

Wykład 1

- Informacje o przedmiocie
- Zakres przedmiotu
- Literatura
- Pojęcia podstawowe

1

Wstęp do informatyki

Liczba godzin: Semestr 1, wyk. 30 godz., ćw. 30 godz.

Wykład: Marek Gajęcki

Ćwiczenia: Marek Gajęcki
Rafał Grzeszczuk
Jarosław Baniewicz
Paweł Jemioło

Cel wykładu: Celem przedmiotu jest zapoznanie studentów z podstawowymi pojęciami informatyki, programowaniem w języku proceduralnym oraz wprowadzenie do podstawowych algorytmów i struktur danych.

Slajdy i inne materiały:
<https://upei2.cel.agh.edu.pl/wiet/>

2

Wstęp do informatyki

Wyznaczanie oceny końcowej:

1. Warunkiem otrzymania oceny końcowej jest otrzymanie pozytywnych ocen z ćwiczeń i egzaminu.
2. Warunkiem przystąpienia do egzaminu jest uzyskanie pozytywnej oceny z ćwiczeń.
3. Ocenę końcową obliczamy ze wzoru $OK = \max(\text{int}(2 \cdot SR + 0.5)/2, 3.0)$ gdzie SR jest średnią arytmetyczną z ocen zaliczenia ćwiczeń i egzaminów uzyskanych we wszystkich terminach.
4. Jeżeli pozytywną ocenę z ćwiczeń i egzaminu uzyskano w pierwszym terminie oraz ocena końcowa jest mniejsza niż 5.0 to ocena końcowa jest podnoszona o 0.5

3

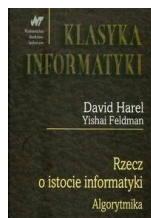
Zakres przedmiotu

- Pojęcia podstawowe
- Mechanizmy języka strukturalnego
- Skalarne typy danych w językach programowania
- Strukturalne typy danych w językach programowania
- Procedury i funkcje - przekazywanie parametrów
- Rekurencja
- Typ wskaźnikowy
- Zastosowania typu wskaźnikowego
- Pojęcie złożoności obliczeniowej
- Przykłady struktur danych
- Przykłady algorytmów
- Reprezentacja liczb w komputerze
- Pojęcia architektury komputera, paradygmatów programowania

4

Literatura

- D. Harel „Rzecz o istocie informatyki - algorytmika”
- J.G. Brookshear „Informatyka w ogólnym zarysie”
- J. Mieścicki „Wstęp do informatyki nie tylko dla informatyków”
- N. Wirth „Algorytmy + struktury danych = programy”
- J. Bentley „Perełki oprogramowania”
- J. Bentley „Więcej perełek oprogramowania”
- T.H. Cormen „Wprowadzenie do algorytmów”
- D.E. Knuth „Sztuka programowania”
- Mark Lutz „Python. Wprowadzenie”



5

Informatyka (Computer Science)

Informatyka to nauka o przetwarzaniu informacji.

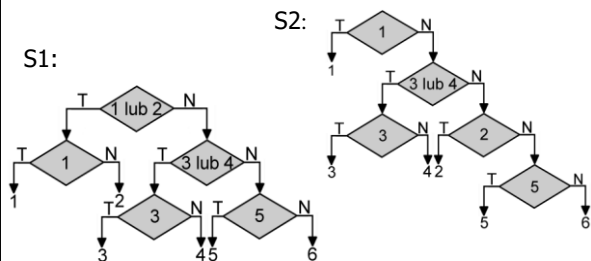
Aktualnie obejmuje wiele zagadnień, między innymi:

- algorytmika, struktury danych, języki programowania
- mikroprocesory, architektury komputerów
- bazy danych, systemy operacyjne, sieci komputerowe
- kompilatory, kryptografia, metody numeryczne
- inżynieria oprogramowania, projektowanie systemów
- grafika, sztuczna inteligencja, aplikacje internetowe
- złożoność obliczeniowa
- ...

6

Identyfikacja elementów zbioru

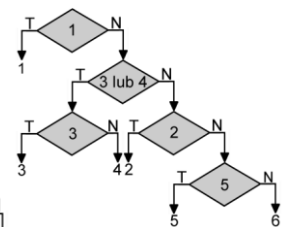
S1:



$$E(S1) = 1/6 \cdot 2 + 1/6 \cdot 2 + 1/6 \cdot 3 + 1/6 \cdot 3 + 1/6 \cdot 3 + 1/6 \cdot 3 = 2.66$$

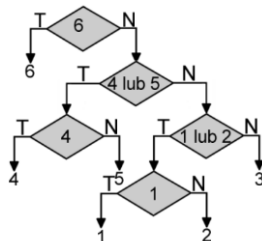
$$E(S2) = 1/6 \cdot 1 + 1/6 \cdot 3 + 1/6 \cdot 3 + 1/6 \cdot 3 + 1/6 \cdot 4 + 1/6 \cdot 4 = 3$$

