

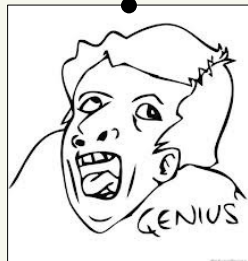
*Czy komputery mogą  
być nieobliczalne ?*

— Czy komputery mogą być nieobliczalne? —

# *Czy ludzie mogą być* **NIEOBLICZALNI?**

**Z PEWNOŚCIĄ TAK!**

▪ **nieobliczalny**  
( w języku potocznym )



nieprzewidywalny

niepoczytalny

szalony

oryginalny

twórczy

# *Jakie obiekty matematyczne nazywa się NIEOBLICZALNYMI?*

✓ LICZBY

*liczby niewymierne, których rozwinięcia dziesiętne nie można obliczyć z dowolną zadaną dokładnością, za pomocą żadnego algorytmu.*

✓ FUNKCJE

*funkcje, których niektóre wartości nie mogą zostać wyznaczone za pomocą żadnego algorytmu.*

✓ PROBLEMY

*problemy, których nie można rozwiązać, we wszystkich przypadkach szczególnych, za pomocą żadnego algorytmu.*

- Problem **nieobliczalny** = problem algorytmicznie nierozwiązywalny.

# *Algorytmy, komputery, OBLICZENIA...*

## ALGORYTM

*opisuje systematycznie, krok po kroku, metodę rozwiązywania pewnego problemu.*

## KOMPUTER

*realizuje metodę algorytmiczną za pomocą programu komputerowego.*

## W KOMPULERZE

*wszelkie obiekty są kodowane liczbowo, a operacje na nich są realizowane jako obliczenia.*

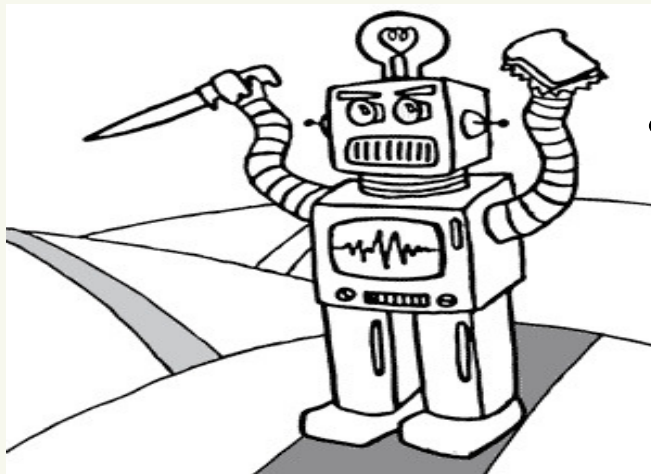
- Realizacja algorytmu → ciąg **obliczeń**.
- Problem rozwiązywalny algorytmicznie → problem **obliczalny**.
- Problem nierozwiązywalny algorytmicznie → problem **nieobliczalny**.

— Czy komputery mogą być nieobliczalne? —

# *G.W. Leibniz i jego* **CALCULEMUS**



**CALCULEMUS** czyli  
**POLICZMY !**



zapiszmy myśli symbolicznie



zakodujemy symbole liczbowo



złęczmy obliczenia maszynie



zamieńmy wyniki na symbole



zinterpretujemy symbole

## LEIBNIZ *o nieobliczalności*



**Wszystko jest  
*obliczalne* !**

- ✓ *W przyszłości, gdy komputery staną się potężniejsze, każdy problem będzie można rozwiązać za pomocą obliczeń.*
- ✓ *Okolo 200 lat później podobne przypuszczenie wyrazi inny słynny matematyk, David Hilbert.*

— Czy komputery mogą być nieobliczalne? —

# *Alan TURING i pierwszy problem nieobliczalny*



- ✓ *W roku 1936 Alan Turing przedstawia matematyczny dowód nieobliczalności tzw. problemu stopu.*

*Czy istnieje taki **uniwersalny** algorytm, który analizując zapis każdego innego algorytmu oraz dowolnych jego danych,*

***rozstrzygnie jednoznacznie***

*czy analizowany algorytm **zakończy** przetwarzanie swoich danych, czy też będzie je przetwarzał w **nieskończoność**?*

**NIE  
ISTNIEJE !**

## *Inne problemy* NIEOBLICZALNE

Czy dane równanie *diofantyczne*, z dowolną liczbą niewiadomych i **całkowitymi** współczynnikami, ma choć jedno **rozwiązanie** w zbiorze liczb **całkowitych** ?

