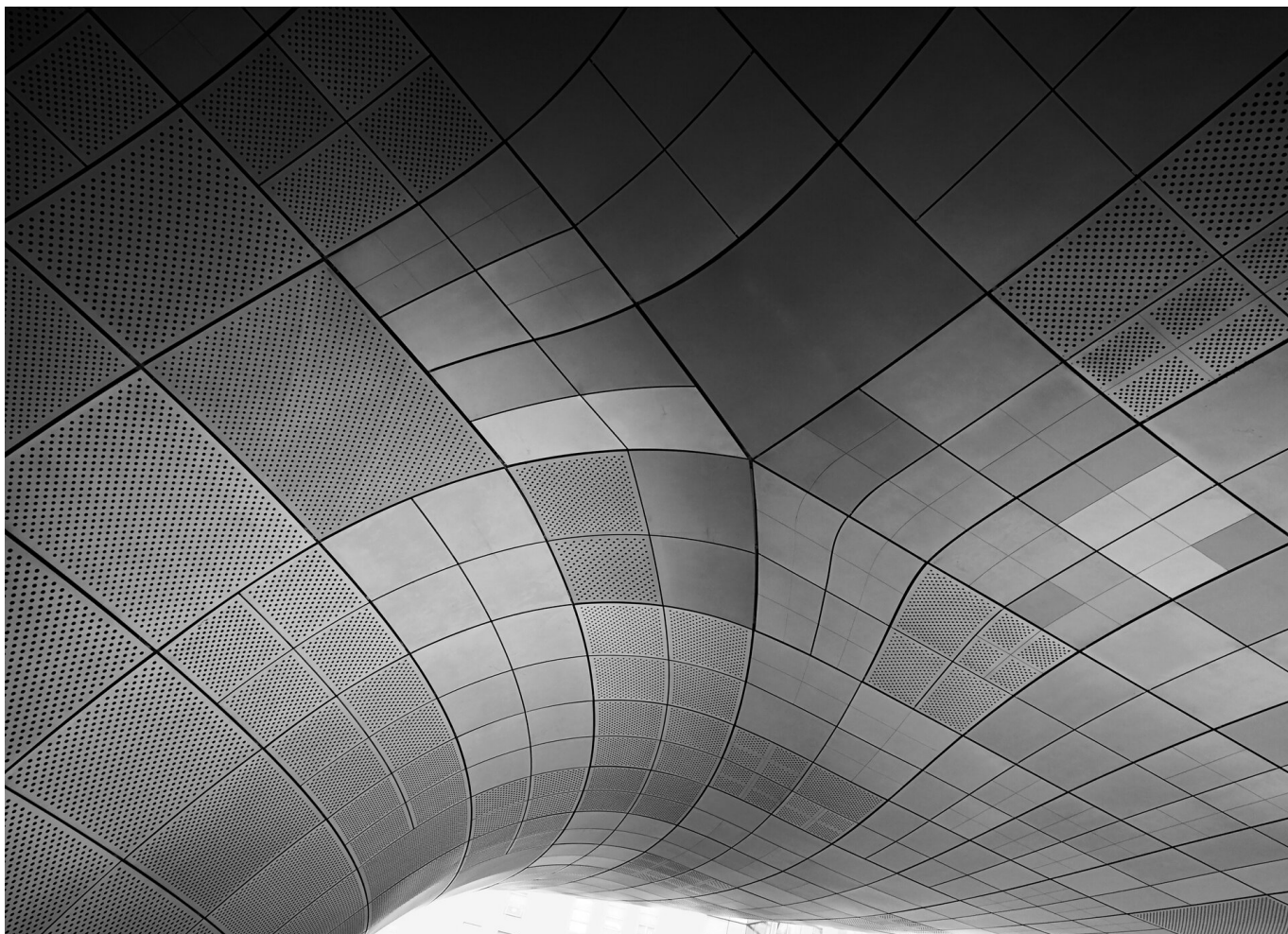




Algorytm Euklidesa – zadania maturalne

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Prezentacja multimedialna](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Algorytm Euklidesa – zadania maturalne

Źródło: Alex Lehner, domena publiczna.

