

# Teoretyczne podstawy informatyki

<http://hibiscus.if.uj.edu.pl/~erichter/Dydaktyka2008/TPI-2008>

# Zakres tematyczny

---

1. Co to jest informacja?
2. Algorytmy i struktury danych, poprawność algorytmu
3. Złożoność obliczeniowa cz. I
4. Rekursja, indukcja, iteracja, teoria prawdopodobieństwa
5. Modele danych: drzewa, listy, zbiory, relacje, grafy
6. Złożoność obliczeniowa, cz. II
7. Wzorce, automaty, wyrażenia regularne i gramatyki
8. Języki formalne, problemy NP-zupełne

## Literatura

1. H. Abelson, et al., *Struktura i interpretacja programów komputerowych*
2. [A. V. Aho, J. D. Ullman, \*Wykłady z informatyki z przykładami w języku C\*](#)
3. T. H. Cormen, Ch. F. Leiserson, R. L. Rivest, *Wprowadzenie do algorytmów*
4. [A. Drozdek, D. L. Simon, \*Struktury danych w języku C\*](#)
5. D. Harel, *Rzecz o istocie informatyki*
6. [J.E. Hopcroft, J. Ullman, \*Wprowadzenie do teorii automatów, języków i obliczeń\*](#)
7. S. Kowalski, A. W. Mostowski, *Teoria automatów i lingwistyka matematyczna*
8. Ch. H. Papadimitriou, *Złożoność obliczeniowa*
9. W. Sikorski, *Wykłady z podstaw informatyki*
10. W. M. Turski, *Propedeutyka Informatyki*
11. N. Wirth, *Algorytmy i struktury danych = programy*

## Zaliczenie przedmiotu

- zaliczenie ćwiczeń z zadań rachunkowych
- egzamin pisemny: podany obowiązujący zestaw zagadnień z wykładu

# Informatyka

---

Mimo że informatyka jest stosunkowo nową dziedziną nauki, już wpływa na niemal wszystkie aspekty działalności człowieka.

- Jej wpływ na funkcjonowanie społeczeństw jest widoczny w rozpowszechnianiu się komputerów, systemów informatycznych, edytorów tekstu, arkuszy kalkulacyjnych itd...
- Ważną cechą informatyki jest ułatwianie samego programowania i czynienie programowania bardziej niezawodnym

# Informatyka: mechanizacja abstrakcji

---

Zasadniczo jednak **informatyka jest**

- *nauką o abstrakcji* , czyli nauką o tworzeniu właściwego modelu reprezentującego problem i wynajdowaniu odpowiedniej techniki mechanicznego jego rozwiązywania
- Informatycy tworzą abstrakcje rzeczywistych problemów w formach które mogą być rozumiane i przetwarzane w pamięci komputera

## *Abstrakcja*

oznaczać będzie pewne uproszczenie, zastąpienie skomplikowanych i szczegółowych okoliczności występujących w świecie rzeczywistym zrozumiałym modelem umożliwiającym rozwiązanie naszego problemu. Oznacza to że *abstrahujemy od szczegółów* które nie mają wpływu lub mają minimalny wpływ na rozwiązanie problemu. Opracowanie odpowiedniego modelu ułatwia zajęcie się istotą problemu.

# Informatyka: mechanizacja abstrakcji

---

## W ramach tego wykładu omówimy

- **modele danych:** abstrakcje wykorzystywane do opisywania problemów
- **struktury danych:** konstrukcje języka programowania wykorzystywane do reprezentowania modeli danych. Przykładowo język C udostępnia wbudowane abstrakcje takie jak struktury czy wskaźniki, które umożliwiają reprezentowanie skomplikowanych abstrakcji takich jak grafy
- **algorytmy:** techniki wykorzystywane do otrzymywania rozwiązań na podstawie operacji wykonywanych na danych reprezentowanych przez abstrakcje modelu danych, struktury danych lub na inne sposoby

## Trochę historii .....

---

Gdzieś między 400 a 300 rokiem p.n.e wielki grecki matematyk **Euklides** **wynalazł algorytm znajdowania największego wspólnego dzielnika** (nwd) dwóch dodatnich liczb całkowitych. Szczegóły algorytmu są nieistotne... algorytm Euklidesa uważa się za pierwszy kiedykolwiek wymyślony niebanalny algorytm.

