



# Algorytmy (Część I)

wer. 13 z drobnymi modyfikacjami!

Wojciech Myszka

2023-11-13 15:43:16 +0100



HR EXCELLENCE IN RESEARCH



Politechnika Wroclawska

# Krótkie podsumowanie

Wiemy:



# Krótkie podsumowanie

Wiemy:

1. co to jest komputer;



# Krótkie podsumowanie

Wiemy:

1. co to jest komputer;
2. z jakich elementów jest skonstruowany;



# Krótkie podsumowanie

Wiemy:

1. co to jest komputer;
2. z jakich elementów jest skonstruowany;
3. co to jest procesor...



# Krótkie podsumowanie

Wiemy:

1. co to jest komputer;
2. z jakich elementów jest skonstruowany;
3. co to jest procesor...
4. i jak działa;



# Krótkie podsumowanie

Wiemy:

1. co to jest komputer;
2. z jakich elementów jest skonstruowany;
3. co to jest procesor...
4. i jak działa;
5. troszeczkę o tym, że komputery potrzebują programów;



# Krótkie podsumowanie

Wiemy:

1. co to jest komputer;
2. z jakich elementów jest skonstruowany;
3. co to jest procesor...
4. i jak działa;
5. troszeczkę o tym, że komputery potrzebują programów;
6. co to jest program.





# Krótkie podsumowanie

Wiemy:

1. co to jest komputer;
2. z jakich elementów jest skonstruowany;
3. co to jest procesor...
4. i jak działa;
5. troszeczkę o tym, że komputery potrzebują programów;
6. co to jest program.



# Krótkie podsumowanie

Wiemy:

1. co to jest komputer;
2. z jakich elementów jest skonstruowany;
3. co to jest procesor...
4. i jak działa;
5. troszeczkę o tym, że komputery potrzebują programów;
6. co to jest program.

Nie bardzo wiemy:



# Krótkie podsumowanie

Wiemy:

1. co to jest komputer;
2. z jakich elementów jest skonstruowany;
3. co to jest procesor...
4. i jak działa;
5. troszeczkę o tym, że komputery potrzebują programów;
6. co to jest program.

Nie bardzo wiemy:

1. skąd się bierze program...



# Jak powstaje program

Jest jakoś tak:



# Jak powstaje program

Jest jakoś tak:



# Jak powstaje program

Jest jakoś tak:

- ▶ Rozmowa z klientem (odbiorcą programu)



# Jak powstaje program

Jest jakoś tak:

- ▶ Rozmowa z klientem (odbiorcą programu)
- ▶ Projekt i algorytm(y)



# Jak powstaje program

Jest jakoś tak:

- ▶ Rozmowa z klientem (odbiorcą programu)
- ▶ Projekt i algorytm(y)
- ▶ Programowanie





# Jak powstaje program

Jest jakoś tak:

- ▶ Rozmowa z klientem (odbiorcą programu)
- ▶ Projekt i algorytm(y)
- ▶ Programowanie
- ▶ Testy



# Jak powstaje program

Jest jakoś tak:

- ▶ Rozmowa z klientem (odbiorcą programu)
- ▶ Projekt i algorytm(y)
- ▶ Programowanie
- ▶ Testy
- ▶ Dokumentacja



# Jak powstaje program

Jest jakoś tak:

- ▶ Rozmowa z klientem (odbiorcą programu)
- ▶ Projekt i algorytm(y)
- ▶ Programowanie
- ▶ Testy
- ▶ Dokumentacja
- ▶ Instalacja u klienta



# Jak powstaje program

Jest jakoś tak:

- ▶ Rozmowa z klientem (odbiorcą programu)
- ▶ Projekt i algorytm(y)
- ▶ Programowanie
- ▶ Testy
- ▶ Dokumentacja
- ▶ Instalacja u klienta
- ▶ Eksploatacja/Wsparcie



# Jak powstaje program

Jest jakoś tak:

- ▶ Rozmowa z klientem (odbiorcą programu)
- ▶ Projekt i algorytm(y)
- ▶ Programowanie
- ▶ Testy
- ▶ Dokumentacja
- ▶ Instalacja u klienta
- ▶ Eksploatacja/Wsparcie
- ▶ ...



# Jak powstaje program

Jest jakoś tak:

- ▶ Rozmowa z klientem (odbiorcą programu)
- ▶ Projekt i algorytm(y)
- ▶ Programowanie
- ▶ Testy
- ▶ Dokumentacja
- ▶ Instalacja u klienta
- ▶ Eksploatacja/Wsparcie
- ▶ ...



# Algorytm

Słowo „algorytm” jest bardzo nowe (w pewnym sensie).



# Algorytm

Słowo „algorytm” jest bardzo nowe (w pewnym sensie).

Pochodzi od nazwiska Muḥammad ibn Mūsā al-Khwārizmī — perskiego matematyka (IX w.) i pierwotnie oznaczało (każde) obliczenia w dziesiętnym systemie obliczeniowym.





# Algorytm

Słowo „algorytm” jest bardzo nowe (w pewnym sensie).

Pochodzi od nazwiska Muḥammad ibn Mūsā al-Khwārizmī — perskiego matematyka (IX w.) i pierwotnie oznaczało (każde) obliczenia w dziesiętnym systemie obliczeniowym.

Algorytm to **skończony** ciąg **jasno** zdefiniowanych czynności koniecznych do wykonania pewnego rodzaju zadań, sposób postępowania **prowadzący** do rozwiązania problemu. ([Wikipedia](#))



# Algorytm

Słowo „algorytm” jest bardzo nowe (w pewnym sensie).

Pochodzi od nazwiska Muḥammad ibn Mūsā al-Khwārizmī — perskiego matematyka (IX w.) i pierwotnie oznaczało (każde) obliczenia w dziesiętnym systemie obliczeniowym.

Algorytm to **skończony** ciąg **jasno** zdefiniowanych czynności koniecznych do wykonania pewnego rodzaju zadań, sposób postępowania **prowadzący** do rozwiązania problemu. ([Wikipedia](#))

Czasami rezygnuje się z żądania **skończoności**. Czasami, jeżeli algorytm się nie kończy — nazywamy go **metodą obliczeniową**.



