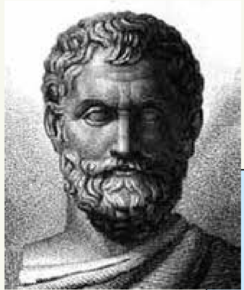
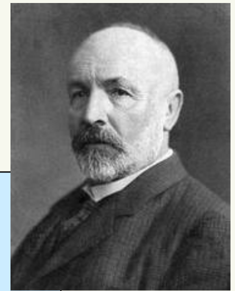
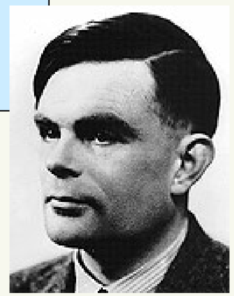




OBLICZALNOŚĆ

I

NIEOBLICZALNOŚĆ



Dwa konteksty obliczalności

OBLICZALNE i NIEOBLICZALNE

- ***problemy*** *(kontekst informatyczny)*
- ***liczby*** *(kontekst matematyczny)*

Problem nieobliczalny

jest to problem nierozwiązywalny algorytmicznie (w ogóle lub tylko praktycznie), za pomocą algorytmów danego typu.

***Przykład**: problem stopu maszyny Turinga.*

Liczba nieobliczalna

jest to taka liczba niewymierna, w przypadku której nie istnieje algorytm obliczania kolejnych cyfr jej rozwinięcia dziesiętnego (od pewnego miejsca).

***Przykład:** liczba Ω G. Chaitina.*

Uwaga *Klasycznie obliczalność definiuje się w odniesieniu do maszyn cyfrowych.*

Od problemów do liczb

Teza

Informatyczny kontekst obliczalności (kwestię algorytmicznej rozwiązywalności problemów) daje się sprowadzić do kontekstu matematycznego (liczbowego).

DLACZEGO ?

- Bo jeśli pewien problem ma algorytmiczne rozwiązanie, to istnieje rozwiązujący go program komputerowy, a ponieważ komputerowa realizacja programu jest ciągiem obliczeń, to sam problem ma pewne rozwiązanie liczbowe (które człowiek tak lub inaczej interpretuje).
  Krótko: rozwiązanie problemu jest liczbą.

- Bo rozwiązujący dany problem program komputerowy, można przedstawić jako pewną liczbę (przyjmując określoną metodę kodowania).
  Krótko: metoda rozwiązania (program) jest liczbą.

Informatyczny kontekst obliczalności

Jeśli jest dany problem P, to:

- albo istnieje efektywny program rozwiązujący problem P (o małej złożoności czasowej/pamięciowej).

- albo istnieją tylko nieefektywne programy rozwiązujące P (o dużej złożoności czasowej/pamięciowej).

- albo nie istnieje żaden program rozwiązujący P (we wszystkich przypadkach szczególnych).

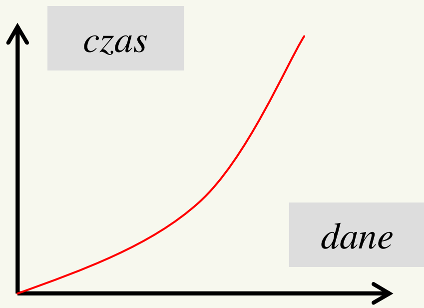
Obliczalność problemu P zależy od tego, która z powyższych alternatyw jest dla problemu P prawdziwa.

Kluczowe pojęcia: **efektywność**, **złożoność**, **rozwiązywalność**.

ZŁOŻONOŚĆ *czasowa algorytmów*

Złożoność czasowa jest to:

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

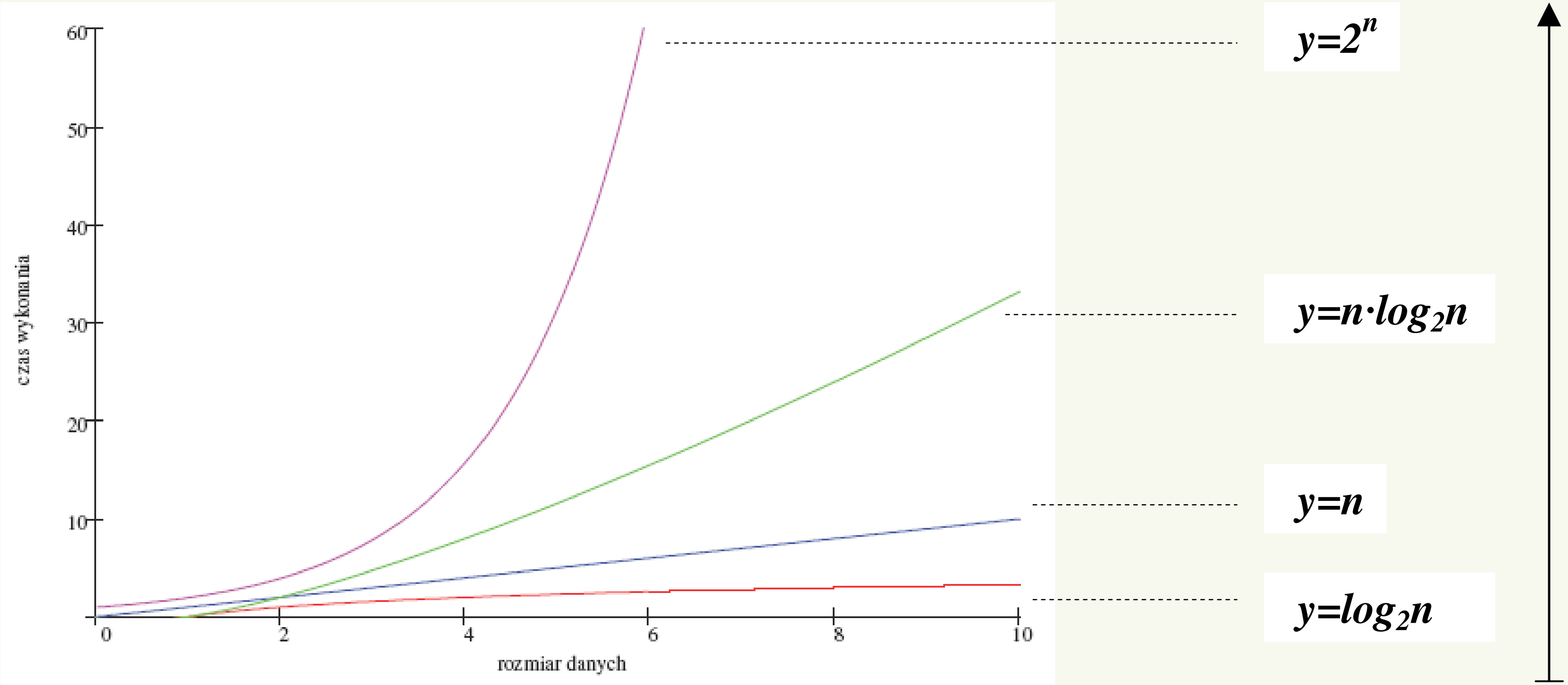


Miarą złożoności są funkcje matematyczne, na przykład: n, n^2, $\log_2 n$, $n \cdot \log_2 n$, 2^n

$O(n)$	*złożoność liniowa*
$O(n^2)$	*złożoność kwadratowa*
$O(a^n)$	*złożoność wykładnicza*

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

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

łatwe
(łatwo-obliczalne)

istnieją dla nich algorytmy o złożoności wielomianowej lub niższej

trudne
(trudno-obliczalne)

rozwiązujące je algorytmy mają złożoność co najmniej wykładniczą

nierozwiązywalne
(nieobliczalne)

nie istnieją żadne, rozwiązujące je algorytmy

Problemy o złożoności wykładniczej
PRZYKŁADY

Problem spełnialności (logika)

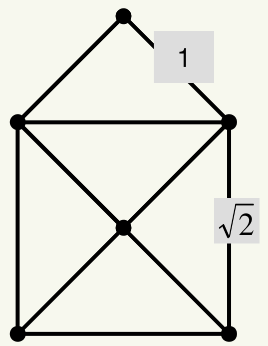
- *Czy istnieje takie wartościowanie zmiennych zdaniowych, przy którym formuła zawierająca n zmiennych jest prawdziwa* ?

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

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)

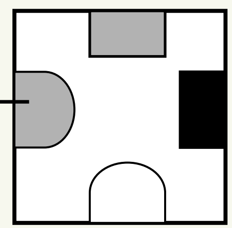
- *Zgadnij szyfr n-znakowy złożony z n różnych znaków nie powtarzających się.*

3	2	4	5	1

n znaków

Przykład 2 (układanka)

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



jedna z n kart

Problemy nieobliczalne

Problem stopu (maszyny Turinga)

✓ *Dla dowolnej maszyny MT_i i jej dowolnych danych wejściowych D_j*

odpowiedz jednoznacznie,

czy MT_i zatrzyma się dla danych D_j tj. zakończy przetwarzanie danych D_j?

Mniej technicznie:

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?

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

Teorio-liczbowy kontekst obliczalności

Liczby obliczalne…
(wg. Turinga)

- Istnieje dla nich algorytm obliczania.
- Istnieje maszyna Turinga obliczająca je z dowolną dokładnością.
- Istnieje algorytm obliczania ich kolejnych cyfr.

