Pierwsza Ławka · zestaw działowy · liceum i technikum

Dział 13

Algorytmy na liczbach: Euklides rekurencyjnie, sito Eratostenesa, czynniki pierwsze, szybkie potęgowanie, schemat Hornera



Algorytm Euklidesa w wersji rekurencyjnej i jego zastosowania, sito Eratostenesa, rozkład liczby na czynniki pierwsze, szybkie potęgowanie iteracyjne i rekurencyjne (także modulo), schemat Hornera – zapis algorytmów, analiza i programy. Osiem kart: teoria z przykładami, ćwiczenia, utrwalenie w formule CKE, zadanie przy komputerze z plikami danych (ZIP), kartkówka i sprawdzian w grupach A i B – z kluczem i zasadami oceniania.

Informatyka | klasa 2 LO i technikum | Dział 13 | zakres rozszerzony (ZR)

	Karta	Rodzaj	Poziom	Zadania
1	**Algorytmy na liczbach: Euklides rekurencyjnie, sito Eratostenesa, czynniki pierwsze, szybkie potęgowanie, schemat Hornera – najważniejsze wiadomości** Teoria, przykłady krok po kroku, rozgrzewka	Najważniejsze wiadomości	ZR	3 zad. · 6 pkt
2	**Algorytmy na liczbach: Euklides rekurencyjnie, sito Eratostenesa, czynniki pierwsze, szybkie potęgowanie, schemat Hornera – ćwiczenia podstawowe** Ćwiczę umiejętności z działu	Ćwiczenia	ZR	7 zad. · 17 pkt
3	**Algorytmy na liczbach: Euklides rekurencyjnie, sito Eratostenesa, czynniki pierwsze, szybkie potęgowanie, schemat Hornera – utrwalenie** Zadania w formule CKE: polecenia i punktacja jak w arkuszach maturalnych	Zadania typu maturalnego	ZR	6 zad. · 15 pkt
4	**Algorytmy na liczbach: Euklides rekurencyjnie, sito Eratostenesa, czynniki pierwsze, szybkie potęgowanie, schemat Hornera – w praktyce, przy komputerze** Pliki danych w archiwum ZIP · wyniki w plikach wynikiN.txt	Ćwiczenia	ZR	1 zad. · 9 pkt
5	**Kartkówka: Euklides, sito i potęgowanie – grupa A** 15 minut · 8 pkt · bez komputera	Kartkówka	ZR	4 zad. · 8 pkt 15 min

6	**Kartkówka: Euklides, sito i potęgowanie – grupa B** 15 minut · 8 pkt · bez komputera	Kartkówka	ZR	4 zad. · 8 pkt 15 min
7	**Sprawdzian: Algorytmy na liczbach: Euklides rekurencyjnie, sito Eratostenesa, czynniki pierwsze, szybkie potęgowanie, schemat Hornera – grupa A** 45 minut · 20 pkt · bez komputera	Sprawdzian	ZR	8 zad. · 20 pkt 45 min
8	**Sprawdzian: Algorytmy na liczbach: Euklides rekurencyjnie, sito Eratostenesa, czynniki pierwsze, szybkie potęgowanie, schemat Hornera – grupa B** 45 minut · 20 pkt · bez komputera	Sprawdzian	ZR	8 zad. · 20 pkt 45 min

Jak korzystać z zestawu

- Karta 1 przypomina najważniejsze wiadomości i pokazuje przykłady rozwiązane krok po kroku (z kodem w Pythonie albo opisem kroków w aplikacji) – od niej warto zacząć.
- Karta 2 to ćwiczenia podstawowe, karta 3 – utrwalenie w formule zadań maturalnych CKE (ramka „Jak to sprawdza CKE").
- Karta 4 to praca przy komputerze: pliki danych są w archiwum ZIP dołączonym do zestawu, a odpowiedzi kontrolne dla plików *_przyklad (pliki o nazwie z dopiskiem „_przyklad") podano przy zadaniach. Wyniki zapisuje się w plikach wynikiN.txt – jak na maturze.
- Karty 5–6 to kartkówka (15 minut), karty 7–8 – sprawdzian (45 minut), obie w grupach A i B: te same typy zadań i ta sama punktacja, inne dane. Kartkówka i sprawdzian nie wymagają komputera.
- Klucz zawiera odpowiedzi, przykładowe programy, zasady oceniania w stylu CKE i numery wymagań podstawy. Osobny plik „Powtórka" zbiera teorię, typowe błędy i test „Sprawdź się".