# Algorithm

In mathematics and computer science, an algorithm is a **finite sequence**



# Algorithm

In mathematics and computer science, an algorithm is a **finite sequence of well-defined,**



# Algorithm

In mathematics and computer science, an algorithm is a **finite sequence of well-defined, computer-implementable**



# Algorithm

In mathematics and computer science, an algorithm is a **finite sequence of well-defined, computer-implementable instructions**, typically to solve a **class of problems**



# Algorithm

In mathematics and computer science, an algorithm is a **finite sequence of well-defined, computer-implementable** instructions, typically to solve a **class of problems** or to perform a computation. Algorithms are always **unambiguous**



# Algorithm

In mathematics and computer science, an algorithm is a **finite sequence** of **well-defined, computer-implementable** instructions, typically to solve a **class of problems** or to perform a computation. Algorithms are always **unambiguous** and are used as specifications for performing calculations, data processing, automated reasoning, and other tasks.

As an effective method, an algorithm can be expressed within a **finite** amount of space and time,





# Algorithm

In mathematics and computer science, an algorithm is a **finite sequence** of **well-defined, computer-implementable** instructions, typically to solve a **class of problems** or to perform a computation. Algorithms are always **unambiguous** and are used as specifications for performing calculations, data processing, automated reasoning, and other tasks.

As an effective method, an algorithm can be expressed within a **finite** amount of space and time, and in a **well-defined** formal language



# Algorithm

In mathematics and computer science, an algorithm is a **finite sequence** of **well-defined, computer-implementable** instructions, typically to solve a **class of problems** or to perform a computation. Algorithms are always **unambiguous** and are used as specifications for performing calculations, data processing, automated reasoning, and other tasks.

As an effective method, an algorithm can be expressed within a **finite** amount of space and time, and in a **well-defined** formal language for calculating a function. Starting from an initial state and initial input (perhaps empty), the instructions describe a computation that, when executed, proceeds through a finite number of well-defined successive states, eventually producing “output” and terminating at a final ending state. The transition from one state to the next is not necessarily deterministic; some algorithms, known as randomized algorithms, incorporate random input.



# Algorytm I

## Przepis kucharski

- ▶ **Składniki:** 22 dag twardej czekolady półsłodkiej, 2 łyżki stołowe wody,  $\frac{1}{4}$  filiżanki cukru pudru, 6 jajek rozdzielonych na żółtka i białka...
- ▶ **Przepis:**  
„Włóż czekoladę z dwiema łyżkami stołowymi wody do garnka o podwójnym dnie. Kiedy czekolada się rozpuści, domieszaj cukier puder; dodaj po trochu masło. Odstaw. Ubijaj żółtka około 5 minut, aż staną się gęste i nabiorą koloru cytrynowego. Delikatnie dołóż czekoladę. Ponownie lekko podgrzej, aby rozpuścić czekoladę, jeśli to będzie konieczne. Domieszaj rum i wanilię. Ubijaj



# Algorytm II

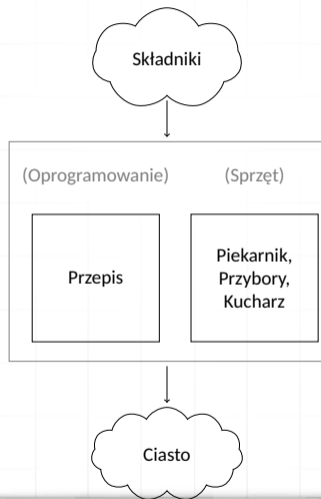
Przepis kucharski

białka aż do spienienia. Ubijając dodaj 2 łyżki stołowe cukru i ubijaj dalej, aż utworzą się sztywne pagórki. Delikatnie połącz białka z masą czekoladowo-żółtkową. Wlej do oddzielnych naczyń, które będą podane na stół. Ochładzaj przez co najmniej 4 godziny. Wedle życzenia podawaj z bitą śmietaną. Wyjdzie z tego 6 do 8 porcji.”

Na podstawie *Kuchnia francuska* [2, 1]



# Przepis kucharski



# Krotki algorytm

- ▶ Mamy listę (bazę danych) zawierającą rekordy wszystkich pracowników (każdy rekord zawiera nazwisko, inne dane personalne i płacę).
- ▶ Interesuje nas suma zarobków wszystkich pracowników.
- ▶ Poniżej algorytm realizujący zadanie:
  1. zapisz na boku liczbę 0;
  2. przeglądaj listę pracowników dodając płacę każdego pracownika do zanotowanej liczby;
  3. gdy dojdiesz do końca listy pracowników — wydrukuj zanotowany wynik.



# Prosty algorytm

Kilka uwag

1. Czy algorytm jest poprawny?



# Prosty algorytm

Kilka uwag

1. Czy algorytm jest poprawny?
2. *Zapis* algorytmu jest krótki i jego długość się nie zmienia.





# Prosty algorytm

Kilka uwag

1. Czy algorytm jest poprawny?
2. *Zapis* algorytmu jest krótki i jego długość się nie zmienia.
3. *Proces* opisywany przez algorytm (realizacja algorytmu) zmienia się wraz z długością danych.



# Prosty algorytm

Kilka uwag

1. Czy algorytm jest poprawny?
2. *Zapis* algorytmu jest krótki i jego długość się nie zmienia.
3. *Proces* opisywany przez algorytm (realizacja algorytmu) zmienia się wraz z długością danych.
4. Algorytm rozwiązuje **każdy problem** „tej klasy”: dwie firmy, pierwsza zatrudniająca jedną osobę i druga zatrudniająca milion osób, mogą z niego korzystać; będzie dobry dla jednej i drugiej.



