o **myśleniu**:



Wykształcona małpa (Educated Monkey)

Czyli odmałpienie tabliczki mnożenia

Służy do:

- mnożenia dwóch liczb
- dzielenia dwóch liczb
- rozkładu liczby na czynniki

Z podkładką, może służyć do dodawania

Pojęcia:

- podstawowe operacje matematyczne,
- posługiwanie urządzeniami do liczenia – elektroniczny kalkulator później, tutaj widać, jak są wykonywane działania
- algorytm

Dzieci były zainteresowane, gdzie można kupić takie urządzenie !!!!



1916

Dla 5 x 5

<https://www.youtube.com/watch?v=mSk6bhBIYXQ>

Wady w uczeniu matematyki

- Wady w nauczaniu matematyki w szkołach, jak i w informatyce, a więc wywodzące się z informatyki, mogą pomóc studentom:
- uwaga skupiona
 - wzory jako interpretacja
 - algorytmy
 - a więc, brak
 - brak rzeczy
- Niektóre *mental tools* zaliczane do myślenia komputacyjnego, mogą pomóc studentom:
- skupić **uwagę na problemie**, niż na obliczeniach
 - rozwinąć **kreatywne myślenie** w matematyce posługując się *CT mental tools*
 - wyjść **poza wzory**
 - **tworzyć algorytmy** zamiast korzystać z czarnych skrzynek
 - **stosować w praktyce** pojęcia i metody matematyczne
- stąd brak motywacji u uczniów

Nieco odmienne traktowanie **algorytmów** w matematyce i informatyce.

Matematyka

- (W.W. Sawyer): Matematyka, to ... skrzynka z narzędziami
narzędzie = algorytm = zamknięty schemat

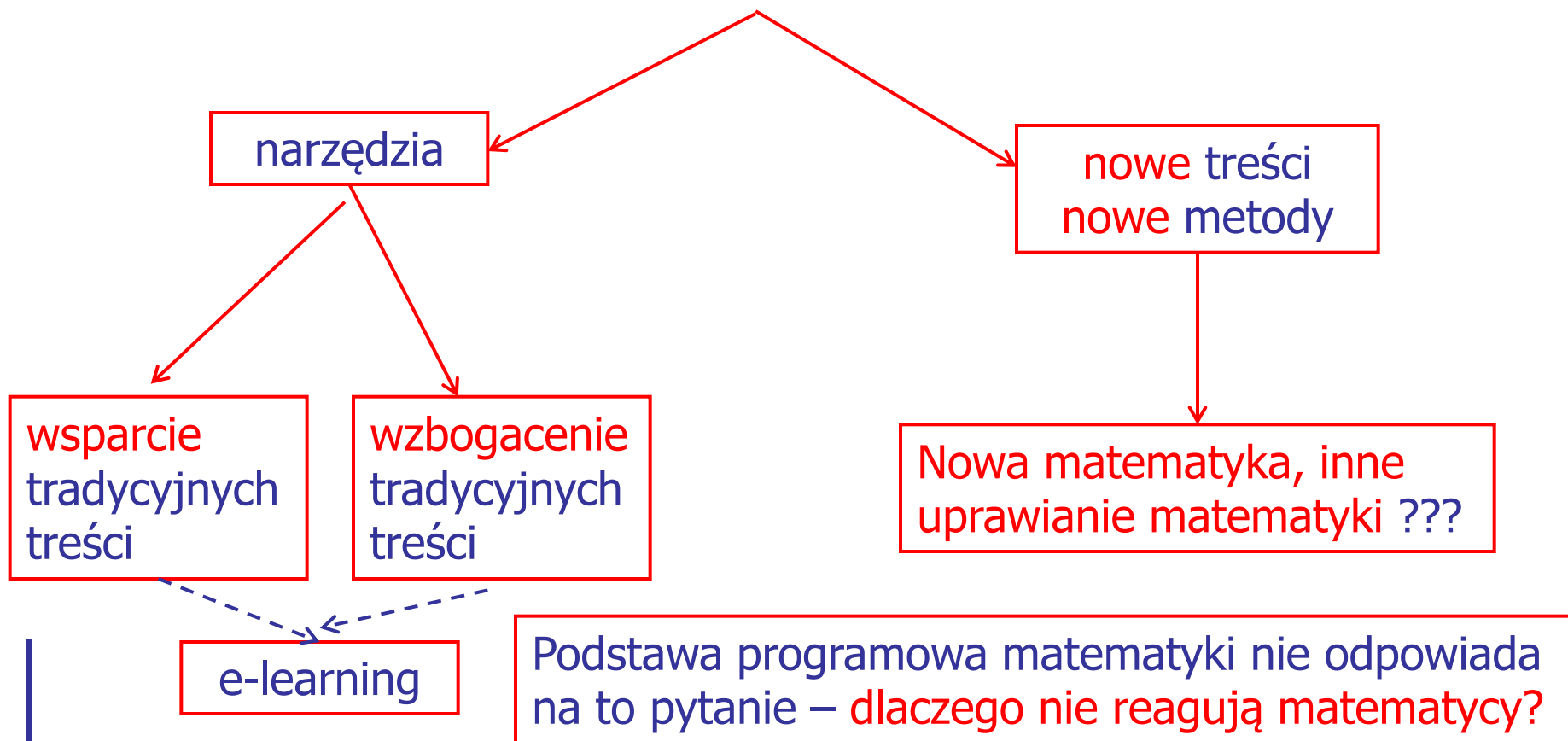
Informatyka

- **Algorytm**: celem działania, **obiekt twórczości**, rozwoju, pielęgnacji, ...

Szansa dla matematyki – **nowe spojrzenie płynące z informatyki**

Komputery a matematyka (szkolna/uczelniana)

W jakim stopniu (i czy?) informatyka, komputery i technologia powinna zmienić matematykę i jej nauczanie?