Znamy już [Algorytm Euklidesa](#) w dwóch wariantach. Stosujemy go jako metodę wyznaczania największego wspólnego dzielnika dwóch liczb. Wykorzystujemy go chociażby podczas skracania ułamków.

W tym e-materiale zapoznamy się z przykładowym zadaniem maturalnym dotyczącym tego zagadnienia.

Implementacje algorytmu Euklidesa w poszczególnych językach programowania przedstawiamy w e-materiałach:

- [Algorytm Euklidesa w języku C++](#),
- [Algorytm Euklidesa w języku Java](#),
- [Algorytm Euklidesa w języku Python](#).

Twoje cele

- Przeanalizujesz zastosowanie algorytmu Euklidesa podczas wyliczania wspólnego mianownika dwóch ułamków.
- Prześledzisz konstrukcję zadań maturalnych, wykorzystujących ten algorytm.
- Zastosujesz w praktyce wiadomości dotyczących algorytmu Euklidesa.

Przeczytaj

Algorytm Euklidesa pozwala na szybkie znalezienie Największego Wspólnego Dzielnika (NWD) dwóch liczb naturalnych dodatnich. Znając jego wartość, będziemy również w stanie wyznaczyć Najmniejszą Wspólną Wielokrotność (NWW) tych liczb.

Zależność ta jest szczególnie istotna przy znajdowaniu wspólnego mianownika dla dwóch ułamków. Informacja o NWW ich mianowników pozwoli nam określić, przez jaką liczbę musimy pomnożyć ich liczniki i mianowniki, aby sprowadzić je do wspólnego mianownika. Dzięki temu, w kolejnym kroku, będziemy mogli wykonać na nich operację dodawania lub odejmowania.

Zadanie typu maturalnego

Rozwiążemy teraz jedno z zadań typu maturalnego.

Zadanie to zostało stworzone przez CKE i pochodzi ze zbioru zadań maturalnych z informatyki (zadanie nr **18.1**), dostępnego na oficjalnej stronie internetowej CKE.

```
1 Dane:
2     liczby całkowite dodatnie a i b.
3 Wynik:
4     Największy wspólny dzielnik liczb a i b.
5
6 funkcja NWD(a, b)
7     dopóki b != 0 wykonuj
8         r = a mod b
9         a = b
10        b = r
11
12     zwróć a i zakończ
```

Przeanalizuj działanie funkcji `NWD()` i dla wskazanych argumentów podaj liczbę [operacji modulo](#).

a	b	liczba operacji mod w wierszu 8
25	15	
116	324	
762	282	

Naszym celem jest analiza działania funkcji `NWD()` i podanie liczby wykonanych operacji modulo dla argumentów podanych w tabeli.

Dla pierwszej pary zmiennych `a` i `b`, czyli `NWD(25, 15)`, pierwsze przejście pętli w funkcji wygląda następująco:

```
1 funkcja NWD(25, 15)
2     dopóki b != 0 wykonuj
3         r = 25 mod 15 // czyli 10
4         a = 15
5         b = 10
6
7     zwróć a i zakończ
```

Po jednym wykonaniu pętli `dopóki` liczba wykonanych operacji modulo wynosi 1. Z kodu funkcji możemy wywnioskować, że liczba wykonanych operacji modulo będzie równa liczbie wykonanych pętli dla podanej pary liczb `a` i `b`.

Kontynuujemy analizę działania funkcji dla pierwszego przykładu. Po zakończeniu ostatniego przejścia pętli, nasze zmienne mają wartości:

- `a = 15`,
- `b = 10`,
- `liczba wykonanych operacji modulo = 1`.

Zmienna `b` jest w dalszym ciągu różna od 0. Pętla zostanie wykonana jeszcze raz, realizując następującą operację:

```
1 r = 15 mod 10 // czyli 5
2 a = 10
3 b = 5
```

Po tym wykonaniu pętli, nasze zmienne przyjmują wartości:

- `a = 10`,
- `b = 5`,
- `liczba wykonanych operacji modulo = 2`.