Liczby nieobliczalne…
(wg. Turinga)

- *Nie istnieje dla nich algorytm obliczania.*
- *Żadna maszyna Turinga nie potrafi obliczyć ich z dowolną zadaną dokładnością.*
- *Nie istnieje algorytm obliczania ich kolejnych cyfr.*

Od niewymierności do nieobliczalności

1. Istnieją liczby niewymierne.
(Pitagorejczycy, VI w p.n.e)

2. Liczby naturalne i wymierne są równoliczne.
(G. Cantor, koniec wieku XIX)

3. Liczby naturalne i rzeczywiste nie są równoliczne.
(G. Cantor, koniec wieku XIX)

4. Zbiór liczb niewymiernych ma moc continuum.
(G. Cantor, koniec wieku XIX)

5. Istnieją różne podklasy liczb niewymiernych.
(XIX i XX wiek)

6. Istnieją niewymierne liczby nieobliczalne.
(Turing 1936)

7. Zbiór liczb nieobliczalnych ma moc continuum.

8. Liczby nieobliczalne są czysto losowe.
(G. Chaitin 1987)

Niewymierne liczby obliczalne

Niektóre liczby niewymierne – jest ich tylko przeliczalnie wiele – są algorytmicznie wyznaczalne, czyli obliczalne (w sensie turingowskim).

$$\sqrt{2} = 1{,}414213 \ \ldots$$

liczba algebraiczna, rozw. równania x^2-2=0

$$l = \frac{c_1}{10^{1!}} + \frac{c_2}{10^{2!}} + \frac{c_3}{10^{3!}} + \ldots$$

liczby Louiville'a (przestępne)

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \ldots = 4 \cdot \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1}$$

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \ldots = \sum_{n=0}^{\infty} \frac{1}{n!}$$

najszerzej znane obliczalne liczby przestępne

Odkrycie Chaitina

Używając jako narzędzia pracy **oprogramowania** komputerowego (…) G. Chaitin skonstruował — jak pisze — „*przewrotne (niezwykle skomplikowane) 200-stronicowe **równanie algebraiczne** z parametrem N i 17 tysiącami zmiennych*”.

Następnie postawił pytanie: „*Czy dla każdej całkowitej wartości liczbowej parametru N istnieje **skończona** czy też **nieskończona** ilość całkowitych liczbowych rozwiązań?*”

Odpowiedź wypadła zdumiewająco. Jeśli bowiem do równania podstawiać kolejne wartości liczbowe parametru N oraz w przypadku skończonej liczby rozwiązań przyjmować **0**, zaś w przypadku nieskończonej **1**, to jego rozwiązaniem będzie ciąg zer i jedynek, którego w żaden sposób nie można odróżnić od ciągu zestawiającego wyniki nieskończenie wielu **rzutów monetą**.

Nieobliczalną liczbę rzeczywistą z przedziału między 0 i 1, odpowiadającą ciągowi otrzymanych zer i jedynek, G. J. Chaitin nazwał następnie Ω.

- **Ω = 0010111011001001100001...**

Interpretacje odkrycia Chaitina

J. Dębowski o odkryciu Chaitina:

Jak się okazuje, kolejne cyfry liczby Ω odpowiadają nieskończonej liczbie zupełnie **przypadkowych** faktów arytmetycznych. Wiemy wprawdzie, że każdy bit Omegi musi być albo zerem albo jedynką, ale nie wiemy i nigdy wiedzieć nie będziemy (!), kiedy i dlaczego wystąpi w niej zero, a kiedy i dlaczego — jedynka. Sytuacja jest więc w maksymalnym stopniu matematycznie **nieprzewidywalna**.
Innymi słowy, Omega (Ω), ponieważ stanowi skrajnie **nieuporządkowaną** sekwencję zer i jedynek, jest nieredukowalna do żadnego algorytmu — jest, jak się powiada, **algorytmicznie nieupraszczalna** (niekompresowalna).

Chaitin o losowości i niekompresowalności:

Można więc zdefiniować losowość jako coś, czego w ogóle nie można skompresować. Jedyny sposób, by opisać zupełnie losowy obiekt lub liczbę komuś, to pokazać go i powiedzieć: „**To jest to**”.
Ponieważ nie ma żadnej struktury lub **prawidłowości**, nie ma żadnego krótszego opisu. Drugą skrajnością jest obiekt lub liczba, który ma bardzo regularną strukturę. Być może można by go opisać, mówiąc, że jest to milion powtórzeń 01, na przykład. Jest to bardzo duży obiekt o bardzo krótkim opisie.