Podstawa programowa dla informatyki – K-12

Wspólne Cele kształcenia – Wymagania ogólne – dla wszystkich etapów

- I. **Rozumienie, analizowanie i rozwiązywanie problemów** na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.
- II. **Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych**: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi. ← **Technologia**
- III. **Posługiwanie się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi**, w tym: znajomość zasad działania urządzeń cyfrowych i sieci komputerowych oraz wykonywania obliczeń i programów.
- IV. **Rozwijanie kompetencji społecznych**, takich jak: komunikacja i współpraca w grupie w tym w środowiskach wirtualnych, udział w projektach zespołowych oraz organizacja i zarządzanie projektami.
- V. **Przestrzeganie prawa i zasad bezpieczeństwa**. Respektowanie prywatności informacji i ochrony danych, netykiety, norm współżycia społecznego, praw własności intelektualnej; ocena i uwzględnienie zagrożeń, związanych z technologią.

Spiralna realizacja na kolejnych etapach
Spiralny rozwój zasobów (J. Bruner)

Myślenie komputacyjne



J. Wing, 2006: użyteczne postawy i umiejętności, jakie każdy, nie tylko informatyk, także matematyk powinien starać się wykształcić i stosować

to procesy myślowe angażowane w formułowanie problemu i przedstawianie jego rozwiązań w taki sposób, aby komputer – człowiek lub maszyna – mógł skutecznie wykonać.

Umiejętności składające się na myślenie komputacyjne:

- **abstrakcja, redukcja i dekompozycja** złożonych problemów
- **tworzenie przybliżonych rozwiązań (aproksymacji)**, gdy dokładne rozwiązanie nie jest możliwe
- **stosowanie rekurencji**, czyli myślenia indukcyjnego (rekurencja = iteracja)
- **tworzenie reprezentacji i modelowania** danych, problemów i rozwiązań
- **stosowanie heurystyk** (G. Polya, *Jak to rozwiązać*)
- **uogólnianie**

Redukcja: matematyk a informatyk

Uczeń poznaje/zna (IV – VI klasa) liniowy algorytm do:

- znalezienia min lub max wśród n liczb – $(n - 1)$ operacji

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:

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

$O(n^3)$

Informatyk: $\min_1 A + \min_2 A > \max A$

```
min1 := ∞; min2 := ∞; max := - ∞;  
while there is new data x in A do begin  
  if x < min2 then  
    if x < min1 then begin min2 := min1; min1 := x end  
    else min2 := x  
  else if max < x then max := x  
end
```

$O(n)$

Liniowy, prosty, elegancki algorytm

Sudoku – od klasy 1 ...

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 ...

Proste Sudoku – kwadrat łaciński

Uzpełnij poniższy kwadrat tak, aby każda figura występowała dokładnie raz w każdym wierszu i w każdej kolumnie – jest to tak zwany kwadrat łaciński.

✓

★ ★ ★

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 ...

Proste Sudoku – kwadrat łaciński

Uzpełnij poniższy kwadrat tak, aby każda figura występowała dokładnie raz w każdym wierszu i w każdej kolumnie – jest to tak zwany kwadrat łaciński.

✓

★ ★ ★

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 ...

Sudoku ze zwierzętami

Rozwiąż poniższe Sudoku – w każdym wierszu, w każdej kolumnie i w każdym zaznaczonym kwadracie 2 x 2 powinny się znaleźć różne zwierzęta.

✓

★ ★ ★

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 ...

Sudoku III

Rozwiąż następujące Sudoku – w każdym wierszu, w każdej kolumnie i w każdym zaznaczonym kwadracie 3 x 3 powinny się znaleźć wszystkie liczby od 1 do 9 i to tylko raz.

2		4	1	5		8	9	3
9	1		6	8	4	7	2	5
7	8	9	2	3	1			4
5	8		7	1	9		3	6
1	4	6	2	3		5	7	9
	7	9	5	4	6		1	8
8	2		3	6	5		4	1
4	3	1	8	9		6	5	7
6	9		4	7	1	3		2

✓

★ ★ ★

Pojęcia:

- abstrakcja – nie ma znaczenia, co układamy
- reguła prowadzi do dekompozycja,
- rozkład zadania/problemu,
- praca krokowa – algorytm – kolejność uzupełnień
- uogólnienie

Tok zajęć

SP: Klasy 1-3

Praca z rzeczywistą sytuacją problemową?

- zebranie informacji i danych – **abstrakcja** na niskim poziomie, nie wszystkie dane są potrzebne
- wykrywanie w danych dla problemu
- analiza danych i **reprezentacja** danych (lista, tabela, tabele powiązane)
- **dekompozycja** danych i/lub **problemu**
- projektowanie algorytmu – modelowanie
 - podejście ad hoc – **heurystyka**
 - **metody/algorytmy informatyczne**: algorytm liniowy, metoda warunków rekurencyjnych
 - współbieżność, interakcja – zdarzenia
 - automatyzacja rozwiązania – program
- **symulacja komputerowego/computerowego**
 - testowanie i poprawianie
 - głębsza analiza problemu oraz modyfikowanie

Sytuacja: na podłodze obrazki zwierząt

Problem: znajdź najłżejszego ptaka

Dane: selekcja/wybór ptaków – kura ptakiem? lata? – **abstrakcja**

Zasada w danych: lata (z pomocą komputera):

Reprezentacja danych: ptaki w rzędzie, w jakiejś kolejności (**lista**, **ciąg**)

Dekompozycja: np. na domowe i inne

Algorytm: losowy wybór, systematyczny algorytm: przeglądanie liniowe – **abstrakcja**, bo liczy się tylko waga

Modyfikacje: danych – inne zwierzęta, metody: **uporządkuj** od najłżejszych

Komputer, program: **projekt** w Scratchu, w innym języku – **automatyzacja** sytuacji **wyabstrahowanej** – liczby



Tok zajęć: elementy myślenia komputacyjnego

LO, klasy 1-3

Praca z rzeczywistą sytuacją p

- zebranie informacji i danych – wszystkie dane są potrzebne
- wykrywanie w danych dla problem
- analiza danych i **reprezentacja** lista, tabela, tabele powiązane
- **dekompozycja** danych i/lub pr
- projektowanie algorytmu – mo
 - podejście ad hoc – **heurystyka**
 - **metody/algorytmy informatyczne**
 - współbieżność, interakcja – zd
 - automatyzacja rozwiązania – p
- **symulacja komputerowego mo**
 - testowanie i poprawianie
 - głębsza analiza problemu oraz