Oznaczenia

ZP	zakres podstawowy
ZR	zakres rozszerzony (treści tylko dla ZR)
(0–2 pkt)	liczba punktów możliwych do zdobycia za zadanie
Zadanie 4.1.	wiązka zadań do wspólnego materiału źródłowego
Jak to sprawdza CKE	typ zadania i typowy błąd ze sprawozdań CKE
Na maturze	wymagania z podstawy programowej, których dotyczy temat

Wymagania z podstawy programowej (Dz.U. 2024 poz. 1019)

- **I+II.R1a** zapisuje za pomocą listy kroków 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,
- **I+II.R1c** zapisuje za pomocą listy kroków lub pseudokodu i implementuje w wybranym języku programowania algorytmy poznane na wcześniejszych etapach oraz algorytmy: c) generowania liczb pierwszych metodą sita Eratostenesa,
- **I+II.R1h** zapisuje za pomocą listy kroków lub pseudokodu i implementuje w wybranym języku programowania algorytmy poznane na wcześniejszych etapach oraz algorytmy: h) obliczania wartości wielomianu za pomocą schematu Hornera,
- **I+II.R1i** zapisuje za pomocą listy kroków lub pseudokodu i implementuje w wybranym języku programowania algorytmy poznane na wcześniejszych etapach oraz algorytmy: i) szybkiego potęgowania liczb w wersji iteracyjnej i rekurencyjnej,
- **I+II.R2a** wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów: a) rozkładania liczby na czynniki pierwsze,

Materiał oryginalny – żadne zadanie nie jest kopią zadania CKE ani podręcznika; dane w plikach są fikcyjne. Podstawa programowa LO i technikum w brzmieniu z 2024 r. (Dz.U. 2024 poz. 1019). Wymagania: I+II.R1a, I+II.R1c, I+II.R1h, I+II.R1i, I+II.R2a. Miejsce w programach nauczania: „Informatyka na czasie" (Nowa Era) ZR – algorytmy na liczbach (nazwy serii podano tylko informacyjnie). Do użytku własnego i pracy z własnymi uczniami.

Algorytmy na liczbach: Euklides rekurencyjnie, sito Eratostenesa, czynniki pierwsze, szybkie potęgowanie, schemat Hornera – najważniejsze wiadomości