# Prosty algorytm

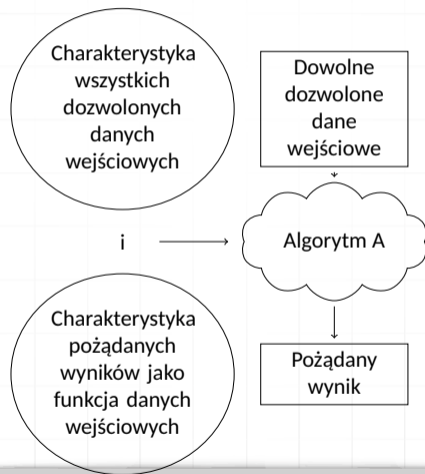
Kilka uwag

1. Czy algorytm jest poprawny?
2. *Zapis* algorytmu jest krótki i jego długość się nie zmienia.
3. *Proces* opisywany przez algorytm (realizacja algorytmu) zmienia się wraz z długością danych.
4. Algorytm rozwiązuje **każdy problem** „tej klasy”: dwie firmy, pierwsza zatrudniająca jedną osobę i druga zatrudniająca milion osób, mogą z niego korzystać; będzie dobry dla jednej i drugiej.
5. ...



# Rozwiązanie algorytmiczne

Za [1]



# Algorytm Euklidesa

Oto jedna z jego wersji algorytmu Euklidesa:



# Algorytm Euklidesa

Oto jedna z jego wersji algorytmu Euklidesa:

*Dane są dwie dodatnie liczby całkowite  $m$  i  $n$ , należy znaleźć ich **największy wspólny dzielnik (NWD)** tj. największą dodatnią liczbę całkowitą, która dzieli całkowicie zarówno  $m$  jak i  $n$ .*



# Algorytm Euklidesa

Oto jedna z jego wersji algorytmu Euklidesa:

Dane są dwie dodatnie liczby całkowite  $m$  i  $n$ , należy znaleźć ich **największy wspólny dzielnik (NWD)** tj. największą dodatnią liczbę całkowitą, która dzieli całkowicie zarówno  $m$  jak i  $n$ .

1. [Znajdowanie reszty] Podziel  $m$  przez  $n$  i niech  $r$  oznacza resztę z tego dzielenia. (Mamy  $0 \leq r < n$ .)



# Algorytm Euklidesa

Oto jedna z jego wersji algorytmu Euklidesa:

Dane są dwie dodatnie liczby całkowite  $m$  i  $n$ , należy znaleźć ich **największy wspólny dzielnik (NWD)** tj. największą dodatnią liczbę całkowitą, która dzieli całkowicie zarówno  $m$  jak i  $n$ .

1. [Znajdowanie reszty] Podziel  $m$  przez  $n$  i niech  $r$  oznacza resztę z tego dzielenia. (Mamy  $0 \leq r < n$ .)
2. [Czy wyszło zero?] Jeśli  $r = 0$  zakończ algorytm; odpowiedzią jest  $n$ .





# Algorytm Euklidesa

Oto jedna z jego wersji algorytmu Euklidesa:

Dane są dwie dodatnie liczby całkowite  $m$  i  $n$ , należy znaleźć ich **największy wspólny dzielnik (NWD)** tj. największą dodatnią liczbę całkowitą, która dzieli całkowicie zarówno  $m$  jak i  $n$ .

1. [Znajdowanie reszty] Podziel  $m$  przez  $n$  i niech  $r$  oznacza resztę z tego dzielenia. (Mamy  $0 \leq r < n$ .)
2. [Czy wyszło zero?] Jeśli  $r = 0$  zakończ algorytm; odpowiedzią jest  $n$ .
3. [Upraszczanie] Wykonaj  $m \leftarrow n$ ,  $n \leftarrow r$  i wróć do kroku 1.



# Dzielenie z resztą

Dygresja

$$\frac{5}{3} =$$



# Dzielenie z resztą

Dygresja

$$\frac{5}{3} = 1\frac{2}{3}$$



# Dzielenie z resztą

Dygresja

$$\frac{5}{3} = 1\frac{2}{3} = 1,6666(6)$$



# Dzielenie z resztą

Dygresja

$$\frac{5}{3} = 1\frac{2}{3} = 1,6666(6) = 1,6667^{0,0001}$$



# Dzielenie z resztą

Dygresja

$$\frac{5}{3} = 1\frac{2}{3} = 1,6666(6) = 1,6667^{0,0001} = 1 \text{ reszta } 2$$



# Dzielenie z resztą

Dygresja

$$\frac{5}{3} = 1\frac{2}{3} = 1,6666(6) = 1,6667^{0,0001} = 1 \text{ reszta } 2$$

bo

$$1 \times 3 + 2 = 5$$



# Algorytm Euklidesa

## Przykład

m	n	r
24	44	

1. [Znajdowanie reszty] Podziel  $m$  przez  $n$  i niech  $r$  oznacza resztę z tego dzielenia. (Mamy  $0 \leq r < n$ .)
2. [Czy wyszło zero?] Jeśli  $r = 0$  zakończ algorytm; odpowiedzią jest  $n$ .
3. [Upraszczanie] Wykonaj  $m \leftarrow n, n \leftarrow r$  i wróć do kroku 1.





# Algorytm Euklidesa

Przykład

m	n	r
24	44	24

$$24/44 = 0 \text{ r } 24$$

1. [Znajdowanie reszty] Podziel  $m$  przez  $n$  i niech  $r$  oznacza resztę z tego dzielenia. (Mamy  $0 \leq r < n$ .)
2. [Czy wyszło zero?] Jeśli  $r = 0$  zakończ algorytm; odpowiedzią jest  $n$ .
3. [Upraszczanie] Wykonaj  $m \leftarrow n, n \leftarrow r$  i wróć do kroku 1.



# Algorytm Euklidesa

## Przykład

m	n	r
24	44	24

1. [Znajdowanie reszty] Podziel  $m$  przez  $n$  i niech  $r$  oznacza resztę z tego dzielenia. (Mamy  $0 \leq r < n$ .)
2. [Czy wyszło zero?] Jeśli  $r = 0$  zakończ algorytm; odpowiedzią jest  $n$ .
3. [Upraszczanie] Wykonaj  $m \leftarrow n$ ,  $n \leftarrow r$  i wróć do kroku 1.