Projekt: polemika z wypowiedzią Umberta Eco: *jeśli ktoś myśli, że książka zniknie, to się myli*

Sytuacja: teksty drukowane i elektroniczne związane z tematem

Problem: znajdź argumenty za i przeciw; przeprowadź dyskusję, spisz ją

Dane: selekcja/wyбір fragmentów ze źródeł – **abstrakcja**

Zasada w danych: dotyczą książki, w tym Eco

Reprezentacja danych: **szablon** tekstu dyskusji z przeciwnymi argumentami (**tabela, format**)

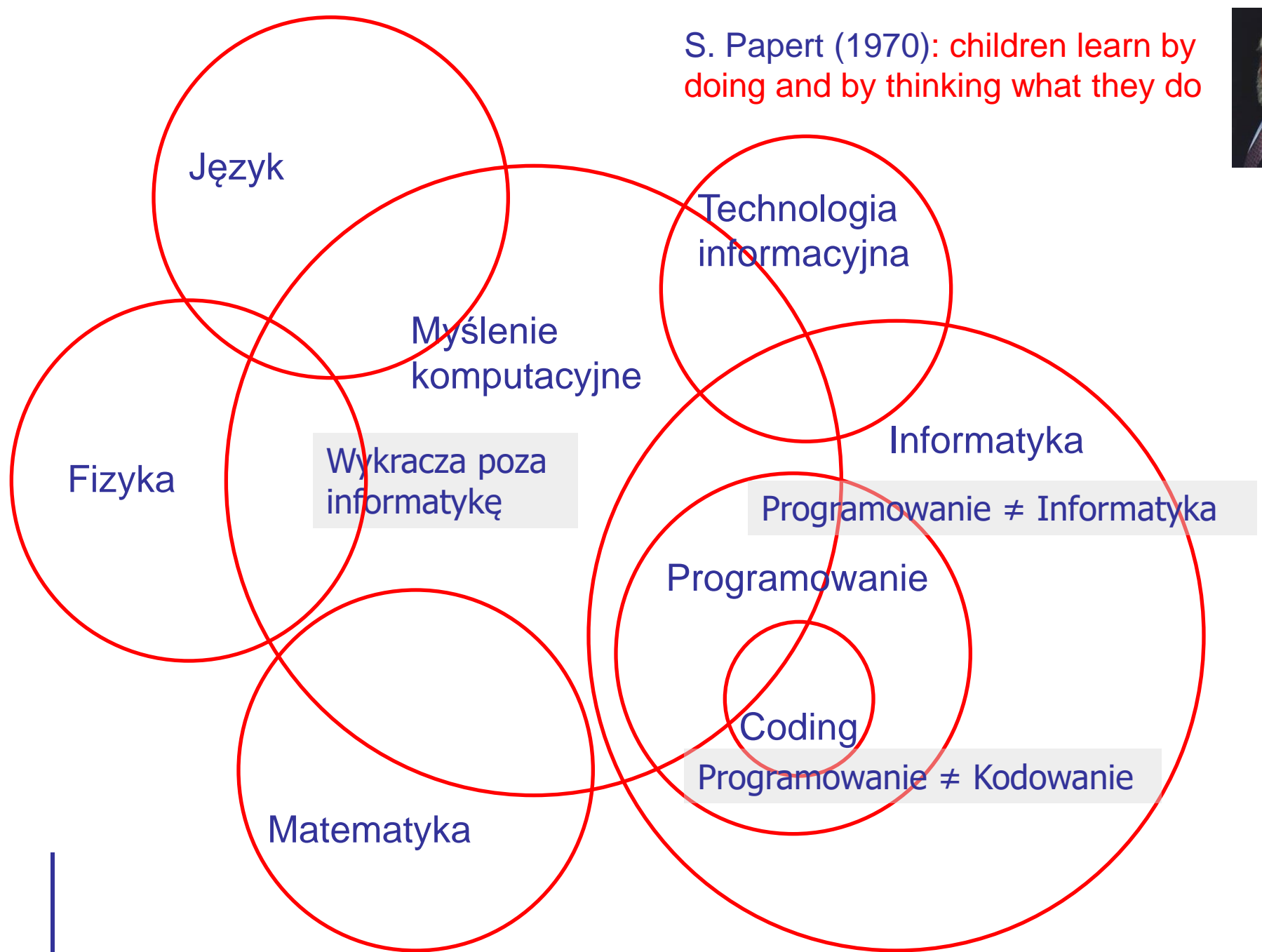
Dekompozycja: argumenty „za” i „przeciw”

Algorytm: metoda postępowania: zorganizowana dyskusja, uporządkowany zapis

Modyfikacje: danych – ważny dla konkluzji głos innych osób – uzasadnienie, włączenie

Komputer, program: **projekt** – **automatyzacja** przebiegu projektu, **ocena własna i nauczyciela, udostępnienie** w systemie prowadzenia projektów

S. Papert (1970): children learn by doing and by thinking what they do



Rozumienie języka – przykład (USA)

Please expand:

Rozwiń:

$$(a + b)^3$$

$$(a + b)^3$$
$$(a + b)^3$$
$$(a + b)^3$$

Kilka powiązań: matematyka a informatyka

- **pojęcie funkcji** – brak na matematyce (K-8) od początku na informatyce
- wartość pierwiastka – przykład **modelowania obliczeniowego**, matematycznego
- potęgowanie – **rekurencja, efektywność**
- NWD, algorytm Euklidesa – **logarytm**
- geometria a **grafy**

Funkcja

Brak tego pojęcia w **podstawie matematyki I – VIII klasa !!!**

W warunkach i sposobach realizacji podstawy:

... operowanie wykresami zależności pozwala na intuicyjne opanowanie **trudnych i abstrakcyjnych pojęć takich jak funkcja**, monotoniczność, ekstrema, przy użyciu minimalnej wiedzy matematycznej (**nie należy wprowadzać tych pojęć w szkole podstawowej**).