Zmienna `b` jest w dalszym ciągu różna od zera. Wykonamy więc jeszcze jedno przejście pętli `dopóki`, z następującymi operacjami:

```
1
```

```
1 r = 10 mod 5 // czyli 0
2 a = 5
3 b = 0
```

W tym momencie, wartości naszych zmiennych prezentują się tak:

- $a = 5$,
- $b = 0$,
- liczba wykonań operacji modulo = 3.

Widzimy więc, że było to ostatnie wykonanie pętli `dopóki`, ponieważ jej warunek nie jest już spełniany - b osiągnęło wartość 0.

Kończymy pętlę, a tym samym działanie funkcji poprzez zwrócenie wartości argumentu a - czyli 5. Jest to NWD dwóch liczb.

Nie zapominajmy jednak, że zgodnie z treścią zadania, mamy podać, ile razy została wykonana operacja modulo.

Jak już zauważyliśmy, w każdym przejściu pętli `dopóki` wystąpi jedna taka operacja.

W przypadku liczb 25 i 15 pętla ta wykonała się trzy razy. Tak więc w pierwszym wierszu kolumny liczba operacji mod wpisalibyśmy odpowiednio wartość 3.

Znajdziemy teraz odpowiedzi dla pozostałych dwóch par liczb a i b .

Dla $a = 116$, $b = 324$ kolejne reszty z dzielenia r osiągną następujące wartości:

1. 116,
2. 92,
3. 24,
4. 20,
5. 4,
6. 0.

Jest sześć reszt, tak więc operacja modulo wykonała się sześć razy.

Oto wartości r dla ostatniego przykładu z zadania:

1. 198,
2. 84,
3. 30,
4. 24,
5. 6,
6. 0.

Ponownie mamy więc do czynienia z sześcioma wykonaniami operacji modulo. W ten sposób prześledziliśmy działanie algorytmu Euklidesa.

Słownik

operacja modulo

inaczej: operacja obliczania reszty z dzielenia

Prezentacja multimedialna

Polecenie 1

Napisz program, który sprowadzi podane dwa ułamki do postaci nieskracalnej, a następnie wypisze większy z dwóch mianowników. Możesz założyć, że mianowniki na każdym etapie działania algorytmu będą różne.

Jako dane testowe do programu przyjmij następujące wartości:

- `licznik1 = 5`
- `mianownik1 = 25`
- `licznik2 = 3`
- `mianownik2 = 18`

Ważne!

Ułamkiem nieskracalnym nazwiemy taki ułamek, w którym największy wspólny dzielnik licznika i mianownika wynosi 1.

Przykłady ułamków nieskracalnych:

$$\frac{3}{7}, \frac{15}{17}, \frac{6}{19}$$

Specyfikacja problemu:

Dane:

- `licznik1` – liczba całkowita dodatnia, będąca licznikiem pierwszego ułamka
- `mianownik1` – liczba naturalna dodatnia będąca mianownikiem pierwszego ułamka
- `licznik2` – liczba całkowita dodatnia, będąca licznikiem drugiego ułamka

- mianownik2 – liczba naturalna dodatnia będąca mianownikiem drugiego ułamka

Wynik:

- mianownik – liczba całkowita będąca większym z dwóch mianowników uzyskanych po skróceniu ułamków do postaci nieskracalnej.

Wyjście dla podanych danych:

1 6

Polecenie 2

Porównaj swoje rozwiązanie z prezentacją.



Euklides z Aleksandrii – grecki matematyk żyjący na przełomie IV i III w. p.n.e., twórca jednego z najśłynniejszych dzieł matematycznych pt. *Elementy*.

Źródło: es.wikipedia.org, domena publiczna.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

Zapiszemy w pseudokodzie algorytm, który sprowadzi ułamek o mianowniku `mianownik1` i liczniku `licznik1` oraz ułamek o mianowniku `mianownik2` i liczniku `licznik2` do formy nieskracalnej, a następnie wypisze większy z dwóch mianowników.

2

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

Nasz algorytm zaczniemy od zainicjowania zmiennych odpowiadających za mianowniki i liczniki odpowiednich ułamków.

|

3

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

Sprowadzając ułamki do formy nieskracalnej, będziemy musieli znaleźć NWD licznika i mianownika. Obliczymy go za pomocą zoptymalizowanego algorytmu Euklidesa.