Teoria, przykłady krok po kroku, rozgrzewka

ZR </> KARTA 1/8

Imię i nazwisko: ..... Klasa: ..... Data: ..... Wynik: ..... / 6 pkt

Najważniejsze wiadomości · 6 pkt

Przypomnij sobie

- **Algorytm Euklidesa – wersja rekurencyjna**: NWD(a, 0) = a, a dla b > 0: NWD(a, b) = NWD(b, a mod b). Funkcja wywołuje samą siebie z mniejszą drugą liczbą, aż dojdzie do b = 0 (warunek końca rekurencji). Liczba wywołań = liczba obrotów pętli w wersji iteracyjnej + 1.
- **Zastosowania NWD**: NWW(a, b) = a div NWD(a, b) · b; skracanie ułamków; liczby względnie pierwsze (NWD = 1); NWD kilku liczb: NWD(a, b, c) = NWD(NWD(a, b), c); np. podział prostokąta a × b na jednakowe kwadraty o największym boku.
- **Sito Eratostenesa** wyznacza wszystkie liczby pierwsze ≤ n: zapisujemy liczby 2…n, bierzemy najmniejszą niewykreśloną p i wykreślamy jej wielokrotności, zaczynając od p · p (mniejsze wykreśliły już mniejsze liczby pierwsze). Kończymy, gdy p · p > n – wszystkie niewykreślone liczby są pierwsze. Sito opłaca się, gdy potrzebujemy wielu liczb pierwszych naraz.
- **Rozkład na czynniki pierwsze**: dzielimy n przez d = 2, 3, 4, … tak długo, jak się dzieli, zapisując d. Wystarczy d · d ≤ n; jeśli na końcu n > 1, to pozostałe n jest czynnikiem pierwszym. Dzielniki złożone (4, 6, …) nigdy nie dzielą bieżącego n, bo ich czynniki zostały już usunięte.
- **Szybkie potęgowanie**: $x^n = (x^{n/2})^2$ dla n parzystego i $x \cdot x^{n-1}$ dla nieparzystego. Wersja iteracyjna przegląda bity wykładnika: w = 1; dopóki n > 0: jeśli n nieparzyste, w = w · x; x = x · x; n = n div 2. Mnożeń jest około $2 \cdot \log_2 n$ zamiast n − 1 (dla n = 1 000 000 – kilkadziesiąt zamiast miliona).
- **Potęgowanie modulo**: (a · b) mod m = ((a mod m) · (b mod m)) mod m, więc po każdym mnożeniu bierzemy resztę – liczby pozostają małe. W Pythonie robi to pow(x, n, m) (szybciej niż pow(x, n) % m – dokumentacja Pythona).
- **Schemat Hornera**: $a_0x^n + a_1x^{n-1} + \ldots + a_n = (\ldots((a_0x + a_1)x + a_2)x + \ldots)x + a_n$; w = a_0, potem w = w · x + a_i. Dla wielomianu stopnia n: n mnożeń i n dodawań. Zamiana zapisu z systemu p na dziesiętny to Horner dla x = p.
- **Efektywność (I.R5)**: test pierwszości do √n – około √n dzieleń dla jednej liczby; sito do n – około n log log n operacji dla wszystkich liczb naraz; Euklides – liczba kroków rośnie jak logarytm z mniejszej liczby; szybkie potęgowanie – log n.

Przykłady z rozwiązaniem

Przykład 1. Oblicz NWD(1155, 420) funkcją rekurencyjną. Ile razy funkcja zostanie wywołana?

```
def nwd(a, b):
    if b == 0:
        return a
    return nwd(b, a % b)
```

nwd(1155, 420) → nwd(420, 315), bo 1155 mod 420 = 315 → nwd(315, 105) → nwd(105, 0) → zwraca 105. Wywołań: **4**. NWW(1155, 420) = 1155 div 105 · 420 = 4620.

Przykład 2. **Wyznacz sitem Eratostenesa liczby pierwsze ≤ 50.**

```
def sito(n):
    pierwsza = [True] * (n + 1)
    pierwsza[0] = pierwsza[1] = False
    p = 2
    while p * p <= n:
        if pierwsza[p]:
            for k in range(p * p, n + 1, p):
                pierwsza[k] = False
        p = p + 1
    return [i for i in range(n + 1) if pierwsza[i]]
```

p = 2: wykreślamy 4, 6, …, 50; p = 3: od 9 co 3; p = 5: od 25 co 5; p = 7: 49. p = 8 → 64 > 50 – koniec. Zostają: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47 (15 liczb).

Przykład 3. **Rozłóż 5544 na czynniki pierwsze.**

```
def czynniki(n):
    wynik = []
    d = 2
    while d * d <= n:
        while n % d == 0:
            wynik.append(d)
            n = n // d
        d = d + 1
    if n > 1:
        wynik.append(n)
    return wynik
```

5544 : 2 = 2772; : 2 = 1386; : 2 = 693; 693 : 3 = 231; : 3 = 77; 77 : 7 = 11; d = 8: 8 · 8 = 64 > 11 – pętla się kończy, 11 > 1 dopisujemy. $5544 = 2 \cdot 2 \cdot 2 \cdot 3 \cdot 3 \cdot 7 \cdot 11 = 2^3 \cdot 3^2 \cdot 7 \cdot 11$.