**Słowo algorytm** wywodzi się od nazwiska perskiego matematyka Muhammeda Alchwarizmi (łac.. Algorismus), który żył w IX wieku p.n.e i któremu przypisuje się podanie reguł dodawania, odejmowania, mnożenia i dzielenia zwykłych liczb dziesiętnych.

Jedną z najwcześniejszych maszyn wykonujących proces sterowany czymś co można nazwać algorytmem jest **krosno tkackie** wynalezione w 1801 roku przez Josepha Jacquarda. Tkany wzór określały karty z otworami wydziurkowanymi w różnych miejscach. Te otwory, które wyczuwał specjalny mechanizm, sterowały wyborem nitek i innymi czynnościami maszyny.

# Maszyna Babbage

---

Jedną z najważniejszych i najbardziej barwnych postaci w historii informatyki był Charles Babbage. Ten angielski matematyk, częściowo zbudowawszy w roku 1833 urządzenie zwane maszyną różnicową, służące do obliczania pewnych wzorów matematycznych, obmyślił i zrobił plany godnej uwagi maszyny zwanej **maszyną analityczną**

- **maszyna różnicowa**  $\Rightarrow$  realizowała konkretne zadanie
- **maszyna analityczna**  $\Rightarrow$  realizowała konkretny algorytm czyli program zakodowany w postaci otworów wydziurkowanych na kartach

Maszyny Babbage były w swej naturze mechaniczne, oparte raczej na dźwigniach, trybach i przekładniach, a nie na elektronice i krzemie

- **Koncepcje zawarte w projekcie maszyny analitycznej Babbage'a tworzą podstawę wewnętrznej struktury i zasad działania dzisiejszych komputerów**



# Algorytmika i komputery

---

- **Połowa lat trzydziestych** to niektóre z najbardziej fundamentalnych prac nad teorią algorytmów, uzmysławiających możliwości i ograniczenia algorytmów wykonywanych przez maszyny
  - **Kluczowe postacie to:** Alan Turing (Anglik), Kurt Goedel (Niemiec), Andriej A. Markow (Rosjanin), Alonzo Church, Emil Post i Stephen Kleene (Amerykanie)
  - **Lata pięćdziesiąte i sześćdziesiąte** to szybkie postępy w budowie komputerów: era badań jądrowych i kosmicznych, postępy w dziedzinie łączności wspieranej przez komputery (filtrowanie i analiza); gospodarka, bankowość, itd.
- **Uznanie informatyki za niezależną dyscyplinę akademicką** nastąpiło w połowie lat sześćdziesiątych

# Komputery i bity

---

- Każdy komputer może wykonywać bezpośrednio jedynie niewielką ilość skrajnie prostych operacji, takich jak przerzucenie, wyzerowanie lub sprawdzenie bitu (wartość bitu określa zazwyczaj pewna cecha elektroniczna, taka jak to czy w pewnym punkcie znajduje się ładunek dodatni czy ujemny).
  - Różne komputery różnią się między sobą wielkością (tj. liczbą dostępnych bitów), rodzajami operacji elementarnych które mogą wykonać, szybkością z którą te operacje są wykonywane, fizycznym nośnikiem realizującym bity oraz wewnętrzną organizacją i, bardzo znacznie środowiskiem zewnętrznym.
- **W jaki sposób proste operacje na bitach realizują tak niezwykle funkcje które może spełniać komputer:**
- proces
  - algorytm, który tym procesem kieruje

# Informacja i zasady jej zapisu

---

1. Czym jest informacja?
2. Systemy zapisu liczb
  - System dwójkowy
  - System szesnastkowy
  - Bajty
3. Znaki i teksty
4. Wielkości liczbowe
5. Obrazy i dźwięki
6. Kompresja i szyfrowanie

# Czym jest informacja?

---

## Istnieje kilka różnych definicji pojęcia informacja

(encyklopedia PWN)

- konstatacja stanu rzeczy, wiadomość
- obiekt abstrakcyjny który w sposób zakodowany może być przesyłany, przetwarzany i używany do sterowania
- powiadamianie społeczeństwa lub określonych zbiorowości w sposób zobiektywizowany, systematyczny i konkretny za pomocą środków masowego przekazu

## Nas interesuje ta druga definicja

- Informacją zajmuje się nauka zwana Teorią Informacji  
Dotyczy ona przekazywania wiadomości ze źródła wiadomości do ich przeznaczenia - odbiorcy
- Informację możemy mierzyć ilościowo i jakościowo