W celu wykonania kolejnych kroków algorytmu, będziemy zmuszeni modyfikować i nadpisywać wartości licznika i mianownika. Ponieważ nie chcemy ich stracić, zadeklarujemy teraz zmienne pomocnicze a i b , którym przypiszemy wartości licznika i mianownika danego ułamka.

|

4

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

Zapis algorytmu zaczniemy od użycia pętli `dopóki`, która będzie wykonywana tak długo, jak wartość zmiennej b będzie różna od 0.

|

5

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

W środku pętli zastosujemy poznaną wcześniej procedurę zoptymalizowanego algorytmu Euklidesa.

6

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

Po wykonaniu pętli, w zmiennej a będzie zapisany NWD licznika i mianownika pierwszego ułamka. W celu sprowadzenia ułamka do nieskracalnej formy, podzielimy licznik i mianownik przez ich NWD.

7

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

Teraz powtórzymy procedurę dla drugiego ułamka.

8

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

W tym momencie ułamki posiadają już formę nieskracalną.

Teraz musimy zastosować instrukcję warunkową jeżeli. W jej teście logicznym sprawdzimy, czy zmienna mianownik1 jest większa od mianownik2.

|

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

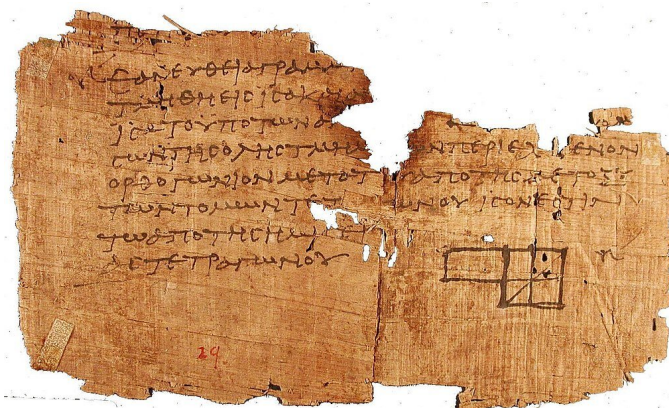
W przypadku, gdy mianownik drugiego ułamka jest większy, to właśnie jego wypiszemy.

Oto pełne rozwiązanie zadania:

|

9

10



Zapisany na papirusie fragment dzieła *Elementy*.

Źródło: en.wikipedia.org, domena publiczna.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

Na zakończenie warto utworzyć funkcję NWD, która będzie miała dwa parametry: a i b. Dzięki temu unikniemy powtarzania się tego samego

kodu. Poniżej przedstawiamy rozwiązanie zadania bazujące na funkcji NWD.

|

Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Polecenie 3

Zapoznaj się z poniższym zadaniem typu maturalnego. Następnie przeanalizuj prezentację, w której przedstawione zostało rozwiązanie zadania.

Podpunkt a)

Wykorzystując funkcję NWD oraz następującą zależność:

$$NWD(a_1, a_2, \dots, a_n) = NWD(NWD(a_1, a_2, \dots, a_{n-1}), a_n)$$

możemy wyznaczyć największy wspólny dzielnik n liczb całkowitych dodatnich $a_1, a_2, \dots, a_n$.

Przykład:

$$\begin{aligned} NWD(15, 24, 60) &= NWD(NWD(15, 24), 60) = \\ &= NWD(3, 60) = 3 \end{aligned}$$

Uzupełnij poniższą tabelkę i dla wskazanych n liczb całkowitych dodatnich $a_1, a_2, \dots, a_n$ oblicz ich największy wspólny dzielnik.

$a_1, a_2, \dots, a_n$	$NWD(a_1, a_2, \dots, a_n)$
36, 24, 72, 150	

$a_1, a_2, \dots, a_n$	$NWD(a_1, a_2, \dots, a_n)$
119, 187, 323, 527, 731	
121, 330, 990, 1331, 110, 225	

```

1 Dane:
2     liczby całkowite dodatnie a i b.
3 Wynik:
4     Największy wspólny dzielnik liczb a i b.
5
6 funkcja NWD(a, b)
7     dopóki b != 0 wykonuj
8         r = a mod b
9         a = b
10        b = r
11
12     zwróć a i zakończ

```

Powyższe zadanie zostało stworzone przez CKE i pochodzi ze zbioru zadań maturalnych z informatyki (zadanie nr **18.2**), dostępnego na oficjalnej stronie internetowej CKE.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

W celu wyznaczenia największego wspólnego dzielnika liczb: {36, 24, 72, 150}, będziemy postępować zgodnie z przedstawionym w poleceniu przykładem.