Przykład 4. **Oblicz 3^{13} szybkim potęgowaniem iteracyjnym i rekurencyjnym; policz mnożenia. Oblicz też W(3) dla $W(x) = 2x^3 - 3x^2 + 5$ schematem Hornera.**

```
def potega(x, n):
    w = 1
    while n > 0:
        if n % 2 == 1:
            w = w * x
        x = x * x
        n = n // 2
    return w
```

$13 = 1101_2$. n = 13: w = 3, x = 9; n = 6: x = 81; n = 3: w = 243, x = 6561; n = 1: w = 1594323, $x = 6561^2$.
Mnożeń: 7 (zamiast 12). Wersja rekurencyjna:

```
def potega_r(x, n):
    if n == 0:
        return 1
    if n % 2 == 0:
        p = potega_r(x, n // 2)
        return p * p
    return x * potega_r(x, n - 1)
```

wywołania dla n = 13, 12, 6, 3, 2, 1, 0 – 7 wywołań, 6 mnożeń.

```
def horner(a, x):        # a[0] – współczynnik przy najwyższej potędze
    w = a[0]
    for i in range(1, len(a)):
        w = w * x + a[i]
    return w
```

Horner dla [2, −3, 0, 5], x = 3: $w = 2 \to 2 \cdot 3 - 3 = 3 \to 3 \cdot 3 + 0 = 9 \to 9 \cdot 3 + 5 =$ **32**.

Zadania

Zadanie 1. (0–3 pkt) ..... / 3

Uzupełnij.

1. NWD(84, 0) = _____
2. nwd(48, 18) wywoła kolejno nwd(18, _____)
3. Rozkład 360 = _____

Zadanie 2. (0–1 pkt) ..... / 1

W sicie Eratostenesa dla n = 100 ostatnią liczbą p, której wielokrotności wykreślamy, jest

A. 10 B. 7 C. 11 D. 97

Zadanie 3. (0–2 pkt) ..... / 2

Oceń prawdziwość zdań.

1. Szybkie potęgowanie oblicza x^{16} mniejszą liczbą mnożeń niż mnożenie x przez siebie 15 razy.	P	F
2. Schemat Hornera dla wielomianu stopnia 4 wykonuje 4 mnożenia.	P	F

Algorytmy na liczbach: Euklides rekurencyjnie, sito Eratostenesa, czynniki pierwsze, szybkie potęgowanie, schemat Hornera – ćwiczenia podstawowe

ZR </> KARTA 2/8

Ćwiczę umiejętności z działu

Imię i nazwisko: ..... Klasa: ..... Data: ..... Wynik: ..... / 17 pkt

Ćwiczenia · 17 pkt

Zadanie 1. (0–3 pkt) ..... / 3

Przeanalizuj funkcję nwd(a, b) (karta 1). Uzupełnij tabelę.

```
def nwd(a, b):
    if b == 0:
        return a
    return nwd(b, a % b)
```

a	b	liczba wywołań nwd	wynik
252	105		
89	55		
360	84		

Zadanie 2. (0–3 pkt) ..... / 3

Wykonujesz sito Eratostenesa dla n = 40. Wypisz wszystkie wielokrotności, które funkcja sito oznacza w kroku p = 3 i p = 5 (od p · p, także te wykreślone wcześniej).

1. p = 3: _____
2. p = 5: _____
3. Na którym p sito się kończy (pierwsze p z p · p > 40)? _____

Zadanie 3. (0–2 pkt) ..... / 2

Uczeń zmienił w funkcji sito warunek pętli na **while p * p < n**. Podaj dwie liczby n, dla których funkcja zwróci błędny wynik, i wyjaśnij dlaczego.

```
def sito(n):
    pierwsza = [True] * (n + 1)
    pierwsza[0] = pierwsza[1] = False
    p = 2
    while p * p < n:
        if pierwsza[p]:
            for k in range(p * p, n + 1, p):
                pierwsza[k] = False
        p = p + 1
    return [i for i in range(n + 1) if pierwsza[i]]
```

Miejsce na odpowiedź

Zadanie 4. (0–3 pkt) ..... / 3

Rozłóż liczby na czynniki pierwsze (zapisz iloczyn czynników w kolejności niemalejącej).