• —  $x^2 + 2y^3 - 4y^2 + z^4 = 0$

Czy dane dwa **języki sztuczne**, z określonymi regułami budowania słów, pozwalają **zbudować**, zgodnie ze swoimi regułami, to samo (dane z góry i dowolne) *słowo* ?

• — `abbbcaabbbababc`



# *Problemy* TRUDNO-OBLICZALNE

Problemy **nieobliczalne** są:

- nierozwiązywalne algorytmicznie
- raczej teoretyczne
- pomijalne w praktyce

Przykład: PROBLEM STOPU

Problemy **trudno-obliczalne** są:

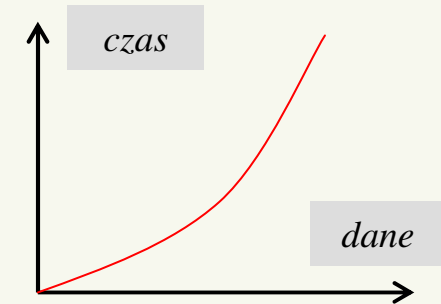
- rozwiązywalne za pomocą nieefektywnych algorytmów
- praktycznie nieobliczalne
- niepomijalne w praktyce

Przykłady: bardzo LICZNE

# Złożoność czasowa algorytmów

Złożoność czasowa jest to:

- własność algorytmów, która informuje o tym, jak rośnie czas wykonywania algorytmu, gdy rośnie rozmiar danych wejściowych.



Miarą złożoności są funkcje matematyczne, na przykład:  $n$ ,  $n^2$ ,  $\log_2 n$ ,  $n \cdot \log_2 n$ ,  $2^n$ .