2

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

$\text{NWD}(36, 24, 72, 150) =$

$\text{NWD}(\text{NWD}(36, 24, 72), 150) =$

$\text{NWD}(\text{NWD}(\text{NWD}(36, 24), 72), 150)$

Najbardziej zagnieżdżona funkcja $\text{NWD}(36, 24)$ zwraca wynik 12. Zastępujemy więc $\text{NWD}(36, 24)$ liczbą 12. Otrzymujemy:

$\text{NWD}(\text{NWD}(12, 72), 150) =$

3

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

Wynikiem wywołania $\text{NWD}(12, 72)$ znów jest 12. Zatem ponownie zastępujemy $\text{NWD}(12, 72)$ liczbą 12 i w efekcie otrzymujemy:

$\text{NWD}(12, 150) =$

4

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

Wykonanie $\text{NWD}(12, 150)$ zwróci wynik 6. Tym samym udało nam się rozwiązać pierwszy przykład polecenia. Kolejne przykłady będą polegały na podobnej zasadzie działania. Najkorzystniej jest rozpisać kolejne wywołania funkcji NWD , jak zostało to przedstawione w pierwszym przykładzie.

5

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

W drugim przykładzie, należy wyznaczyć NWD zbioru: {119, 187, 323, 527, 731}.

$\text{NWD}(119, 187, 323, 527, 731) =$

$\text{NWD}(\text{NWD}(119, 187, 323, 527), 731) =$

$\text{NWD}(\text{NWD}(\text{NWD}(17, 323), 527), 731)$

6

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

Po prawidłowym rozpisaniu zagnieżdżeń funkcji NWD, przechodzimy do obliczenia najbardziej zagnieżdżonej: $\text{NWD}(17, 323)$. Otrzymany wynik (17) podstawiamy do kolejnego wywołania NWD.

$\text{NWD}(\text{NWD}(17, 527), 731) =$

$\text{NWD}(17, 731) = 17$

Tym samym obliczyliśmy największy wspólny dzielnik dla podanego zbioru.

7

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

W ostatnim przykładzie mamy do czynienia z największym zbiorem liczb: {121, 330, 990, 1331, 110, 225}

$\text{NWD}(121, 330, 990, 1331, 110, 225) =$

$\text{NWD}(\text{NWD}(121, 330, 990, 1331, 110), 225) =$

$\text{NWD}(\text{NWD}(\text{NWD}(121, 330, 990, 1331), 110), 225) =$

$\text{NWD}(\text{NWD}(\text{NWD}(\text{NWD}(121, 330, 990), 1331), 110), 225) =$

$\text{NWD}(\text{NWD}(\text{NWD}(\text{NWD}(\text{NWD}(121, 330), 990), 1331), 110), 225)$

8

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

Obliczając wyniki poszczególnych wywołań funkcji, otrzymamy:

$\text{NWD}(\text{NWD}(\text{NWD}(\text{NWD}(11, 990), 1331), 110), 225) =$

$\text{NWD}(\text{NWD}(\text{NWD}(11, 1331), 110), 225) =$

$\text{NWD}(\text{NWD}(11, 110), 225) =$

$\text{NWD}(11, 225) = 1$

Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Podpunkt b)

Napisz algorytm, który korzystając z funkcji $\text{NWD}(a, b)$, wyznaczy największy wspólny dzielnik ciągu liczb $a_1, a_2, \dots, a_n$.

Specyfikacja problemu:

Dane:

- $a_1, a_2, \dots, a_n$ – liczby całkowite dodatnie.