# Algorytm Euklidesa

Przykład

m	n	r
24	44	24
44		

1. [Znajdowanie reszty] Podziel  $m$  przez  $n$  i niech  $r$  oznacza resztę z tego dzielenia. (Mamy  $0 \leq r < n$ .)
2. [Czy wyszło zero?] Jeśli  $r = 0$  zakończ algorytm; odpowiedzią jest  $n$ .
3. [Upraszczanie] Wykonaj  $m \leftarrow n$ ,  $n \leftarrow r$  i wróć do kroku 1.



# Algorytm Euklidesa

## Przykład

m	n	r
24	44	24
44	24	

1. [Znajdowanie reszty] Podziel  $m$  przez  $n$  i niech  $r$  oznacza resztę z tego dzielenia. (Mamy  $0 \leq r < n$ .)
2. [Czy wyszło zero?] Jeśli  $r = 0$  zakończ algorytm; odpowiedzią jest  $n$ .
3. [Upraszczanie] Wykonaj  $m \leftarrow n, n \leftarrow r$  i wróć do kroku 1.



# Algorytm Euklidesa

## Przykład

m	n	r
24	44	24
44	24	20

$$44/24 = 1 \text{ r } 20$$

1. [Znajdowanie reszty] Podziel  $m$  przez  $n$  i niech  $r$  oznacza resztę z tego dzielenia. (Mamy  $0 \leq r < n$ .)
2. [Czy wyszło zero?] Jeśli  $r = 0$  zakończ algorytm; odpowiedzią jest  $n$ .
3. [Upraszczenie] Wykonaj  $m \leftarrow n$ ,  $n \leftarrow r$  i wróć do kroku 1.



# Algorytm Euklidesa

## Przykład

m	n	r
24	44	24
44	24	20

1. [Znajdowanie reszty] Podziel  $m$  przez  $n$  i niech  $r$  oznacza resztę z tego dzielenia. (Mamy  $0 \leq r < n$ .)
2. [Czy wyszło zero?] Jeśli  $r = 0$  zakończ algorytm; odpowiedzią jest  $n$ .
3. [Upraszczenie] Wykonaj  $m \leftarrow n, n \leftarrow r$  i wróć do kroku 1.



# Algorytm Euklidesa

## Przykład

m	n	r
24	44	24
44	24	20
24		

1. [Znajdowanie reszty] Podziel  $m$  przez  $n$  i niech  $r$  oznacza resztę z tego dzielenia. (Mamy  $0 \leq r < n$ .)
2. [Czy wyszło zero?] Jeśli  $r = 0$  zakończ algorytm; odpowiedzią jest  $n$ .
3. [Upraszczenie] Wykonaj  $m \leftarrow n, n \leftarrow r$  i wróć do kroku 1.



# Algorytm Euklidesa

## Przykład

m	n	r
24	44	24
44	24	20
24	20	

1. [Znajdowanie reszty] Podziel  $m$  przez  $n$  i niech  $r$  oznacza resztę z tego dzielenia. (Mamy  $0 \leq r < n$ .)
2. [Czy wyszło zero?] Jeśli  $r = 0$  zakończ algorytm; odpowiedzią jest  $n$ .
3. [Upraszczenie] Wykonaj  $m \leftarrow n$ ,  $n \leftarrow r$  i wróć do kroku 1.





# Algorytm Euklidesa

## Przykład

m	n	r
24	44	24
44	24	20
24	20	4

$$24/20 = 1 \text{ r } 4$$

1. [Znajdowanie reszty] Podziel  $m$  przez  $n$  i niech  $r$  oznacza resztę z tego dzielenia. (Mamy  $0 \leq r < n$ .)
2. [Czy wyszło zero?] Jeśli  $r = 0$  zakończ algorytm; odpowiedzią jest  $n$ .
3. [Upraszczanie] Wykonaj  $m \leftarrow n$ ,  $n \leftarrow r$  i wróć do kroku 1.



# Algorytm Euklidesa

## Przykład

m	n	r
24	44	24
44	24	20
24	20	4

1. [Znajdowanie reszty] Podziel  $m$  przez  $n$  i niech  $r$  oznacza resztę z tego dzielenia. (Mamy  $0 \leq r < n$ .)
2. [Czy wyszło zero?] Jeśli  $r = 0$  zakończ algorytm; odpowiedzią jest  $n$ .
3. [Upraszczanie] Wykonaj  $m \leftarrow n$ ,  $n \leftarrow r$  i wróć do kroku 1.



# Algorytm Euklidesa

## Przykład

m	n	r
24	44	24
44	24	20
24	20	4
20		

1. [Znajdowanie reszty] Podziel  $m$  przez  $n$  i niech  $r$  oznacza resztę z tego dzielenia. (Mamy  $0 \leq r < n$ .)
2. [Czy wyszło zero?] Jeśli  $r = 0$  zakończ algorytm; odpowiedzią jest  $n$ .
3. [Upraszczenie] Wykonaj  $m \leftarrow n, n \leftarrow r$  i wróć do kroku 1.



# Algorytm Euklidesa

## Przykład

m	n	r
24	44	24
44	24	20
24	20	4
20	4	

1. [Znajdowanie reszty] Podziel  $m$  przez  $n$  i niech  $r$  oznacza resztę z tego dzielenia. (Mamy  $0 \leq r < n$ .)
2. [Czy wyszło zero?] Jeśli  $r = 0$  zakończ algorytm; odpowiedzią jest  $n$ .
3. [Upraszczanie] Wykonaj  $m \leftarrow n, n \leftarrow r$  i wróć do kroku 1.



# Algorytm Euklidesa

## Przykład

m	n	r
24	44	24
44	24	20
24	20	4
20	4	0

$20/4 = 5 \text{ r } 0$

1. [Znajdowanie reszty] Podziel  $m$  przez  $n$  i niech  $r$  oznacza resztę z tego dzielenia. (Mamy  $0 \leq r < n$ .)
2. [Czy wyszło zero?] Jeśli  $r = 0$  zakończ algorytm; odpowiedzią jest  $n$ .
3. [Upraszczenie] Wykonaj  $m \leftarrow n, n \leftarrow r$  i wróć do kroku 1.



