Pierwsza Ławka · zestaw działowy · liceum i technikum

Dział 13

Rekurencja i dziel i zwyciężaj, metoda wstępująca i zstępująca, fraktale

Funkcje rekurencyjne, stos i drzewo wywołań, metoda wstępująca i zstępująca (zapamiętywanie), dziel i zwyciężaj, rekurencyjne fraktale (zbiór Cantora, fraktal Vicsek, drzewo) – na papierze i w Pythonie. 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 3 LO i technikum | Dział 13 | zakres rozszerzony (ZR)

	Karta	Rodzaj	Poziom	Zadania
1	**Rekurencja i dziel i zwyciężaj, metoda wstępująca i zstępująca, fraktale – najważniejsze wiadomości** Teoria, przykłady krok po kroku, rozgrzewka	Najważniejsze wiadomości	ZR	4 zad. · 8 pkt
2	**Rekurencja i dziel i zwyciężaj, metoda wstępująca i zstępująca, fraktale – ćwiczenia podstawowe** Ćwiczę umiejętności z działu	Ćwiczenia	ZR	6 zad. · 18 pkt
3	**Rekurencja i dziel i zwyciężaj, metoda wstępująca i zstępująca, fraktale – utrwalenie** Zadania w formule CKE: polecenia i punktacja jak w arkuszach maturalnych	Zadania typu maturalnego	ZR	6 zad. · 11 pkt
4	**Rekurencja i dziel i zwyciężaj, metoda wstępująca i zstępująca, fraktale – w praktyce, przy komputerze** Pliki danych w archiwum ZIP · wyniki w plikach wynikiN.txt	Ćwiczenia	ZR	2 zad. · 12 pkt
5	**Kartkówka: rekurencja i fraktale – grupa A** 15 minut · 8 pkt · bez komputera	Kartkówka	ZR	4 zad. · 8 pkt 15 min
6	**Kartkówka: rekurencja i fraktale – grupa B** 15 minut · 8 pkt · bez komputera	Kartkówka	ZR	4 zad. · 8 pkt 15 min
7	**Sprawdzian: Rekurencja i dziel i zwyciężaj, metoda wstępująca i zstępująca, fraktale – grupa A** 45 minut · 20 pkt · bez komputera	Sprawdzian	ZR	8 zad. · 20 pkt 45 min
8	**Sprawdzian: Rekurencja i dziel i zwyciężaj, metoda wstępująca i zstępująca, fraktale – 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.R1** wyróżnia w problemie podproblemy i charakteryzuje: metodę połowienia, stosuje podejście zachłanne i rekurencję;
- **I.R3** w zależności od problemu rozwiązuje go, stosując metodę wstępującą lub zstępującą;
- **I+II.R3b** objaśnia oraz porównuje podstawowe metody i techniki algorytmiczne oraz struktury danych, wykorzystując przykłady problemów i algorytmów, w szczególności: b) rekurencję (do generowania ciągów liczb, potęgowania, sortowania liczb, generowania fraktali),
- **I+II.R3c** objaśnia oraz porównuje podstawowe metody i techniki algorytmiczne oraz struktury danych, wykorzystując przykłady problemów i algorytmów, w szczególności: c) metodę dziel i zwyciężaj (jednoczesne znajdowanie minimum i maksimum, sortowanie przez scalanie i szybkie),
- **I+II.R1j** zapisuje za pomocą listy kroków lub pseudokodu i implementuje w wybranym języku programowania algorytmy poznane na wcześniejszych etapach oraz algorytmy: j) rekurencyjnego tworzenia fraktali;

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.R1, I.R3, I+II.R3b, I+II.R3c, I+II.R1j. Miejsce w programach nauczania: „Informatyka na czasie" (Nowa Era) ZR – algorytmika, klasa 3 (nazwy serii podano tylko informacyjnie). Do użytku własnego i pracy z własnymi uczniami.

PIERWSZA ŁAWKA

Rekurencja i dziel i zwyciężaj, metoda wstępująca i zstępująca, fraktale – najważniejsze wiadomości