# Co przekazuje informację?

---

## Informację przekazuje możliwość porównania dwóch stanów

- Przykład  
Dzwonek dzwonka informuje nas że ktoś nacisnął przycisk, kiedy przycisk się zatnie i dzwonek dalej dzwoni już nie informuje nas o niczym, gdy przestanie dzwonić a my porównamy dwie sytuacje uzyskamy informację że usterka została usunięta
- Dygresja  
**Brak zmian to brak informacji:** niezmienny sygnał nosi nazwę szumu  
Nie można go jednak ignorować, gdyż często zakłóca przekaz właściwej informacji

# Jednostka informacji: bit

---

## Podstawową jednostką informacji jest bit

- Bit jest to **podstawowa elementarna jednostka informacji** wystarczająca do zakomunikowania jednego z co najwyżej dwóch jednakowo prawdopodobnych zdarzeń.
- Bit stanowi podstawę zapisu informacji w różnych typach pamięci komputera.  
Wszystkie inne jednostki stanowią różną wielokrotność bitów.
- **Symbolicznie wartość bitu oznacza się jako „0” lub „1”**  
Jest to oznaczenie stosowane w matematyce oraz przy opisie informacji przechowywanej w pamięci komputera i opisie sposobów kodowania informacji.

# Systemy zapisu liczb

Na ogół operujemy systemami pozycyjnymi, np. rzymski, dziesiętny.

- **System pozycyjny** tzn. że wartość zapisywanego znaku zależy od jego miejsca, położenia
  - ★ „rzymski” = system pozycyjny sekwencyjny
  - ★ „dziesiętny” = system pozycyjny wagowy

$$L = \sum_{i=m}^n a_i N^i$$

gdzie:  $m, n \in \mathbb{C}$

$$m \geq 0, n \geq 0, m \leq n, N \leq 2, a \in \{0, \dots, N-1\}$$

$N$  nazywamy podstawą systemu, zaś  $a$  jest elementem zbioru cyfr dostępnych w danym systemie

- **W systemie dziesiętnym:**  $N = 10, a \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

# System dwójkowy

---

## System dwójkowy jest naturalnym systemem informatyki

Jak zapisujemy informację?

- Za pomocą zjawisk elektrycznych, magnetycznych, świetlnych  
Zamiast skomplikowanych pomiarów które by pozwoliły zapisać 10 cyfr mamy proste i jednoznaczne kodowanie.
  - Materiał półprzewodnikowy: gdy przyłożymy napięcie w jednym kierunku przewodzi prąd (prawie idealnie), a w kierunku przeciwnym nie przewodzi prądu. Mamy więc dwa stany
  - Podobnie jest w magnetyzmie: substancje magnetyczne można namagnesować w dwóch kierunkach
- Wadę systemu dwójkowego stanowi długość liczby, np.  
 $(0010001110100101)_2 = 2^{13} + 2^9 + 2^8 + 2^7 + 2^5 + 2^2 + 2^0 = (9125)_{10}$



# Bajty

- Przyjęło się stosowanie **jednostki liczącej 8 bitów**, nazwanej bajtem
- Bajt to **podstawowa komputerowa jednostka informacji**
- Dzisiaj mamy kilobajty, megabajty, gigabajty, terabajty,...

Nazwa	Liczba bajtów	Potoczne rozumienie
Kilobajt	$2^{10} = 1\ 024$	$10^3$ (tysiąc)
Megabajt	$2^{20} = 1\ 048\ 576$	$10^6$ (milion)
Gigabajt	$2^{30} = 1\ 073\ 741\ 824$	$10^9$ (miliard)
Terabajt	$2^{40} = 11\ 099\ 511\ 627\ 776$	$10^{12}$ (bilion)

Jeden bajt może reprezentować 256 różnych wartości, które mogą oznaczać zapisywane informacje

# Kodowanie informacji

---

Jak to się dzieje że w pamięci komputera można przechowywać teksty, obrazy, dźwięki i liczby znacznie różniące się od zestawu 0 – 256?

➤ Dzięki kodowaniu informacji

Bez kodowania nie ma zapisu różnorodnych informacji w pamięci komputera.

Kodowanie występuje w każdym programie i na każdym poziomie.