$O(n)$

*złożoność liniowa*

$O(n^2)$

*złożoność kwadratowa*

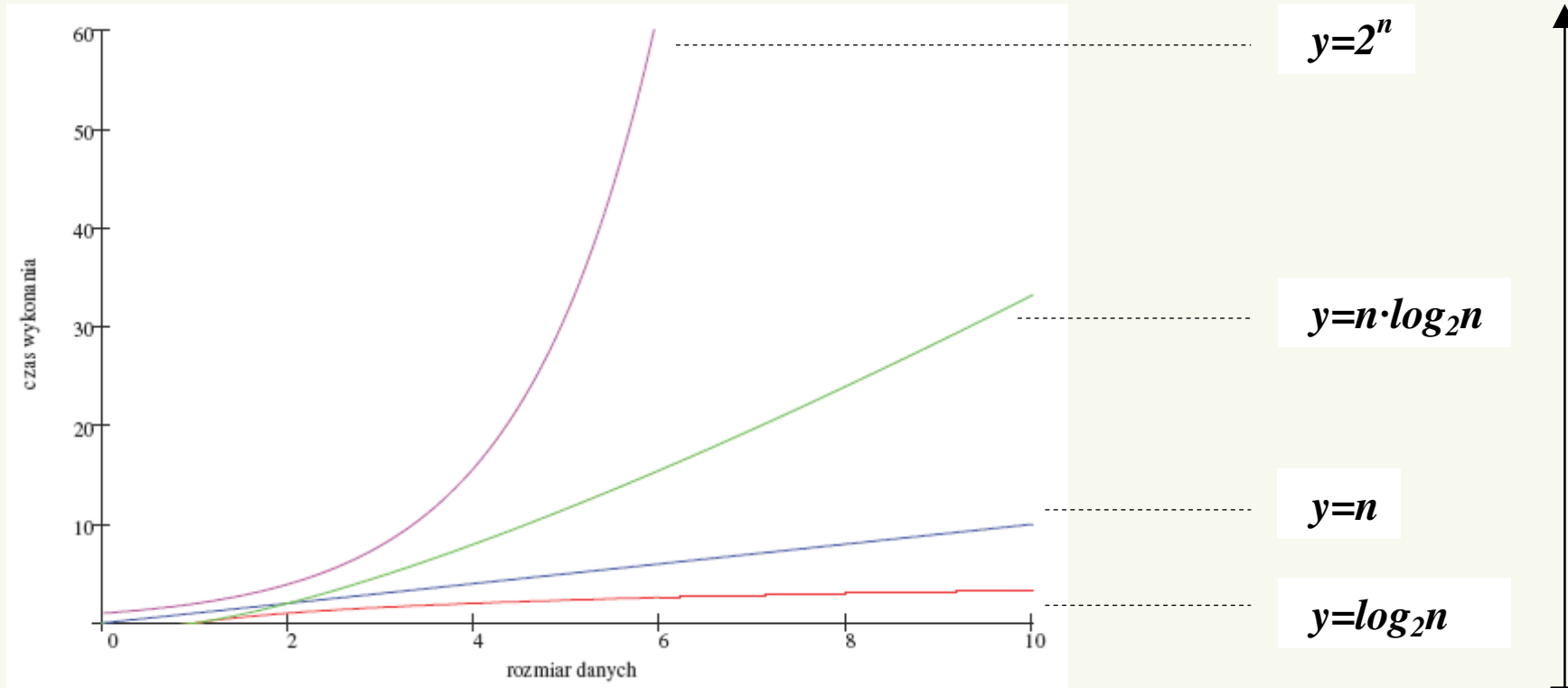
$O(a^n)$

*złożoność wykładnicza*

Złożoność czasowa problemu jest to złożoność czasowa najszybszego algorytmu rozwiązującego ten problem.

— Czy komputery mogą być nieobliczalne? —

# *Jak szybko rosną funkcje złożoności?*



Funkcje logarytmiczne i wielomianowe rosną **wolno**.

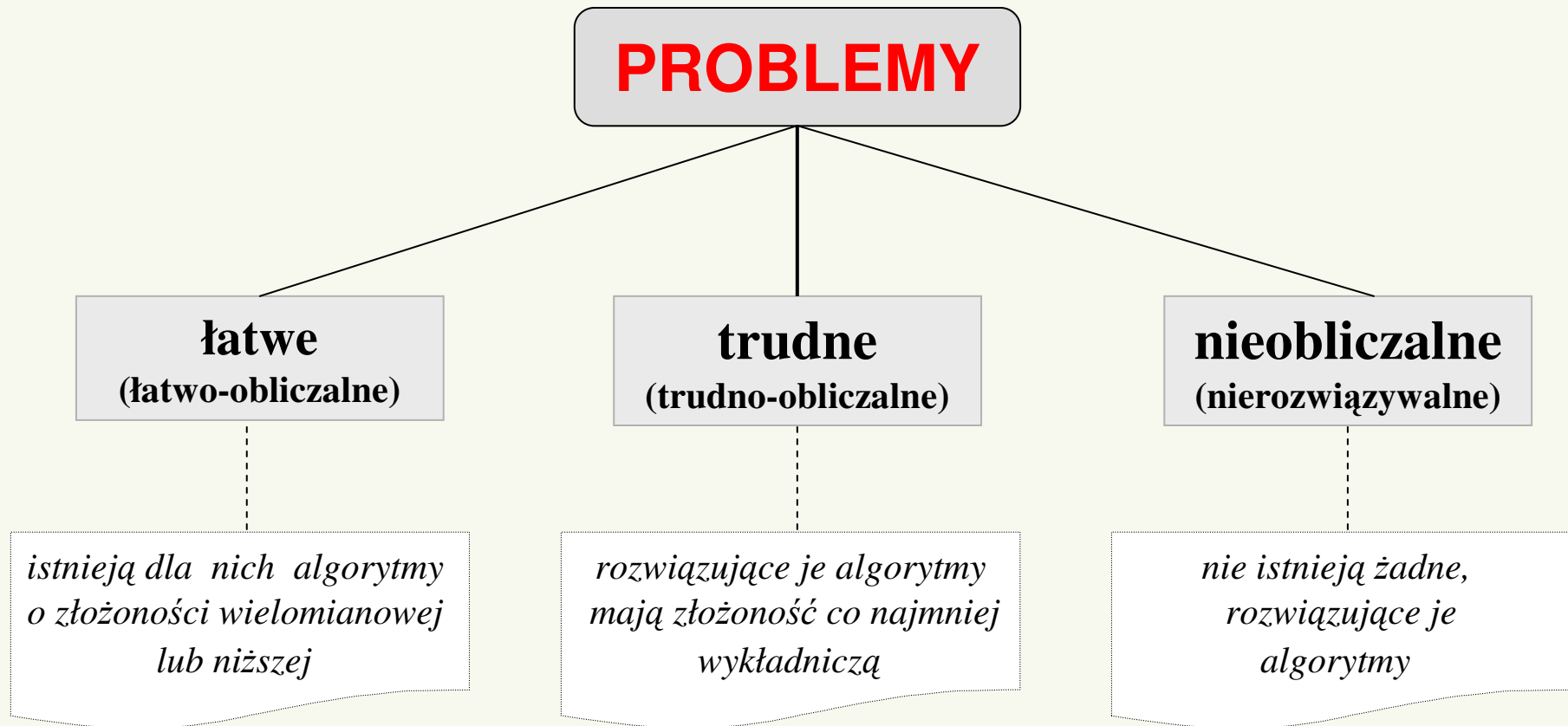
Funkcje wykładnicze i silnia rosną bardzo **szybko** !