Ale funkcja, to także sposób obliczania jej wartości (S. Banach) – **algorytm** I tej dodatkowej wiedzy dostarcza połączenie **matematyka+informatyka**

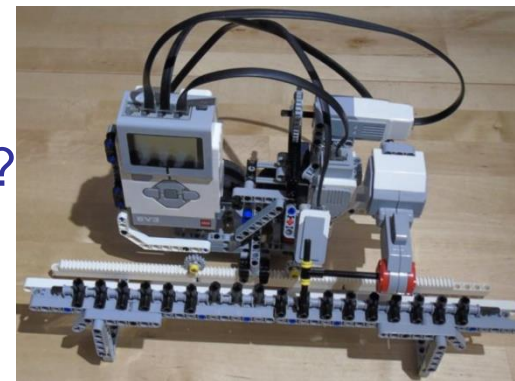
Specyfikacja problemu/algorytmu.

Dane: dziedzina problemu, algorytmu

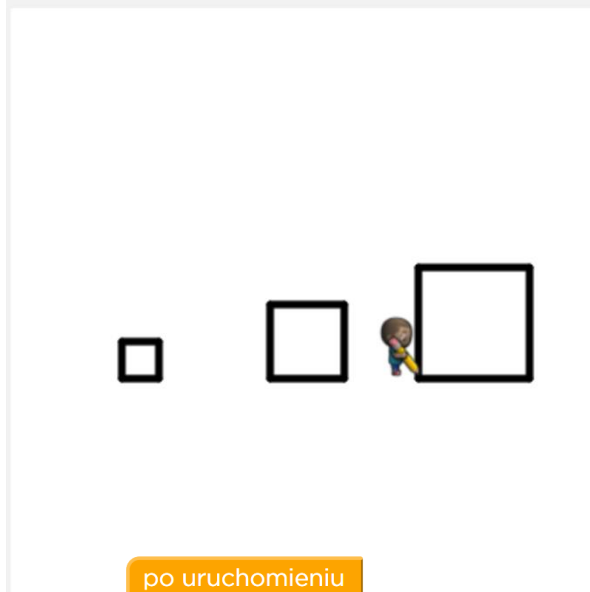
Wynik: przeciwdziedzina algorytmu

A co to jest **algorytm**?

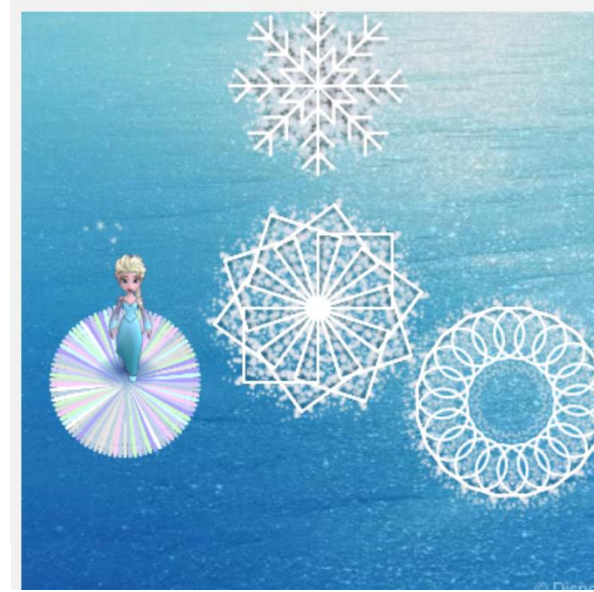
Czy potrzebna jest **maszyna Turinga**? **Mamy ją!**



Funkcja w programowaniu: Godzina kodowania, klasy 4-6



```
po uruchomieniu
narysuj kwadrat 25
skocz do przodu o 100 piksele
narysuj kwadrat 50
skocz do przodu o 100 piksele
narysuj kwadrat 75
```



```
po uruchomieniu
skocz do tyłu o 150 piksele
utwórz śnieżynkę typu fraktal
skocz do przodu o 150 piksele
utwórz śnieżynkę typu kwadrat
skręć w lewo o 90 stopnie
skocz do przodu o 120 piksele
utwórz śnieżynkę typu spirala
skręć w prawo o 180 stopnie
skocz do przodu o 250 piksele
skręć w lewo o 90 stopnie
idź do przodu 50 piksele
utwórz śnieżynkę typu linia
```

Potęgowanie

W podstawie matematyki, VII – VIII klasa:

1. Potęgi o podstawach wymiernych. Uczeń:

- 1) zapisuje iloczyn jednakowych czynników w postaci potęgi o wykładniku całkowitym dodatnim;
- 2) mnoży i dzieli potęgi o wykładnikach całkowitych dodatnich;
- 3) mnoży potęgi o różnych podstawach i jednakowych wykładnikach;
- 4) podnosi potęgę do potęgi;

Nie ma przykładów! **Po co są te operacje?**

A może przykład praktyczny? Kryptografia RSA

Potęgowanie w praktyce

Oblicz (RSA):

$$x^{123456789123456789123456788912345}$$

Na matematyce:

$$x^n = x \cdot x \cdot x \cdot \dots \cdot x$$

$n - 1$ mnożeń: 12345678912345678912345678912344

Superkomputer $10^{15} = 1\,000\,000\,000\,000\,000\,000$ oper/sek

12345678912345678,912345678912344 · sekund

12345678912345678,912345678912344/60 = 205761315205761,31520576131520567 · minut

205761315205761,31520576131520567/60 = 3429355253429,3552534293552534278 · godzin

3429355253429,3552534293552534278/24 = 142889802226,22313555955646889282 · dób

142889802226,22313555955646889282/365 = 391478910,20883050838234649011733 · lat

To będzie trwało $3 \cdot 10^8$ lat

Rozwiązanie informatyczne

```
Power(x, n) { x^n }
```

```
  if n=1 then Power:=x
```

```
  else if n - even then
```

```
    Power:=Power(x, n/2) ^2
```

```
    else Power:=Power(x, n-1) *x
```

$$x^{22} = (x^{11})^2 = (x^{10}x)^2 = ((x^5)^2x)^2 = (((x^2)^2x)^2x)^2:$$

$$\{x^n = (x^{n/2})^2\}$$

$$\{x^n = (x^{n-1})x\}$$

$$x^2, x^4, x^5, x^{10}, x^{11}, x^{22}$$

6 mnożeń zamiast 21

Liczba operacji: :