S2:



Identyfikacja elementów zbioru

S3:



$$P(6) = 0.95$$

$$P(1-5) = 0.01$$

$$E(S1) = 0.01 \cdot 2 + 0.01 \cdot 2 + 0.01 \cdot 3 + 0.01 \cdot 3 + 0.01 \cdot 3 + 0.95 \cdot 3 = 2.98$$

$$E(S3) = 0.01 \cdot 4 + 0.01 \cdot 4 + 0.01 \cdot 3 + 0.01 \cdot 3 + 0.01 \cdot 3 + 0.95 \cdot 1 = 1.12$$

7

8

Teoria informacji

Ilość informacji zawarta w danym zbiorze jest miarą stopnia trudności rozpoznania elementów tego zbioru.

Ilością informacji zawartej w zbiorze $X = \{x_1, x_2, \dots, x_N\}$, nazywamy liczbę:

$$H(X) = - \sum_{i=1}^N p_i \cdot \log_2(p_i)$$



Claude Shannon

gdzie:

p_i ($p_i > 0$, $\sum p_i = 1$) jest prawdopodobieństwem wystąpienia elementu x_i

Przykłady:

$\{0,1\}$ 1 bit
 $\{0..9\}$ 3.32 bita
 zbiór liter języka ang. 4.7 bita

$$-\frac{1}{2} \log_2 \frac{1}{2} = 1$$

$$\left\{ \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4} \right\} \rightarrow ?$$

9

Dane dzisiaj

1024 Bajty = 1 Kilobajt
 1024 Kilobajty = 1 Megabajt
 1024 Megabajty = 1 Gigabajt
 1024 Gigabajty = 1 Terabajt
 1024 Terabajty = 1 Petabajt
 1024 Petabajty = 1 Eksabajt
 1024 Eksabajty = 1 Zettabajt
 1024 Zettabajty = 1 Jottabajt

Kilobajt

Pomiędzy zaraniem cywilizacji, a rokiem 2003 zostało stworzonych 5 Eksabajtów informacji, dziś tworzymy tyle danych w 2 dni.

Prezes Google, Eric Schmidt, 2010 Konferencja Google Atmosphere

10

Cała wiedza świata

- W 2000 r. tylko 25 proc. wszystkich danych zebranych na świecie była w **formie cyfrowej** (reszta na papierze, taśmie filmowej, płytach winylowych, kasetach itp.).
- W 2013 r. w cyfrowej formie zapisane zostało 98 proc. wszystkich danych zgromadzonych na świecie – 1200 eksabajtów (EB).
- Rok 2020 – każdy z nas wytwarza 1.7 MB/s, łączna liczba danych to 45 zettabajtów.
- IBM szacuje, że 90% istniejących danych zostało wytworzone w ostatnich 2 latach.

11

Kompresja danych

Ciąg danych

AABACADABA

1) Kodowanie proste

A - 00
 B - 01
 C - 10
 D - 11

2) Kodowanie Huffmana

A - 0
 B - 10
 C - 110
 D - 111

0.6
 0.2
 0.1
 0.1

Po zakodowaniu

1) 00 00 01 00 10 00 11 00 01 00
 2) 0 0 10 0 110 0 111 0 10 0

20 bitów
 16 bitów

Ilość informacji

$$H(X) = 0.6 \cdot \log(0.6) + 0.2 \cdot \log(0.2) + 0.1 \cdot \log(0.1) + 0.1 \cdot \log(0.1) = 15.7$$

12

Podstawowe pojęcia

Zadanie algorytmiczne – polega na określeniu:

- wszystkich poprawnych danych wejściowych
- oczekiwanych wyników jako funkcji danych wejściowych

Algorytm - specyfikacja ciągu elementarnych operacji, które przekształcają dane wejściowe na wyniki.

Algorytm można przedstawić w postaci:

- werbalnej (*opis słowny*)
- symbolicznej (*schemat blokowy*)
- programu

13

Przykład zapisu algorytmu

Problem: równanie kwadratowe

Dane: współczynniki **a, b, c**

Wyjście: pierwiastki **x1, x2** albo informacja o ich braku

Postać werbalna algorytmu:

„Mając dane współczynniki a, b, c :

oblicz $d = b^2 - 4ac$

Jeżeli d jest nieujemne :

oblicz $p = \sqrt{d}$

oblicz $x1 = (-b-p)/(2a)$

oblicz $x2 = (-b+p)/(2a)$

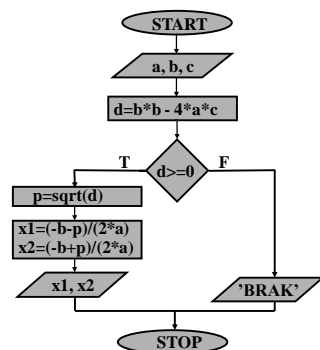
wypisz wartości x1, x2

Jeżeli d jest ujemne :

wypisz "BRAK PIERWIĄSTKÓW"