## *Funkcje złożoności i ich wartości*

	n=10	n=50	n=100	n=300
$5 \cdot n$	50	250	500	1500
$n \cdot \log_2 n$	33	282	665	2469
$n^2$	100	2500	10000	90000
$2^n$	1024	liczba 16-cyfrowa	liczba 31-cyfrowa	liczba 91-cyfrowa
$n!$	3600000	liczba 65-cyfrowa	liczba 161-cyfrowa	liczba 623-cyfrowa

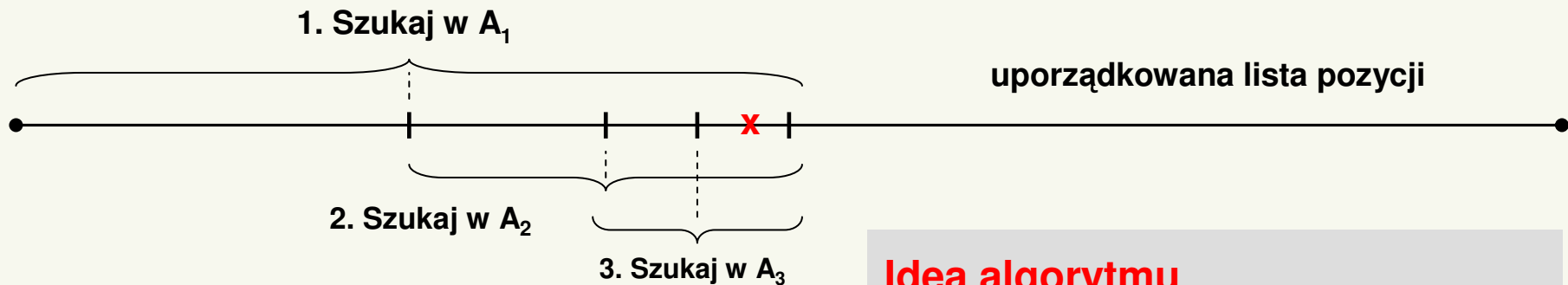
**Uwaga!** Liczba protonów w znanym wszechświecie ma 126 cyfr.

# *Złożoność a klasyfikacja problemów*



# Problemy o złożoności logarytmicznej

## PRZYKŁAD



### Idea algorytmu

( $x$  – szukany element,  
 $A$  – przeszukiwany zbiór )

1. Podziel  $A$  na dwie połowy;  
 $a$  – element środkowy w  $A$ .
2. Jeśli  $x=a$ , zakończ.
3. Jeśli  $x < a$ , to  $A :=$  lewa połowa  $A$ .
4. Jeśli  $x > a$ , to  $A :=$  prawa połowa  $A$ .
5. Przejdź do punktu 1.