liczba bitów w reprezentacji binarnej $n - \log_2 n$

plus

liczba 1-ek w binarnej reprezentacji $n - \log_2 n$

Razem co najwyżej $2 \cdot \log_2 n$ operacji

Dla

$x^{12345678901234567890123456789012345}$

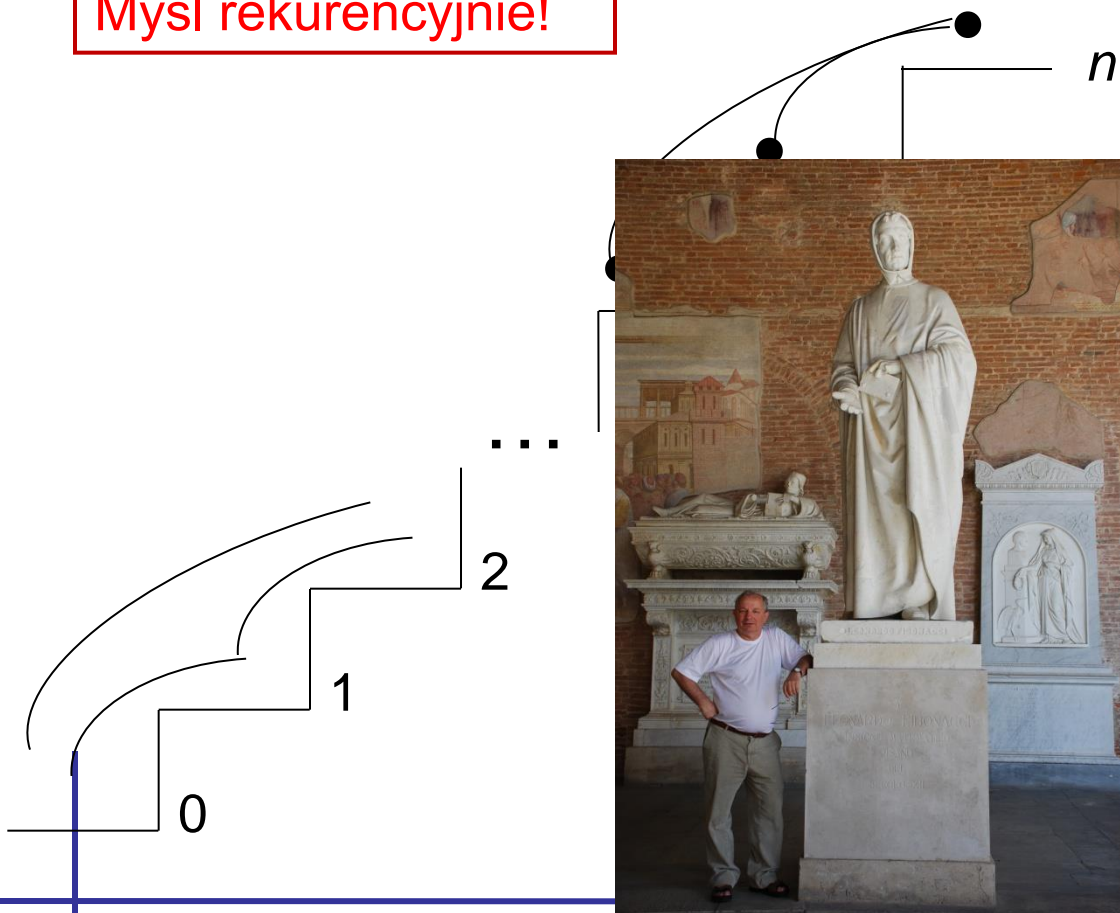
Tylko 200 mnożeń – SZOK!

m	$\log_2 m$
10^4	10
10^6	20
10^9	30
10^{12}	40
10^{20}	66,5
10^{50}	166
10^{100}	332,2

Myślenie rekurencyjne: stary problem

$s(n)$ – liczba sposobów osiągnięcia schodka n

Myśl rekurencyjnie!

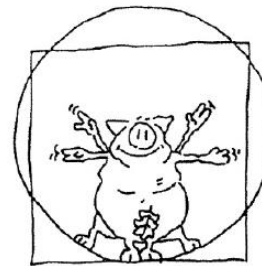


Profesor S. bierze jeden lub dwa schodki – na ile sposobów wyjdzie na piętro n

$s(n-1) + s(n-2)$ dla $n > 2$

Króliki? Ależ to nierealistyczna sytuacja, a taki profesor to codzienność!

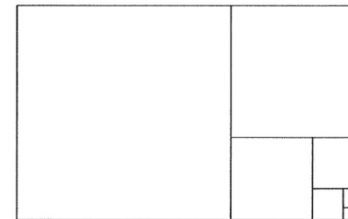
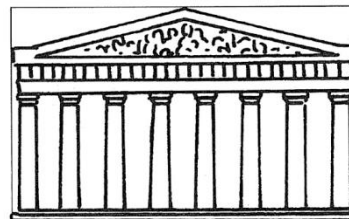
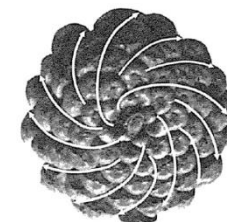
Stary problem, nowe spojrzenie



Nobody is perfect

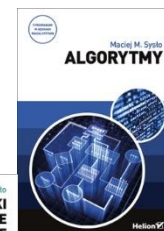
Liczby Fibonacciego – w różnych dziedzinach – myślenie komputacyjne

- **w przyrodzie:** szyszki, słonecznik: skąd? Alan Turing
- **w przyrodzie:** kształt muszli
- **w przyrodzie:** wymiary człowieka
- **w architekturze** – złota proporcja
- konstrukcja prostokąta za pomocą liczb Fibonacciego – złoty prostokąt



Związek liczb Fibonacciego z doskonałością?