Wynik:

Największy wspólny dzielnik liczb $a_1, a_2, \dots, a_n$.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

W celu rozwiązania tego zadania, będziemy musieli wykorzystać przedstawioną w podpunkcie a) zależność:

$$NWD(a_1, a_2, \dots, a_n) = NWD(NWD(a_1, a_2, \dots, a_{n-1}), a_n)$$

2

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

Na sam początek utworzymy zmienną `wynik`.
Przypiszemy do niej pierwszą podaną liczbę: a_1 :

1. `wynik =  $a_1$`

3

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

Obliczanie wyników kolejnych wywołań funkcji NWD.

Następnie, w kolejnych krokach iteracyjnie obliczamy największy wspólny dzielnik zmiennej `wynik` oraz liczby a_i dla kolejnych wartości $i = 2, 3, \dots, n$. W tym celu tworzymy pętlę `for`.

1. `wynik =  $a_1$`

2. dla $i = 2, 3, \dots, n$ wykonuj

4

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

Wewnątrz pętli będziemy aktualizować zmienną `wynik`. Będziemy do niej przypisywać wynik wywołania funkcji NWD z argumentami: `wynik` oraz a_i .

1. `wynik =  $a_1$`

2. dla $i = 2, 3, \dots, n$ wykonuj

3. $\text{wynik} = \text{NWD}(\text{wynik}, a_i)$

5

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PS3Y0kQvR>

Na zakończenie wystarczy zwrócić zmienną `wynik`. Oto rozwiązanie zadania:

1. $\text{wynik} = a_1$
2. dla $i = 2, 3, \dots, n$ wykonuj
3. $\text{wynik} = \text{NWD}(\text{wynik}, a_i)$
4. zwróć `wynik` i zakończ

Sprawdź się

Pokaż ćwiczenia:   

Przyjmijmy, że mamy do dyspozycji dwie liczby naturalne dodatnie a i b , które są mianownikami w ułamkach: $\frac{12}{a}$, $\frac{5}{b}$

Ćwiczenie 1



Napisz algorytm znajdujący NWD dla liczb a i b za pomocą pseudokodu. Twoje rozwiązanie powinno wykorzystywać wersję algorytmu z odejmowaniem.

1

Ćwiczenie 2



Uzupełnij zdanie, wpisując właściwą liczbę.

Podaj liczbę wykonań operacji odejmowania w pseudokodzie napisanym w **Ćwiczeniu 1** dla wartości $a = 35$ i $b = 25$

Ćwiczenie 3



Napisz algorytm w postaci pseudokodu wyznaczający NWW liczb a i b (a, b – liczby naturalne dodatnie)

1

Ćwiczenie 4



Uzupełnij zdanie, wpisując właściwą liczbę.

Dla wartości $a = 27$ i $b = 108$, otrzymamy NWW równą .

Ćwiczenie 5



Zapisz algorytm sprawdzający, czy dwie liczby naturalne dodatnie są względnie pierwsze. Wykorzystaj pseudokod.

Parę liczb nazwiemy względnie pierwszymi, gdy nie mają innego wspólnego dzielnika niż 1. Rozwiązanie powinno wykorzystywać algorytm Euklidesa w wersji z modulo.

1

Ćwiczenie 6

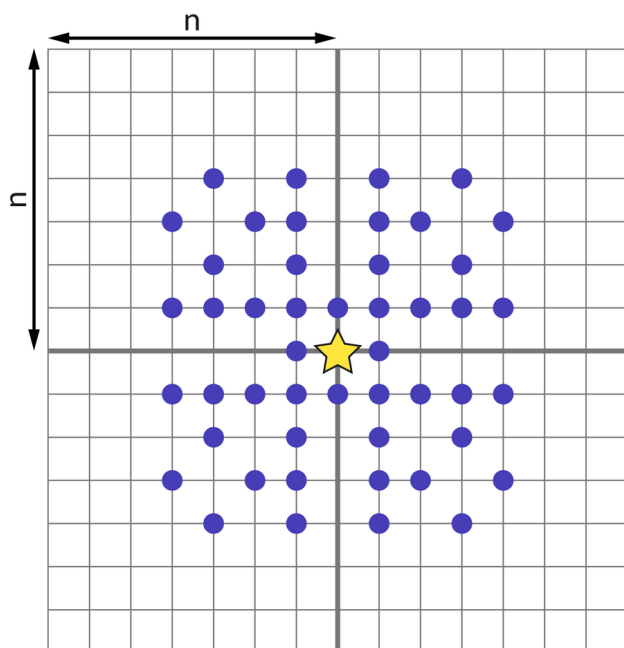


Zapisz rekurencyjną wersję algorytmu Euklidesa za pomocą pseudokodu.