# Znaki i teksty

- Teksty składają się ze znaków
- Podstawą zapisu jest jeden bajt
- 1 bajt przyjmuje 256 różnych wartości
- Ważną cechą kodowania jest **jednoznaczność**:  
przyjęcie pewnego sposobu kodowania powinno być powszechne:  
ASCII: 0 – 127 standardowe, 128 – 256 zależne od kraju

Znaki specjalne	0-31
Spacja	32
Cyfry	48 – 57
Wielkie litery	65 – 90
Małe litery	97 - 122
Pozostałe kody: Kropka, przecinek, itd...	33-47, 58-64, 91-96, 123-127

Np.  
**Litera W: 01010111**  
kod binarny 87

# Znaki i teksty

---

- W rozszerzonym kodzie ASCII znajdują się niektóre znaki matematyczne oraz znaki symulujące **elementy grafiki** na komputerach
- Przetwarzanie informacji nie oznacza samego zapisywania tekstów. Dodatkowe informacje (wyłuszczenie, różne czcionki, akapity...) też trzeba zakodować

Przykład: W kodzie ASCII znaki 0-31 nie są wykorzystane.

Jeżeli umówimy się że po jednym z tych znaków następny zmienia znaczenie, to mamy 255 dodatkowych kodów.

Kod 65 występujący po tym wybranym znaku nie oznacza litery A tylko jedną z funkcji sterujących pracą edytora.

- Dodatkowe kody pozwalają zapisać znacznie więcej informacji ale wymagają dekodowania wg. tych samych reguł z jakimi były kodowane

# Wielkości liczbowe

- Liczby naturalne

Jeden bajt  $\Rightarrow$  0, 1, ..., 255

Dwa bajty  $\Rightarrow$  ...,  $2^{16} - 1$  (czyli około  $6 \cdot 10^4$ )

Cztery bajty  $\Rightarrow$  ...,  $2^{32} - 1$  (czyli około  $4 \cdot 10^9$ )

- Liczby ujemne: kodowanie w **systemie znak-moduł**

Umawiamy się że jeden bit z liczby oznacza jej znak (np. ósmy bit)

Dla jednego bajta otrzymamy liczby (-127, 127)

Ten zakres można rozszerzyć używając dwa bajty, cztery bajty, itd...

- Problem: **niejednoznaczność definicji zera**

+ 0 = 00000000

- 0 = 10000000

# Liczby ujemne

## ➤ kodowanie w systemie *znak-moduł*

Bit	7	6	5	4	3	2	1	0
Znaczenie	znak	64	32	16	8	4	2	1

## ➤ kodowanie w systemie *uzupełnieniowym*

Bit	7	6	5	4	3	2	1	0
Znaczenie	128	- 64	- 32	- 16	- 8	- 4	- 2	- 1

Jeżeli kolejnym bitom przypiszemy wartości jak w tabeli to otrzymamy liczby z zakresu  $(-127, 128)$ .

Nie ma podwójnej reprezentacji zera ale przedział jest niesymetryczny.

Ta asymetria jest wpisana w metodę ponieważ w bajcie możemy zakodować 256 wartości, odliczając ciąg znaków oznaczających zero zostaje nam różnych 255 wartości.

# Wielkości liczbowe

---

## Liczby całkowite: kodowanie w systemie *uzupełnieniowym*

- Wszystkie otrzymane wartości są dokładne
- Istnieje górne i dolne ograniczenie zakresu wartości liczb
- Ograniczenia te zależą od tego ile bajtów przeznaczymy na liczbę oraz od systemu kodowania znaku
- Przy takim zapisie umawiamy się, że przecinek leży za prawym skrajnym znakiem

Ten system kodowania nazywamy też

➤ *systemem stałoprzecinkowym*

Otrzymujemy dla niego **zawsze dokładne wartości**

# Wielkości liczbowe

---

## Liczby rzeczywiste:

- Liczby rzeczywiste mają część całkowitą i ułamkową
- Nie można już przyjąć że przecinek leży po prawej stronie (bo wtedy byśmy mieli tylko liczby całkowite) ani że leży po lewej stronie (bo wtedy byśmy mieli tylko liczby ułamkowe)
- Niezbyt „ekonomiczne” było by używanie kodowania w systemie stałoprzecinkowym (np. przecinek rozdziela dwa bajty)

Co chcemy uzyskać?