Ukazały się w Helionie

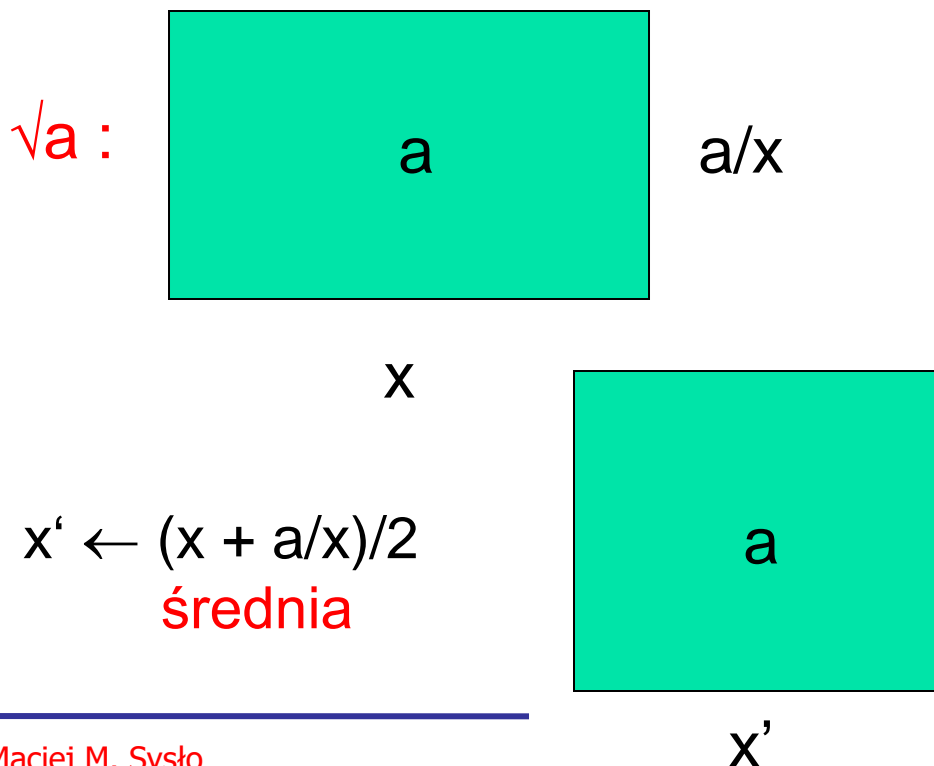


Przybliżone rozwiązania – przykład

Obliczanie wartości pierwiastka kwadratowego z 2, czyli $\sqrt{2} = x$

- komputer nie zna takiej operacji.
- 2 nie jest dokładnie kwadratem żadnej liczby x .

Przybliżamy więc x (dla dowolnego a zamiast 2):



Rozumowanie – *mental tools*:

- abstrakcja
- reprezentacja – przybliżenie
- modelowanie geometryczne
- 3- i wyższe wymiary
- redukcja (do znanego)

Na matematyce, jest to czarna skrzynka.

Przybliżone rozwiązanie

Arkusz jako narzędzie do projektowania i eksperymentowania z algorytmami

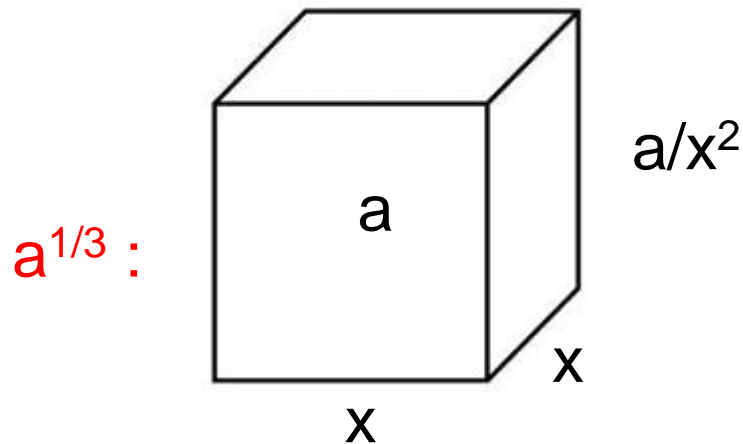
A	B	C	D
Calculation of a square root of a =		2,000000000000000000000000	
Exact value of a square root of a =		1,41421356237310000000	
Initial approximation of x =		2,000000000000000000000000	
		Iteration	Successive approximation
		Initial	2,000000000000000000000000
		1	1,500000000000000000000000
		2	1,4166666666666670000000
		3	1,4142156862745100000000
		4	1,4142135623746900000000
		5	1,4142135623730900000000
		6	1,4142135623730900000000
		7	1,4142135623730900000000
		8	1,4142135623730900000000
		9	1,4142135623730900000000
		10	1,4142135623730900000000

C	D
	2
	=PIERWIASTEK(\$D\$2)
	2
Krok algorytmu	Wartość przybliżenia
Początkowy	=D6
1	=(D10+\$D\$2/D10)/2
=C11+1	=(D11+\$D\$2/D11)/2
=C12+1	=(D12+\$D\$2/D12)/2
=C13+1	=(D13+\$D\$2/D13)/2
=C14+1	=(D14+\$D\$2/D14)/2
=C15+1	=(D15+\$D\$2/D15)/2
=C16+1	=(D16+\$D\$2/D16)/2
=C17+1	=(D17+\$D\$2/D17)/2
=C18+1	=(D18+\$D\$2/D18)/2
=C19+1	=(D19+\$D\$2/D19)/2

Analityczna złożoność obliczeniowa

Przybliżone rozwiązanie – uogólnienie

Dla $a > 0$, oblicz $a^{1/3} = x$, czyli $x^3 = a$



Ciekawostka – użycie **WTF**:

$$2^{1/3} = p/q$$

$$2 = p^3/q^3$$

$$2q^3 = p^3$$

$$q^3 + q^3 = p^3 \text{ ???}$$

$$x' \leftarrow (2x + a/x^2)/3$$

Można **uogólnić** dla: $a^{1/n} = x$, $x^n = a$

Metoda Newtona-Raphsona bez pochodnych

Podstawa programowa matematyki – NWD

Klasy IV – VI. Działania na liczbach naturalnych – jak uczeń znajduje NWD? :