# Algorytm Euklidesa

## Przykład

m	n	r
24	44	24
44	24	20
24	20	4
20	4	0

1. [Znajdowanie reszty] Podziel  $m$  przez  $n$  i niech  $r$  oznacza resztę z tego dzielenia. (Mamy  $0 \leq r < n$ .)
2. [Czy wyszło zero?] Jeśli  $r = 0$  zakończ algorytm; odpowiedzią jest  $n$ .
3. [Upraszczenie] Wykonaj  $m \leftarrow n$ ,  $n \leftarrow r$  i wróć do kroku 1.



# Zadanie domowe

Zrealizować algorytm Euklidesa w arkuszu kalkulacyjnym (bez korzystania z żadnych języków skryptowych).



# NWD w Blockly

```
set m prompt for number with message Podaj m
set n prompt for number with message Podaj n
repeat until get n = 0
do
  set r remainder of get m ÷ get n
  set m get n
  set n get r
print get m
```





# NWD w Blockly

```
set m prompt for number with message Podaj m
set n prompt for number with message Podaj n
repeat until get n = 0
do
  set r remainder of get m ÷ get n
  set m get n
  set n get r
print get m
```

The image shows a Blockly script for calculating the Greatest Common Divisor (NWD) using the Euclidean algorithm. The script consists of the following blocks:

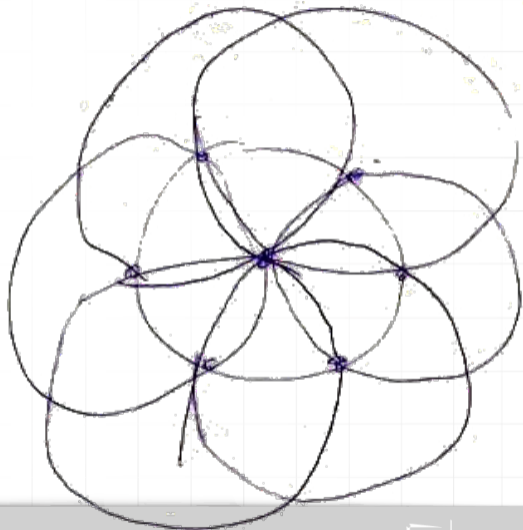
- Two 'set' blocks for variables `m` and `n`, each with a 'prompt for number with message' block and the message 'Podaj m' and 'Podaj n' respectively.
- A 'repeat until' loop with the condition 'get n = 0'. Inside the loop:
  - A 'set r' block with the value 'remainder of get m ÷ get n'.
  - A 'set m' block with the value 'get n'.
  - A 'set n' block with the value 'get r'.
- A 'print' block with the value 'get m'.

Zadanie domowe: wyjaśnić różnice z przedstawionym wcześniej algorytmem.



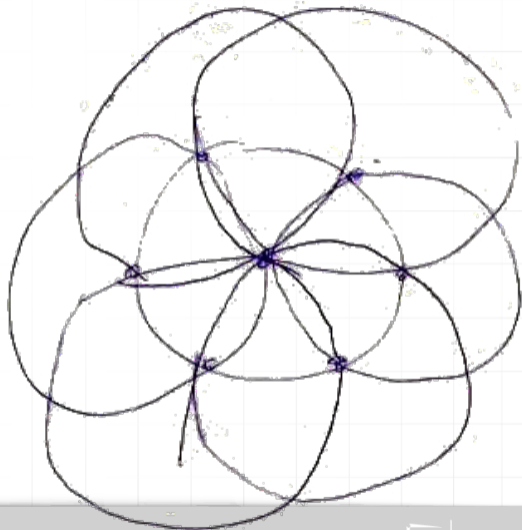
# Problem

- ▶ Ktoś prosił nas o narysowanie „kwiatka” czyli rysunku, wyglądającego jakoś tak:



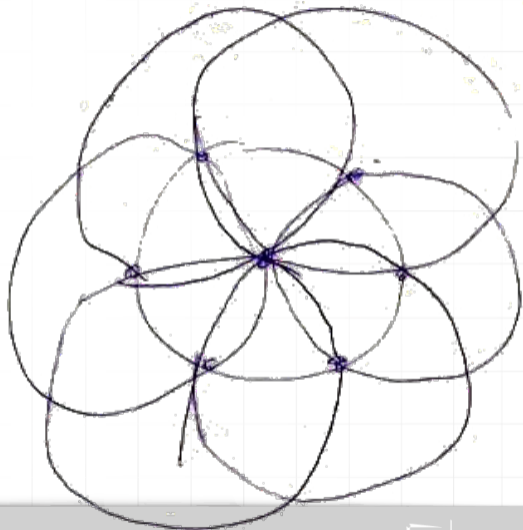
# Problem

- ▶ Ktoś prosił nas o narysowanie „kwiatka” czyli rysunku, wyglądającego jakoś tak:
- ▶ Tylko porządniej...



# Problem

- ▶ Ktoś prosił nas o narysowanie „kwiatka” czyli rysunku, wyglądającego jakoś tak:
- ▶ Tylko porządnie...
- ▶ Jak się za to zabrać?



# Metody

## 1. Cyrkiel...



# Metody

1. Cyrkiel...
2. Jakiś program graficzny (Corel, OpenOffice.org Draw, cokolwiek...)



# Metody

1. Cyrkiel...
2. Jakiś program graficzny (Corel, OpenOffice.org Draw, cokolwiek...)
3. Skonstruować...



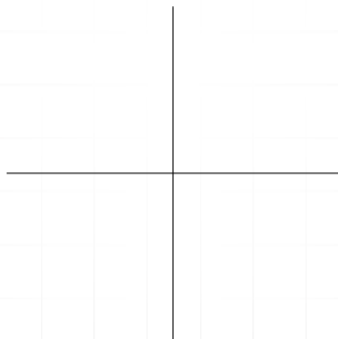
# Metody

1. Cyrkiel...
2. Jakiś program graficzny (Corel, OpenOffice.org Draw, cokolwiek...)
3. Skonstruować...
4. Napisać program komputerowy...