1. 1386 = _____
2. 1001 = _____
3. 9409 = _____

Zadanie 5. (0–3 pkt) ..... / 3

Dla x = 2 policz mnożenia w funkcji potega (iteracyjnej) i wywołania funkcji potega_r (rekurencyjnej) z karty 1.

n	mnożenia – potega	wywołania – potega_r
10		
16		
31		

Zadanie 6. (0–2 pkt) ..... / 2

Oblicz W(2) dla $W(x) = x^4 + 4x^3 + 3x + 2$ schematem Hornera. Uzupełnij kolejne wartości w.

współczynnik	1	4	0	3	2
w	1				

Zadanie 7. (0–1 pkt) ..... / 1

Zamień liczbę z systemu piątkowego na dziesiętny schematem Hornera.

1. 31402_5 = _____

Algorytmy na liczbach: Euklides rekurencyjnie, sito Eratostenesa, czynniki pierwsze, szybkie potęgowanie, schemat Hornera – utrwalenie

Zadania w formule CKE: polecenia i punktacja jak w arkuszach maturalnych

ZR </> KARTA 3/8

Imię i nazwisko: ..... Klasa: ..... Data: ..... Wynik: ..... / 15 pkt

Zadania typu maturalnego · 15 pkt

Jak to sprawdza CKE

Algorytmy na liczbach pojawiają się co roku: analiza algorytmu w tabeli (2024 zad. 2 „Cyfry", 0–2), liczba wywołań funkcji rekurencyjnej (2025 zad. 1, 9 pkt; 2026 zad. 1 „Rekurencja", 7 pkt), zapis algorytmu ze Specyfikacją i Uwagą (2023 zad. 2.1, 0–3; 2026 zad. 2.2, 0–4), NWD przy komputerze (2025 zad. 3.1, 0–2). Sprawozdania 2023–2026: najtrudniejsze są zadania z zapisem algorytmu – zdający pomijają warunki z Uwagi pod treścią.

Zadanie 1. (0–1 pkt) ..... / 1

Oceń prawdziwość podanych zdań. Zaznacz P, jeśli zdanie jest prawdziwe, albo F – jeśli jest fałszywe.

```
def nwd(a, b):
    if b == 0:
        return a
    return nwd(b, a % b)
```

1. Funkcja nwd(a, b) z karty 1 dla a = 34 i b = 21 zostanie wywołana 8 razy.	P	F
2. W sicie Eratostenesa liczba 91 zostanie wykreślona w kroku p = 7.	P	F
3. Rozkład liczby n na czynniki pierwsze może zawierać co najwyżej jeden czynnik większy od $\sqrt{n}$.	P	F

Zadanie 2. (0–1 pkt) ..... / 1

Dokończ zdanie. Zaznacz właściwą odpowiedź spośród podanych.
Funkcja potega(x, n) (wersja iteracyjna z karty 1) dla n = 1000 wykona

```
def potega(x, n):
    w = 1
    while n > 0:
        if n % 2 == 1:
            w = w * x
        x = x * x
        n = n // 2
    return w
```

A. 16 mnożeń B. 10 mnożeń C. 500 mnożeń D. 999 mnożeń

Zadanie 5. (0–3 pkt) ..... / 3

Zapisz w pseudokodzie lub w wybranym języku programowania funkcję nwd3(a, b, c), która obliczy NWD trzech liczb całkowitych dodatnich, korzystając z rekurencyjnej wersji algorytmu Euklidesa. Podaj wynik dla 90, 150, 210.

Miejsce na odpowiedź

Zadanie 6. (0–4 pkt) ..... / 4

Zapisz w pseudokodzie lub w wybranym języku programowania algorytm, który metodą sita Eratostenesa obliczy, ile jest liczb pierwszych nie większych od n.

Przykład: dla n = 100 wynikiem jest 25.