- 3) mnoży i dzieli liczbę naturalną przez liczbę naturalną jednocyfrową, dwucyfrową lub trzycyfrową sposobem pisemnym, w pamięci (w najprostszych przykładach) i za pomocą kalkulatora (w trudniejszych przykładach);
- 4) wykonuje dzielenie z resztą liczb naturalnych;
- 9) rozkłada liczby dwucyfrowe na czynniki pierwsze;
- 13) znajduje największy wspólny dzielnik (NWD) w sytuacjach nie trudniejszych niż typu $NWD(600, 72)$, $NWD(140, 567)$, $NWD(10000, 48)$, $NWD(910, 2016)$ oraz wyznacza najmniejszą wspólną wielokrotność dwóch liczb naturalnych metodą rozkładu na czynniki;
- 16) rozkłada liczby naturalne na czynniki pierwsze, w przypadku gdy co najwyżej jeden z tych czynników jest liczbą większą niż 10;

W podstawie matematyki jest tylko kalkulator, **NIE MA KOMPUTERA.**

Ale to jest problem NP.-trudny

Ale, to nie kształci myślenia abstrakcyjnego

I miód na moje serce – Twierdzenie o ilorazie i reszcie:

- 17) wyznacza wynik dzielenia z resztą liczby a przez liczbę b i zapisuje liczbę a w postaci: $a = b \cdot q + r$.

Pewien brak: $0 \leq r < b$,
ale krótka droga do algorytmu Euklidesa!

Myślenie logarytmiczne

logarytm i algorytm to **anagramy**

$m < n/2$

n : _____

m : _____

$m > n/2$

n : _____

m : _____

- Generowane liczby są połowione co druga
- A zatem, dla 10^{300} , algorytm Euklidesa wykonuje ok. **2000 mnożeń** – to chwila, chwilę czekamy na zaszyfrowany mail.

n	m	r_i
34	21	13
21	13	8
13	8	5
8	5	3
5	3	2
3	2	1
2	1	0

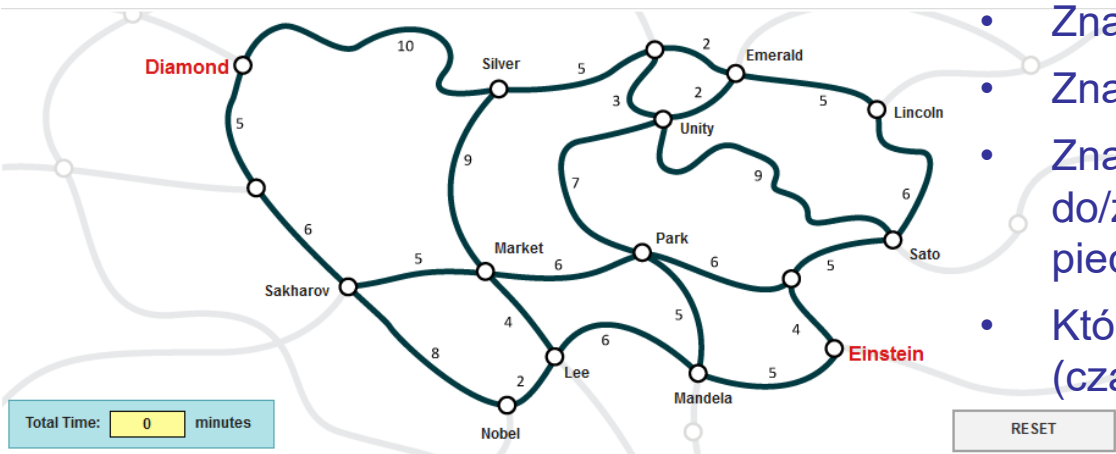
- Logarytm ukryty w algorytmach:
- **!!!!** Algorytmiczna **definicja logarytmu**: ile razy należy podzielić przez dwa liczbę i jej ilorazy, by osiągnąć 1 – można wprowadzić już w gimnazjum!
- Euklides mógł wynaleźć logarytm 300 lat p.n.e., a zrobił to dopiero John Napier 400 lat temu, w 1614 roku.

Geometria – a grafy?

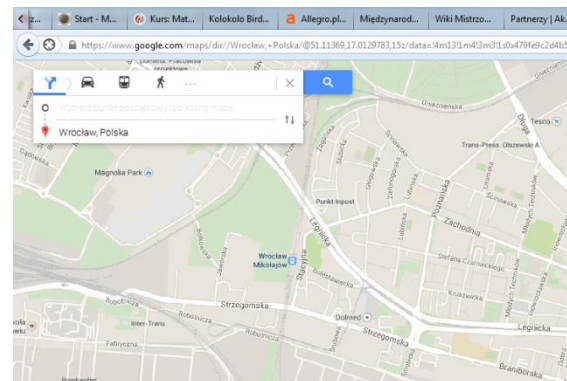
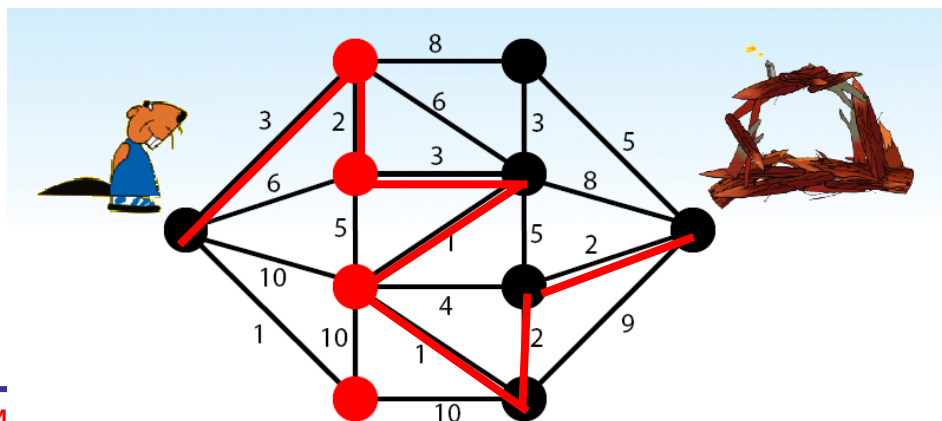
Dlaczego nasi uczniowie wypadli źle (29 miejsce na 32 kraje) w badaniach PISA w zakresie **rozwiązywania problemów**, w 2012-2013?

Testy: najkrótsze drogi

Zajęcia z realnej sytuacji – to uczniów motywuje



- Znajdź swój dom i szkołę na Google map
- Znajdź swoją drogę do/z szkoły
- Znajdź najkrótszą drogę (odległość i czas) do/z szkoły różnymi środkami transportu: na piechotę, na rowerze, samochodem, busem
- Która z tych dróg jest najkrótsza (czas/odległość)?