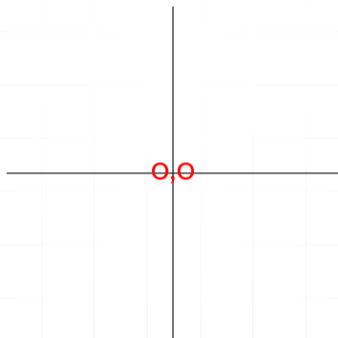




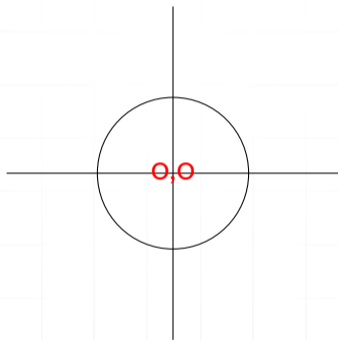
# Kwiatek



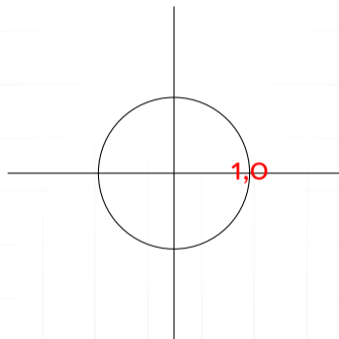
# Kwiatek



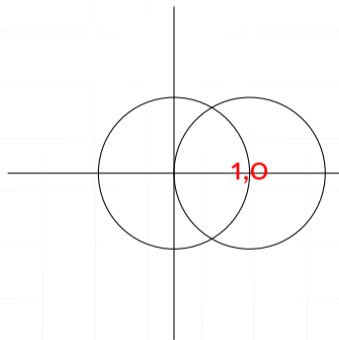
# Kwiatek



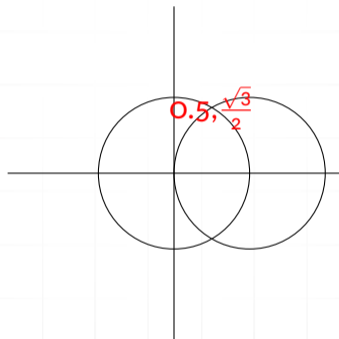
# Kwiatek



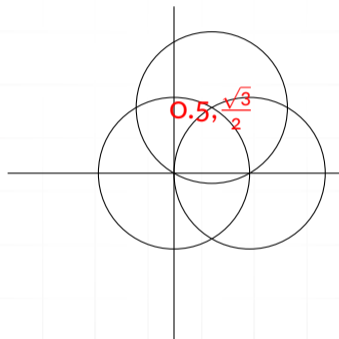
# Kwiatek



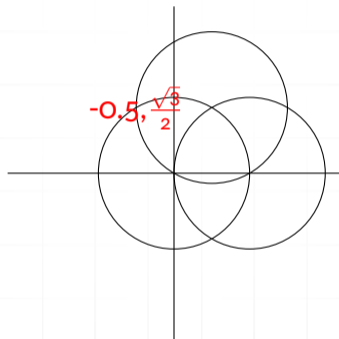
# Kwiatek



# Kwiaterek

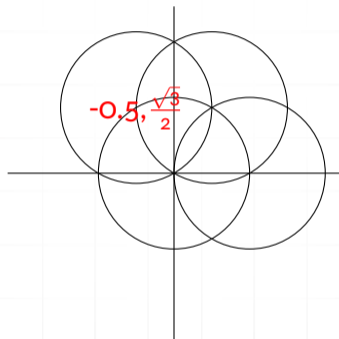


# Kwiatek

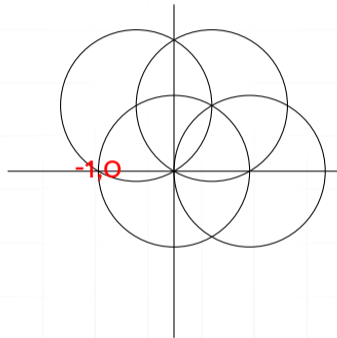




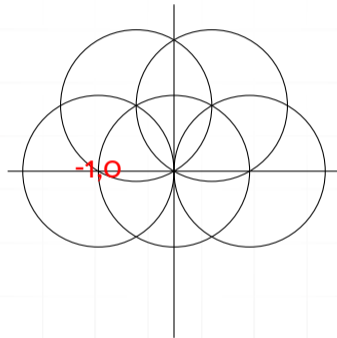
# Kwiatek



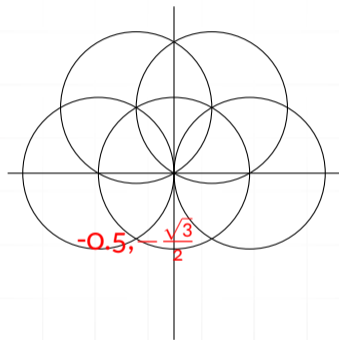
# Kwiatek



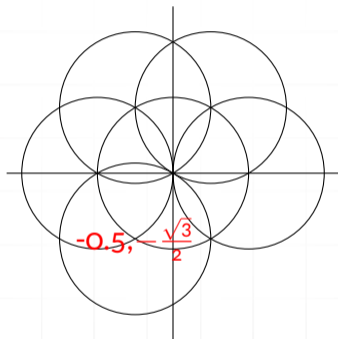
# Kwiatek



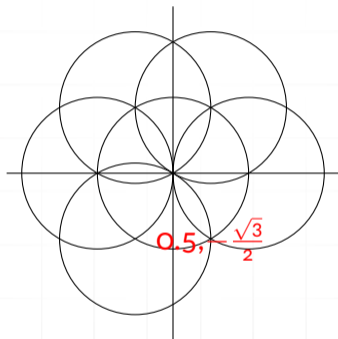
# Kwiatek



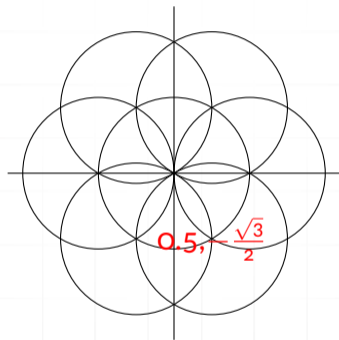
# Kwiatek



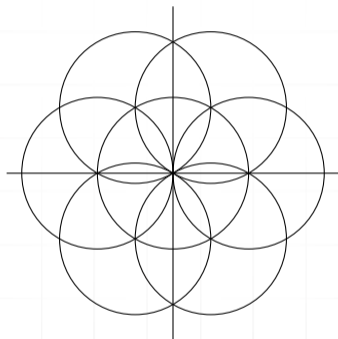
# Kwiatek



# Kwiatek

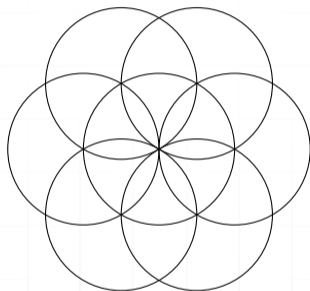


# Kwiatek





# Kwiatek



# Jak to jest zrobione?

```
\draw (0,0) circle (1cm);  
%  
\draw (1,0) circle (1cm);  
\draw (0.5,0.866) circle (1cm);  
\draw (-0.5,0.866) circle (1cm);  
\draw (-1,0) circle (1cm);  
\draw (-0.5,-0.866) circle (1cm);  
\draw (0.5,-0.866) circle (1cm);
```



# Program w Pythonie

```
from turtle import *
setup(600,600,300,300)
title("Kwiatek")
speed(1)
up()
goto(100,0)
down()
setheading(90)
circle(100)
for _ in range(6):
    setheading(heading() - 60)
    down()
    circle(100)
    setheading(heading() + 60)
    up()
    circle(100, 60)

exitonclick()
```




A jak zrobić to?



# Skończoność

Po pierwsze **powinien być skończony**; oznacza to, że po skończonej (być może bardzo dużej) liczbie kroków algorytm się zatrzyma<sup>1</sup>. **Pytanie pomocnicze: Co gwarantuje, że algorytm Euklidesa zakończy się w skończonej liczbie kroków?** Procedura, która ma wszystkie cechy algorytmu poza skończonością nazywana jest *metodą obliczeniową*. **Podaj przykłady metod obliczeniowych realizowanych przez rzeczywiste komputery.**

<sup>1</sup>Ale sama skończoność to jednak za mało — z praktycznego punktu widzenia **dobry** algorytm powinien gwarantować, że obliczenia zostaną zakończone w skończonym ale **rozsądnym** czasie!  Wrocławska