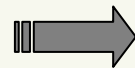
- Przeszukiwanie „nawne” ma złożoność liniową.
- Przeszukiwanie binarne ma złożoność logarytmiczną.

# Problemy o złożoności wykładniczej

## PRZYKŁADY

### Problem spełnialności (logika)

- Czy istnieje takie wartościowanie zmiennych zdaniowych, przy którym formuła zawierająca  $n$  zmiennych jest prawdziwa?

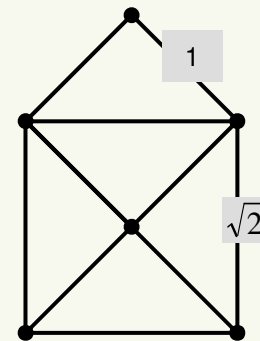


$$(\neg b \wedge (a \rightarrow b)) \rightarrow (\neg a)$$

1                      0                      1                      0

### Problem komiwojażera (grafy)

- W danym grafie z określonymi wagami krawędzi znajdź ścieżkę zawierającą wszystkie wierzchołki, która ma najmniejszą sumaryczną wagę (ścieżkę najkrótszą).



# *Problemy o złożoności silniowej*

## PRZYKŁADY

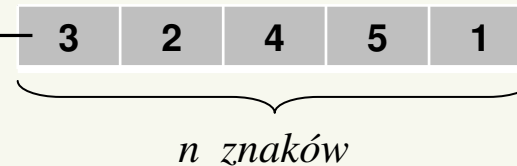
**Złożoność czasową  $n!$  mają:**

- algorytmy sprawdzające wszystkie permutacje danych wejściowych o rozmiarze  $n$ .

*$n!$  to liczba permutacji zbioru  $n$ -elementowego.*

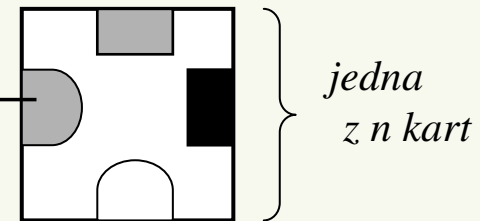
### Przykład 1 (szyfr)

- Zgadnij szyfr  $n$ -znakowy złożony z  $n$  różnych znaków nie powtarzających się.



### Przykład 2 (układanka)

- Czy dla danych  $n$  kart istnieje złożony z nich kwadrat, w którym wszystkie karty stykają się odpowiednimi bokami?

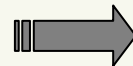




# *Sposób na problemy trudne* **HEURYSTYKI**

**Strategia heurystyczna** jest to:

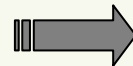
- taka strategia dla problemu P, która wobec trudnej obliczalności problemu P pozwala szybko znaleźć rozwiązania wystarczająco dobre (choć nie idealne).



- **np. w problemie komiwojażera** znajdź ścieżkę nie najkrótszą, lecz wystarczająco krótką.

**Losowa** strategia heurystyczna:

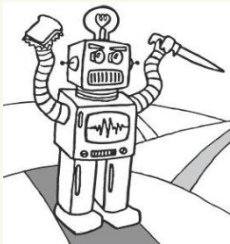
- taka strategia, w której pewne posunięcia zależą od wyborów losowych.



- **np. algorytmy genetyczne** są strategią opartą na heurystyce zaczerpniętej z teorii ewolucji.

# *Czy zatem komputery mogą być* **NIEOBLICZALNE ?**

**TAK**



- bo istnieją ważne problemy **nieobliczalne** (bezwzględnie lub praktycznie), które próbujemy rozwiązywać za pomocą komputerów.
- bo próbując je efektywnie rozwiązać, musimy stosować różne **heurystyki**, zależne często od wyborów losowych.
- bo dopuszczając strategie oparte na wyborach losowych, musimy liczyć się z **nieprzewidywalnością** komputerów.

— Czy komputery mogą być nieobliczalne? —

# Propozycje dla CHĘTNYCH



- *Książka popularno-naukowa, poświęcona m.in. zagadnieniom nieobliczalności i trudnej obliczalności.*



- *Akademicki blog internetowy Witolda Marciszewskiego*  
*Adres: <http://blog.marciszewski.eu>*