Uwaga: Możesz używać tablicy wartości logicznych i operacji arytmetycznych; nie wolno używać funkcji bibliotecznych badających pierwszość.

Specyfikacja
Dane: n – liczba całkowita, $2 \le n \le 1\,000\,000$
Wynik: ile – liczba liczb pierwszych ≤ n

Miejsce na odpowiedź

PIERWSZA ŁAWKA

Zadanie 7. (0–4 pkt) ..... / 4

Zapisz w pseudokodzie lub w wybranym języku programowania algorytm, który dla danej liczby n obliczy największy czynnik pierwszy liczby n (np. dla 360 = 2·2·2·3·3·5 wynik 5).

Przykład: dla n = 360 wynikiem jest 5.

Uwaga: Twój algorytm może używać tylko zmiennych przechowujących pojedyncze liczby całkowite, operacji arytmetycznych (dodawanie, odejmowanie, mnożenie, dzielenie całkowite i reszta z dzielenia), porównań, operatorów logicznych, instrukcji warunkowych, pętli, przypisań i własnych funkcji. Nie wolno używać funkcji wbudowanych, tablic ani list ani zamieniać liczby na napis.

Specyfikacja
Dane: n – liczba całkowita, $n \geq 2$
Wynik: s – największy czynnik pierwszy liczby n

Miejsce na odpowiedź

Zadanie 8. (0–3 pkt) ..... / 3

Wyjaśnij, dlaczego w rozkładzie na czynniki wystarczy sprawdzać dzielniki d z warunkiem $d \cdot d \leq n$ i dlaczego dzielnik złożony (np. 6) nigdy nie podzieli bieżącej liczby n.

Miejsce na odpowiedź

PIERWSZA ŁAWKA

CO JESZCZE JEST W ŚRODKU

Algorytmy na liczbach: Euklides rekurencyjnie, sito Eratostenesa, czynniki pierwsze, szybkie potęgowanie, schemat Hornera – kartkówka,

Podgląd pokazał 10 z 30 stron. Pełna wersja: 30 stron A4, PDF do druku i DOCX do edycji.

- Karta 1: Algorytmy na liczbach: Euklides rekurencyjnie, sito Eratostenesa, czynniki pierwsze, szybkie potęgowanie, schemat Hornera – najważniejsze wiadomości – 3 zadania, 6 pkt
- Karta 2: Algorytmy na liczbach: Euklides rekurencyjnie, sito Eratostenesa, czynniki pierwsze, szybkie potęgowanie, schemat Hornera – ćwiczenia podstawowe – 7 zadań, 17 pkt
- Karta 3: Algorytmy na liczbach: Euklides rekurencyjnie, sito Eratostenesa, czynniki pierwsze, szybkie potęgowanie, schemat Hornera – utrwalenie – 6 zadań, 15 pkt
- Karta 4: Algorytmy na liczbach: Euklides rekurencyjnie, sito Eratostenesa, czynniki pierwsze, szybkie potęgowanie, schemat Hornera – w praktyce, przy komputerze – 1 zadanie, 9 pkt
- Karta 5: Kartkówka: Euklides, sito i potęgowanie – grupa A – 4 zadania, 8 pkt
- Karta 6: Kartkówka: Euklides, sito i potęgowanie – grupa B – 4 zadania, 8 pkt
- Karta 7: Sprawdzian: Algorytmy na liczbach: Euklides rekurencyjnie, sito Eratostenesa, czynniki pierwsze, szybkie potęgowanie, schemat Hornera – grupa A – 8 zadań, 20 pkt
- Karta 8: Sprawdzian: Algorytmy na liczbach: Euklides rekurencyjnie, sito Eratostenesa, czynniki pierwsze, szybkie potęgowanie, schemat Hornera – grupa B – 8 zadań, 20 pkt Karta 1 p

Kup pełną wersję – plik od razu na e-mail

pierwszalawka.pl/material/informatyka-liceum-klasa-2-zr-dzial-13-algorytmy-na-liczbach-euklides-rekurencyjnie-sito