# Cechy algorytmu II

## Dobre zdefiniowanie

Po drugie **powinien być „dobrze zdefiniowany”**. Każdy krok algorytmu musi być opisany precyzyjnie. Wszystkie możliwe przypadki powinny być uwzględnione, a podejmowane akcje dobrze opisane<sup>2</sup>. Oczywiście język naturalny nie jest wystarczająco precyzyjny — może to prowadzić do nieporozumień. z tego powodu używa się bardziej formalnych sposobów zapisu algorytmów, aż po języki programowania...

<sup>2</sup>Zwracam też uwagę, że algorytmy kucharskie nie są odpowiednio precyzyjne: co to znaczy „lekko podgrzej”?



# Cechy algorytmu III

## Dane wejściowe

Po trzecie **powinien mieć precyzyjnie zdefiniowane dane wejściowe**. Pewne algorytmy mogą nie mieć danych wejściowych (mieć zero danych wejściowych). Dane wejściowe to wartości, które muszą być zdefiniowane zanim rozpocznie się wykonanie algorytmu.



# Cechy algorytmu IV

Dane wyjściowe

Po czwarte **zdefiniowane dane wyjściowe**. Daną wyjściową algorytmu Euklidesa jest liczba  $n$  która jest naprawdę największym wspólnym dzielnikiem danych wejściowych. **Osobną sprawą jest pokazanie skąd wynika, że wynik algorytmu Euklidesa jest rzeczywiście NWD liczb  $m$  i  $n$ .**





# Cechy algorytmu V

## Efektywność

Po piątę algorytm **powinien być określony efektywnie** to znaczy jego operacje powinny być wystarczająco proste by można je (teoretycznie?) wykonać w skończonym czasie z wykorzystaniem kartki i ołówka.



# Problem 1

**Wejście:** Dwie liczby  $J$  i  $K$

**Wynik:** Liczba  $J^2 + 3K$

Prosty problem wymagający wykonania elementarnych obliczeń arytmetycznych na dwu liczbach wejściowych.



## Problem 2

**Wejście:** Liczba dodatnia  $K$

**Wynik:** Suma liczb całkowitych od 1 do  $K$

Problem arytmetyczny, ale liczba obliczeń zmienia się w zależności od danych wejściowych.



## Problem 3

**Wejście:** Liczba dodatnia  $K$

**Wynik:** „TAK” gdy  $K$  jest liczbą pierwszą, „NIE”, gdy taką nie jest.

Problem decyzyjny. Zadanie jest również arytmetyczne, ale wynik nie jest numeryczny.



## Problem 4

**Wejście:** Lista  $L$  słów w języku polskim.

**Wynik:** Lista  $L$  uporządkowana alfabetycznie (leksykograficznie)

Nie jest to problem arytmetyczny, konieczne jest operowanie na zmiennej liczbie danych — słów.