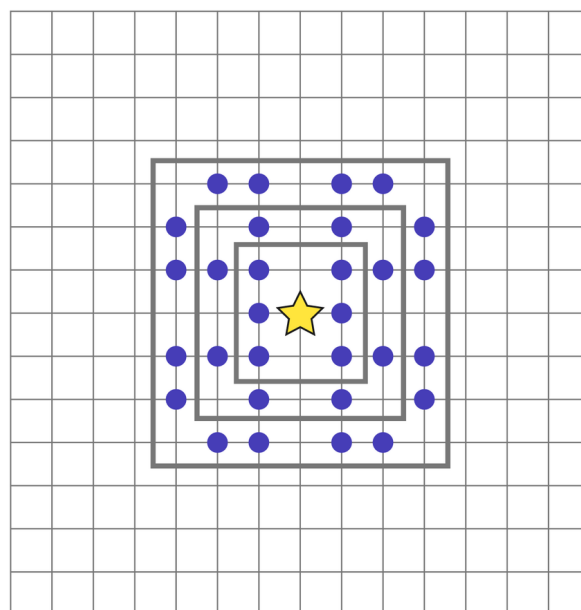
1

Material źródłowy od ćwiczeń nr 8-9

Ćwiczenia 8-9 pochodzą z informatora maturalnego autorstwa CKE. Dokument znaleźć można na oficjalnej stronie internetowej CKE.



Rys. 1



Rys. 2

Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Fani spotykają się z ulubionymi gwiazdami filmowymi w sali, której podłoga ma kształt kwadratu. Podłogę podzielono pionowymi i poziomymi liniami na mniejsze kwadraty rozmiaru 1×1 . Zaproszona gwiazda zawsze siada w środku sali. Fani zajmują miejsca w sali na przecięciach linii. Jedno miejsce może zająć co najwyżej jedna osoba.

Po wejściu na salę wypełniają oni kolejno strefy zaznaczone na Rys. 2. kwadratami, każdą strefę maksymalnie, jak się da. Fani zajmują tylko te miejsca, z których gwiazda jest widoczna, co oznacza, że na odcinku łączącym ich miejsce z miejscem zajmowanym przez gwiazdę nie ma żadnej innej osoby. Na potrzeby zadania fanów i gwiazdę utożsamiamy z punktami, które zajmują. Na Rys.1. przedstawione jest przykładowe rozmieszczenie wszystkich osób zgodnie z opisem. Na Rys. 2. pokazany jest przykładowe całkowite wypełnienie sali w 3 strefach. Jak widać w I strefie zasiądzie 8 fanów, w II strefie – 8, a w III aż 16.

Ćwiczenie 7



Uzupełnij tabelę.

Strefa	Maksymalna liczba fanów w kwadracie	Przyrost fanów w stosunku do poprzedniego poziomu
I	8	-
II	16	8
III	32	16
IV		
V		
VI		



Potraktuj salę (Rys. 1.) jako układ kartezjański, przyjmując pozycję gwiazdy jako punkt $(0, 0)$, a pozycje fanów jako punkty kratowe o określonych współrzędnych (x, y) . Zapisz, wykorzystując pseudokod, algorytm sprawdzający, czy fan może stanąć w miejscu o danych współrzędnych (x, y) .

Specyfikacja problemu:

Dane:

- x, y – liczby całkowite określające położenie fana względem gwiazdy

Wynik:

- Komunikat TAK, jeżeli fan widzi gwiazdę lub komunikat NIE – jeżeli fan gwiazdy nie widzi.



Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Algorytm Euklidesa – zadania maturalne

Grupa docelowa:

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres podstawowy i rozszerzony

Podstawa programowa:

Cele kształcenia – wymagania ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Treści nauczania – wymagania szczegółowe

I. Rozumienie, analizowanie i rozwiązywanie problemów.

Zakres podstawowy. Uczeń:

2) stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy:

a) na liczbach: badania pierwszości liczby, zamiany reprezentacji liczb między pozycyjnymi systemami liczbowymi, działań na ułamkach z wykorzystaniem NWD i NWW,

I + II. Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

1) zapisuje za pomocą listy kroków, schematu blokowego lub pseudokodu, i implementuje w wybranym języku programowania, algorytmy poznane na wcześniejszych etapach oraz algorytmy:

a) algorytm Euklidesa w wersji iteracyjnej i rekurencyjnej wraz z zastosowaniami,

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Przeanalizujesz zastosowanie algorytmu Euklidesa podczas wyliczania wspólnego mianownika dwóch ułamków.
- Prześledzisz konstrukcję zadań maturalnych, wykorzystujących ten algorytm.
- Zastosujesz w praktyce wiadomości dotyczących algorytmu Euklidesa.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda;
- oprogramowanie dla języka C++, w tym kompilator GCC/G++ 4.5 (lub nowszej wersji) i Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio;
- oprogramowanie dla języka Java SE 8 (lub nowszej wersji), w tym Eclipse 4.4 (lub nowszej wersji);
- oprogramowanie dla języka Python 3 (lub nowszej wersji), w tym PyCharm lub IDLE.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Algorytm Euklidesa – zadania maturalne”. Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Przeczytaj”.

Faza wstępna:

1. Nauczyciel wprowadza uczniów szczegółowo w temat lekcji i jej cele. Może posłużyć się wyświetloną na tablicy zawartością sekcji „Wprowadzenie”.
2. **Rozpoznanie wiedzy uczniów.** Nauczyciel zadaje uczniom pytania dotyczące ich aktualnego stanu wiedzy w obszarze poruszanego tematu i programowania, np.
 - co to jest algorytm Euklidesa i jakie ma zastosowanie?
 - do czego służy operacja modulo?Chętni uczniowie udzielają na nie odpowiedzi.

Faza realizacyjna:

1. Odnosząc się do treści zawartej w sekcji „Przeczytaj”, uczniowie w parach konstruują alternatywny przykład, definiując samodzielnie problem, rozwiązanie i ewentualną implementację. Rezultaty omawiane są na forum klasy.
2. **Praca z multimediami.** Uczniowie w zespołach dwuosobowych zapoznają się z treścią polecenia nr 1 z sekcji „Prezentacja multimedialna” i wspólnie analizują kolejne kroki rozwiązania postawionego problemu.
3. **Ćwiczenie umiejętności.** Uczniowie wykonują pierwsze ćwiczenia interaktywne z sekcji „Sprawdź się”. Wyniki pracy omawiane są na forum i komentowane przez nauczyciela.
4. Praca indywidualna – implementacja poznanej techniki do rozwiązywania problemów informatycznych – wykonywanie ćwiczeń z sekcji „Sprawdź się”.

Faza podsumowująca:

1. Wybrana osoba podsumowuje zajęcia, zwracając uwagę na nabyte umiejętności, omawia ewentualne problemy podczas rozwiązywania ćwiczeń.

Praca domowa:

1. Uczniowie opracowują FAQ (minimum 3 pytania i odpowiedzi) do tematu lekcji („Algorytm Euklidesa – zadania maturalne”).
2. Uczniowie implementują algorytmy omawiane w sekcji „Prezentacja multimedialna” w wybranym przez siebie języku programowania.

Materiały pomocnicze:

- Oficjalna dokumentacja techniczna dla języka C++.
- Oficjalna dokumentacja techniczna dla kompilatora GCC/G++ 4.5 (lub nowszej wersji).

- Oficjalna dokumentacja techniczna dla oprogramowania Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio.
- Oficjalna dokumentacja techniczna dla języka Java SE 8 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Eclipse 4.4 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla języka Python 3 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania PyCharm lub IDLE.

Wskazówki metodyczne:

- Multimedia w sekcji „Prezentacja multimedialna” można potraktować jako zadanie domowe dotyczące analizy problemu zawartego w temacie „Algorytm Euklidesa – zadania maturalne”.