- System kodowania dla którego błąd względny będzie tego samego rzędu dla wszystkich wartości biorących udział w obliczeniach



# Wielkości liczbowe

---

Liczby rzeczywiste:


➤ kodowanie w systemie  
*zmiennoprzecinkowym zwanym też  
cecha-mantysa*

- umożliwia zapis liczb rzeczywistych z ustalonym błędem względnym
- system oparty na podziale liczby na część ułamkową zwaną mantysą oraz na wykładnik potęgi podstawy systemu zwany cechą
- opracowany na podstawie zapisu liczby w systemie pozycyjnym wagowym

# Zapis zmiennoprzecinkowy

---

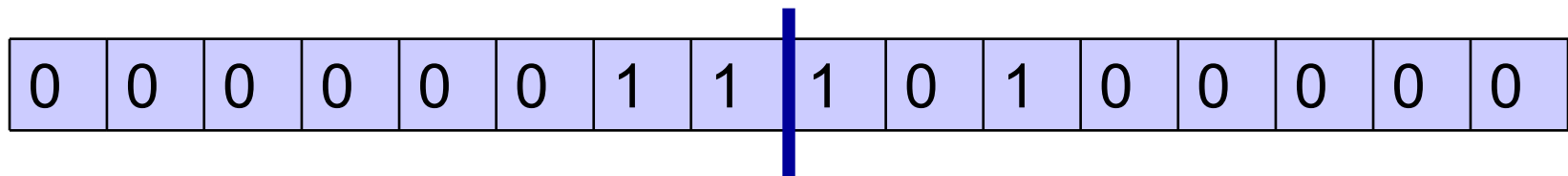
$$L = M \times N^E$$

- M - **mantysa** - liczba mniejsza od jedności  
mantysa znormalizowana należy do przedziału  $< 0.1; 1)$ ,  
co oznacza że pierwszy znak po przecinku musi być różny od zera
- N - **podstawa systemu** zgodnie z zapisem pozycyjnym wagowym
- E - **cecha** - wykładnik potęgi, dzięki któremu przecinek w liczbie zostaje przesunięty tak, aby utworzyć mantysę w zgodzie z powyższą definicją
- (binarnie) 

# Zapis zmiennoprzecinkowy

---

Liczba binarna zapisana w postaci cecha-mantysa na dwóch bajtach



cecha

mantysa

W praktyce zwykle na cechę przeznaczamy jeden bajt, na mantysę minimum trzy bajty.

- Ilość bajtów przeznaczonych na **cechę** decyduje o **zakresie**
- Ilość bajtów przeznaczonych na **mantysę** decyduje o **błędzie**

Liczby ujemne w mantysie są kodowane w systemie znak-moduł, zaś dla cechy w systemie uzupełnieniowym

# Standard zapisu liczb zmiennie-przecinkowych

---

Standard IEEE dla liczby rzeczywistej: (4 bajty)

kolejne bity (od lewej)	znaczenie
1 (jeden)	znak mantysy
2-9 (osiem)	cecha
10-33 (dwadzieścia trzy)	mantysa

Standard IEEE dla liczby podwójnej precyzji: (8 bajtów)

kolejne bity (od lewej)	znaczenie
1 (jeden)	znak mantysy
2-12 (jedenaście)	cecha
13-64 (pięćdziesiąt dwa)	mantysa

# Obrazy, dźwięki, ...

---

Ciągi bajtów muszą przechować teksty, liczby, muzykę, animacje: wszystkie informacje zapisywane w wyniku wykonywanych działań.

Potrzebne jest zakodowanie informacji, inne niż w przypadku liczb czy też tekstów.

## Kodowanie koloru

- bit = 0      biały
- bit = 1      czarny
- kolor budowany jest z kilku bitów

## Kodowanie rysunku

- obraz → mapa bitowa. (plik .bmp)
- dokładne (formaty .tif, .gif)
- uproszczone (format .jpg)

## Kodowanie muzyki

- mp3
- mp4 – DVD

➤ **Im większa precyzja tym większy rozmiar pliku**

# Kompresja i szyfrowanie

---

## Kompresja

Działanie mające na celu zmniejszanie objętości pliku. Przy kompresji wykorzystuje się podobieństwa i regularności występujące w plikach. Program przeprowadza analizę i wybiera fragmenty, które można zapisać w sposób zajmujący mniejszą liczbę bajtów.

- **Kompresja bezstratna:** odtworzona informacja jest identyczna z oryginałem; dekompresja jest w pełni odwracalna.
- **Kompresja stratna:** polega ona na eliminowaniu pewnych elementów oryginału, w celu lepszej efektywności kompresji. Możemy powiązać jakość ze stopniem kompresji.

## Szyfrowanie

Najczęściej stosowane algorytmy oparte na wykorzystaniu liczb pierwszych. Informacja jest szyfrowana przed wysłaniem, odczytanie wymaga posiadania dekodera.