# Problem 5

**Wejście:** Mapa drogowa z zaznaczonymi miastami i odległościami między nimi, na której są wyróżnione dwa miasta A i B.

**Wynik:** Opis najkrótszej ścieżki (trasy) od A do B

Problem poszukiwania wśród wszystkich „najlepszego” wariantu rozwiązania



# Problem 6

**Wejście:** Zbiór osób znajdujących się w pomieszczeniu.

**Wynik:** Wysokość osoby najwyższej (najniższej).



# Problem 6

**Wejście:** Zbiór osób znajdujących się w pomieszczeniu.

**Wynik:** Wysokość osoby najwyższej (najniższej).

**lub** Wskazanie osoby najwyższej





# Problem 6

**Wejście:** Zbiór osób znajdujących się w pomieszczeniu.

**Wynik:** Wysokość osoby najwyższej (najniższej).

**lub** Wskazanie osoby najwyższej

Do czego sprowadza się różnica między tymi algorytmami?



# Przykład 7

## Appendix -Troubleshooting

---

### Problem

My 802.11b PC Card will not associate with the ASUS Wireless Router.

### Solution

Follow these steps:

1. Try to bring the devices closer together; the PC Card may be out of range of the ASUS Wireless Router.
2. Confirm that the ASUS Wireless Router and PC Card have the same SSID.
3. Confirm that the ASUS Wireless Router and PC Card have the same Encryption settings, if enabled.
4. Confirm that the ASUS Wireless Router's Air and Link LEDs are solid green.
5. Confirm that the authorization table includes or excludes the MAC address of the WLAN PC card if "Wireless Access Control" is enabled.

### Problem

The throughput seems slow.

### Solution

To achieve maximum throughput, verify that your antennas are well-placed, not behind metal, and do not have too many obstacles between them. If you move the client closer to the ASUS Wireless Router and throughput increases, you may want to consider adding a second the ASUS Wireless Router and implementing roaming.

- Check antenna, connectors and cabling.
- Verify network traffic does not exceed 37% of bandwidth.
- Check to see that the wired network does not exceed 10 broadcast messages per second.
- Verify wired network topology and configuration.



## Przykład 8

### ***Po nastawieniu telefonu do ładowania nie widać ikony baterii***

*Możliwa przyczyna:* Bateria jest całkowicie wyczerpana lub bardzo długo nie była używana.

*Rozwiązanie:* Ikona baterii może pojawić się na ekranie dopiero po 30 minutach od chwili rozpoczęcia ładowania.

### ***Niektóre opcje menu są wyszarzone***

*Możliwa przyczyna:* Szary tekst wskazuje funkcje, które są tymczasowo niedostępne. Usługa jest nieaktywna lub abonament jej nie obejmuje.

*Rozwiązanie:* Skontaktuj się z operatorem sieci.  
*Możliwa przyczyna:* Ponieważ nie można wysyłać motywów, obrazków i dźwięków chronionych prawem autorskim, menu **Wyślij** jest czasami niedostępne.



# Przykład 9

---

## The battery pack runs down too quickly.

- Charge it sufficiently (→ *step 1 in “Read This First”*).
- You are using the camera in an extremely cold location (page 99).
- The battery terminal is dirty. Clean the battery terminal with a cotton swab, etc., and charge the battery pack.
- The battery pack is dead (page 99). Replace it with a new one.

---

## Cannot turn on the camera.

- Install the battery pack correctly (→ *step 1 in “Read This First”*).
- The battery pack is discharged. Install charged battery pack (→ *step 1 in “Read This First”*).
- The battery pack is dead (page 99). Replace it with a new one.



# Przykład 10



This troubleshooting guide provides solutions to some common problems that you may encounter while installing and/or using ASUS Pocket Wireless AP. These problems require simple troubleshooting that you can perform by yourself. Contact the ASUS Technical Support if you encounter problems not mentioned in this section.

Problem	Action
<b>The ASUS Pocket Wireless AP does not power up.</b>	<ul style="list-style-type: none"><li>• Use a test meter to measure the voltage output of the power source through the power plug.</li><li>• Check if the power plug is properly connected to the device.</li></ul>
<b>Other devices cannot communicate with the ASUS Pocket Wireless AP through a wired network connection.</b>	<ul style="list-style-type: none"><li>• Verify your network configuration to ensure that there is no IP address duplication. Turn off the device in question, then ping the assigned IP address of the device. Make sure no other device responds to that address.</li><li>• Check if the cables have the proper pin outs and connectors. You may also use another LAN cable.</li><li>• Make sure the hub, switch, or computer connected to the ASUS Pocket Wireless AP supports 10Mbps or 100Mbps speed. Do this by check the ASUS Pocket Wireless AP and the Hub LEDs. When you connect the ASUS Pocket Wireless AP to a 10/100 Mbps hub, both the Hub LED and the ASUS Pocket Wireless AP Ethernet LEDs should light up.</li></ul>



# Algorytmy — podsumowanie



Zadanie algorytmiczne składa się ze:

- ▶ scharakteryzowania dopuszczalnego, być może nieskończonego zbioru potencjalnych zestawów danych wejściowych;
- ▶ specyfikacji pożądaných wyników jako funkcji danych wejściowych.

Zakłada się, że zadany jest albo zestaw dozwolonych akcji (operacji) podstawowych albo konfiguracja sprzętowa, w którą je wbudowano. Rozwiązanie zadania algorytmicznego stanowi algorytm złożony z elementarnych instrukcji zadających akcje z ustalonego zbioru.



# Literatura

-  David Harel, Yishai A Feldman, Zdzisław Płoski.  
*Rzecz O Istocie Informatyki: Algorytmika.*  
Klasyka Informatyki. Wydawnictwa Naukowo-Techniczne, Warszawa, wydanie  
wyd. 4 zm., rozsz, 2008.
-  Patricia Sinclair, Ruth Malinowski.  
*French cooking.*  
Weathervane Books, [New York], 1978.



# Kolofon

Prezentacja złożona w systemie  $\text{\LaTeX}$  2 $\epsilon$  z wykorzystaniem klasy beamer. Użyto fontu **Carlito**. Ilustracja na stronie tytułowej pochodzi z dzieła **Gregora Reischa**, **Margarita Philosophica** (1508), zatytułowana jest *Calculating-Table*.