14

Zapis symboliczny a program



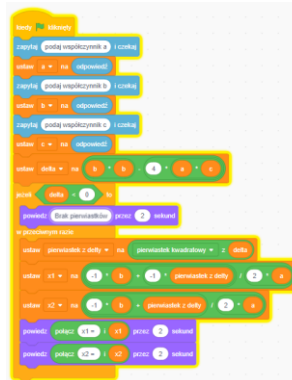
```
from math import sqrt
```

```

a=float(input("a="))
b=float(input("b="))
c=float(input("c="))
d=b*b-4*a*c
if d>=0:
    p=sqrt(d)
    x1=(-b-p)/(2*a)
    x2=(-b+p)/(2*a)
    print("x1=",x1)
    print("x2=",x2)
else:
    print("BRAK")
  
```

15

Wizualny język programowania



16

Algorytm Euklidesa

Algorytm Euklidesa (około 300 r. p.n.e.)
obliczający największy wspólny dzielnik.

Dane: liczby naturalne a, b

Wynik: NWD(a, b)



```

a = int(input("a="))
b = int(input("b="))
while a!=b:
    if a>b:
        a = a-b
    else:
        b = b-a
print(a)
  
```

a	b
24	30
24	6
18	6
12	6
6	6

17

Przykład zadania

Problem: Rozkład liczby na czynniki pierwsze

Proszę napisać program, który dla wczytanej liczby naturalnej wypisuje jej rozkład na czynniki pierwsze.

Przykład:

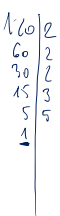
120 : 2, 2, 2, 3, 5

18

Rozkład na czynniki pierwsze

Rozwiązanie pierwsze

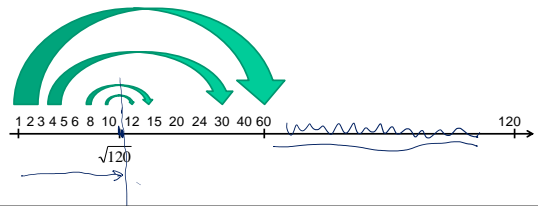
```
end = Nowe
n = int(input("n="))
b = 2
while n>1:
    if n % b == 0:
        print(b)
        n = n // b
    else:
        b = b+1
    # end if
# end while
print("koniec")
```



19

Położenie podzielników liczby

Podzielniki liczby 120

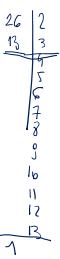


20

Rozkład na czynniki pierwsze

Rozwiązanie drugie

```
n = int(input("n="))
b = 2
while n>1 and b<=sqrt(n):
    if n % b == 0:
        print(b)
        n = n // b
    else:
        b = b+1
    # end if
# end while
if n>1: print(n)
print("koniec")
```



21

Porównanie rozwiązań

Liczba cyfr (Najwyżej 14)	Algorytm pierwszy	Algorytm drugi
6	0.07s	0.0s
7	0.58s	0.0s
8	7.75s	0.0s
9	1m7.2s	0.01s
10	9m43s	0.01s
11	1h23m	0.04s
12	12h27m	0.13s
13	4d9h	0.42s
14	37d12h	1.23s

2^100 ≈ 10^33

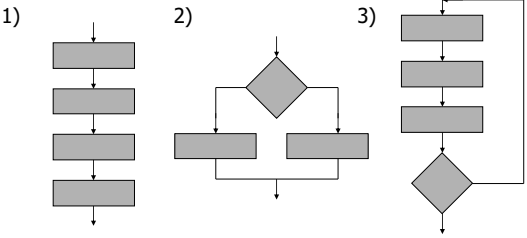
50 cyf
1,23 * 3,3^36

Czy można jeszcze szybciej?

22

Budowa algorytmów

- 1) Bezpośrednie następstwo
- 2) Wybór warunkowy
- 3) Iteracja warunkowa



23

Pytania i zadania

- Ile informacji zawiera 10 znakowe słowo, którego każdy znak z jednakowym prawdopodobieństwem jest jedną z liter a, b, c ?
- Jak stworzyć kody Huffmana dla zbiorów 5,6,7,... elementowych?
- Jak przyspieszyć działanie algorytmu Euklidesa?
- Ile czasu potrzebuje w najgorszym przypadku drugi algorytm aby rozłożyć na czynniki 50 cyfrową liczbę?
- Jak przyspieszyć działanie programu rozkładu na czynniki pierwsze?
- Proszę zaproponować algorytm rozkładający liczbę naturalną N na sumę składników, tak aby ich iloczyn był największy, np. 5 -> 2*3, 7 -> 2*2*3, 9 -> 3*3*3.

24