Myślenie – nie tylko informatyczne

Świetny przykład (A.B. Kwiatkowska): **próbujemy dopasować jedno do drugiego**, porównać zgodność jednego z drugim:

- DNA – na ile jest ono wspólne dla bliźniąt?
- DNA – czy zawiera pewne fragmenty związane z chorobami dziedzicznymi?
- W jakim stopniu pokrywają się prace – plagiat?
- W informatyce – jest to **dopasowanie wzorca**

Podsumowanie – myślenie:

- algorytmiczne (*algorithmic thinking*)
- rekurencyjne (*recursive thinking*)
- redukcyjne (*reductive thinking*)
- logarytmiczne (*logarithmic thinking*)
- projektowe (*design thinking*)
- ...
- **komputacyjne** (*computational thinking*)

Ocenianie

S. Papert: children learn by doing and by thinking what they do

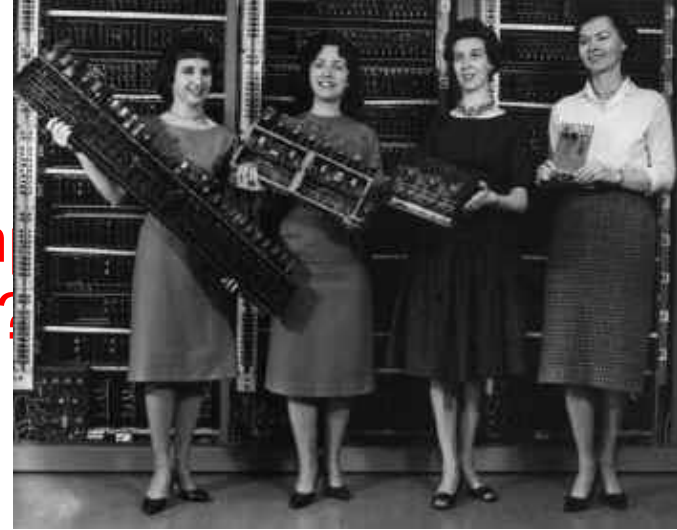
Jak oceniać myślenie (komputacyjne)?

Wątpliwości:

- rezultat – rozwiązanie problemu – nie zawsze świadczy o wykształceniu przez ucznia i zastosowaniu myślenia komputacyjnego
- rozwiązanie może ocenić komputer, ale to nie jest ocena umiejętności ucznia
- to są *mental tools* nie związane ściśle w jednym przedmiotem (na informatyce sugerujemy dobierać przykłady z różnych dziedzin)
- wiele (szkolnych) projektów jest dalekich od posłużenia się tylko komputerem – **myślenie komputacyjne ma wspomóc komputer!**
- badania i statystyki świadczą o ułomności narzędzi badań, ale z drugiej strony pokazują, że naszym uczniom brakuje tego, o czym pisał Papert: *thinking about what they do*

Rozgrzewka

Co ci dwaj Panowie, stojący na tle komputera, trzymają w rękach?



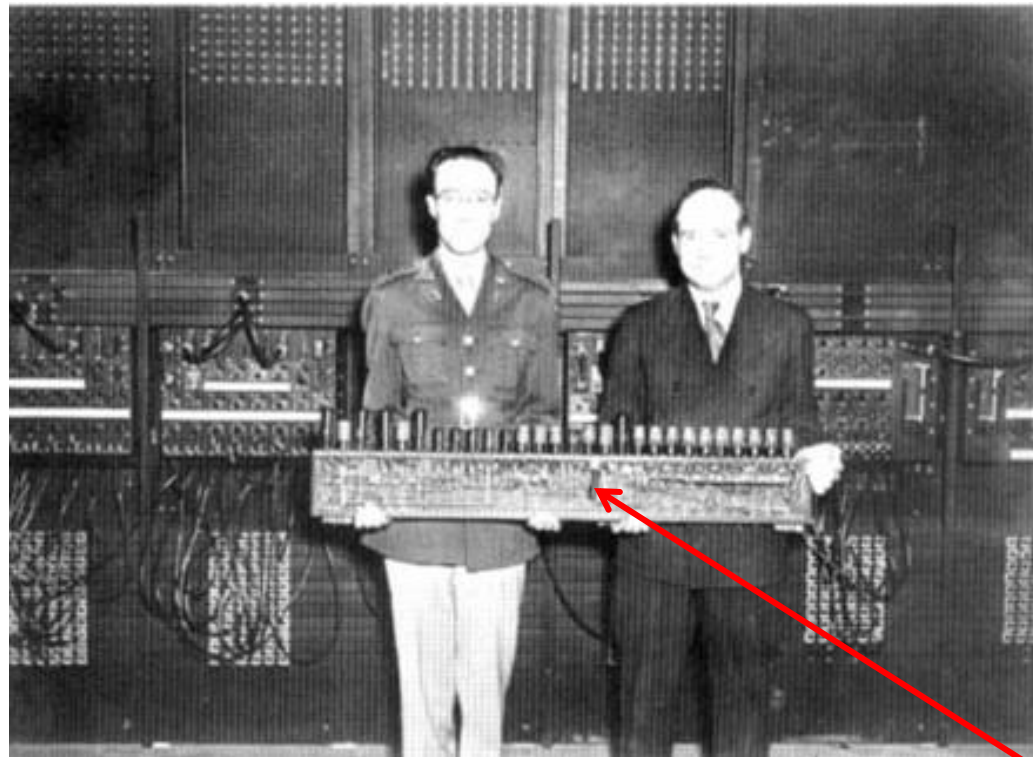
Mysz

Jedną liczbę

Jedną cyfrę

Klawiaturę

Cyfra dziesiętna: 0, 1, ...
zbudowana z 22 lamp!!!



A co po IBM PC/laptopie/tablecie?

Zbuduj własny komputer!!! Na bazie np. brytyjskiego komputera szkolnego **BBC micro:bit** (dostali laureaci Bobra 2018)



Dziękuję Państwu za uwagę
i proszę nie zapomnieć:



<http://mmsyslo.pl>