Teoria, przykłady krok po kroku, rozgrzewka

ZR </> KARTA 1/8

Imię i nazwisko: Klasa: Data: Wynik: ..... / 8 pkt

Najważniejsze wiadomości · 8 pkt

Przypomnij sobie

- **Rekurencja** to definiowanie (albo rozwiązywanie) problemu za pomocą tego samego problemu dla mniejszych danych. Funkcja rekurencyjna wywołuje samą siebie. Musi mieć **przypadek bazowy** (warunek zakończenia), w którym zwraca wynik bez wywołania, a każde wywołanie musi przybliżać dane do przypadku bazowego.
- **Stos wywołań**: każde wywołanie czeka na wynik wywołań, które samo uruchomiło. Liczba czekających naraz wywołań to **głębokość rekurencji**. W Pythonie jest ona ograniczona (sys.getrecursionlimit(), zwykle 1000) – po przekroczeniu program kończy się błędem RecursionError. Brak przypadku bazowego = rekurencja bez końca.
- **Metoda zstępująca** (z góry na dół): zaczynamy od problemu dla n i rozkładamy go na podproblemy – tak działa rekurencja. **Metoda wstępująca** (z dołu do góry): zaczynamy od najmniejszych przypadków i w pętli budujemy wyniki dla coraz większych danych, zapisując je (np. w liście) – bez wywołań rekurencyjnych.
- **Drzewo wywołań** pokazuje, jakie wywołania uruchamia funkcja. Gdy funkcja wywołuje się dwa razy (np. s(n − 1) i s(n − 3)), te same wartości liczone są wielokrotnie, a liczba wywołań rośnie wykładniczo. Pomaga **zapamiętywanie** (spamiętywanie, ang. *memoization*) wyników albo metoda wstępująca – wtedy każdą wartość liczymy raz.
- **Dziel i zwyciężaj**: (1) dzielimy problem na mniejsze części tego samego rodzaju, (2) rozwiązujemy je rekurencyjnie, (3) łączymy wyniki. Przykłady: maksimum w ciągu przez połowy, sortowanie przez scalanie i szybkie, szybkie potęgowanie, wyszukiwanie binarne (tam jedna z połówek jest odrzucana).
- **Liczba operacji w dziel i zwyciężaj**: maksimum ciągu n liczb przez połowy wymaga n − 1 porównań (każde porównanie „łączy” dwie części), a liczba poziomów podziału dla $n = 2^k$ wynosi $k = \log_2 n$ (na stosie czeka wtedy naraz k + 1 wywołań). Rekurencję opisuje zależność, np. $T(1) = 0$, $T(n) = 2 \cdot T(n/2) + 1$.
- **Fraktal** to figura samopodobna: jej części są pomniejszonymi kopiami całości. Fraktal stopnia n rysujemy rekurencyjnie z kilku kopii stopnia n − 1; stopień 0 to figura podstawowa (odcinek, kwadrat). Zbiór Cantora: z odcinka usuwamy środkową tercję i powtarzamy to dla pozostałych części – stopień n ma 2^n odcinków długości 3^{-n}, razem $(2/3)^n$.
- **Fraktal Vicsek** („krzyżyk”): kwadrat dzielimy na 3 × 3 kwadraty i zostawiamy środkowy oraz 4 przyległe do jego boków; stopień n ma 5^n kwadratów o boku $a/3^n$. Ogólnie: jeśli figura składa się z k kopii, to stopień n ma k^n elementów podstawowych.
- **Rekurencja czy pętla?** Każdy algorytm rekurencyjny można zapisać iteracyjnie. Rekurencja daje krótki, czytelny zapis (fraktale, drzewa, dziel i zwyciężaj), ale zużywa pamięć na stos i bywa wolna przy powtarzanych wywołaniach. Na maturze zadania typu „Zapisz nierekurencyjną funkcję…” sprawdzają właśnie metodę wstępującą.

Przykłady z rozwiązaniem

Przykład 1. **Ciąg: a(1) = 2, a(n) = 3 · a(n − 1) − 1. Oblicz a(5) zstępująco (rekurencyjnie) i wstępująco (pętlą).**

```
def a(n):               # zstępująco: od a(n) w dół do a(1)
    if n == 1:
        return 2
    return 3 * a(n - 1) - 1

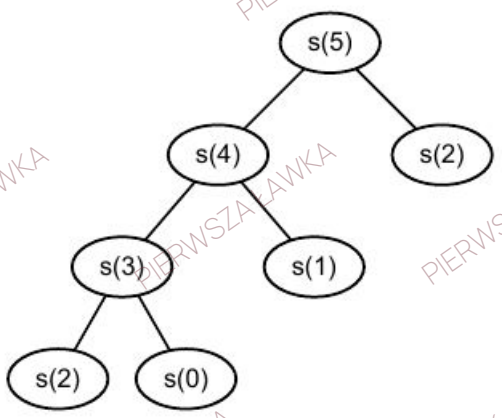
def a_wst(n):           # wstępująco: od a(1) w górę
    x = 2
    for i in range(2, n + 1):
        x = 3 * x - 1
    return x
```

Zstępująco: a(5) = 3 · a(4) − 1, a(4) = 3 · a(3) − 1, …, aż do a(1) = 2 – na stosie czeka 5 wywołań, potem wyniki wracają: a(2) = 5, a(3) = 14, a(4) = 41, a(5) = 122.
Wstępująco: x = 2 → 5 → 14 → 41 → 122 (4 obroty pętli, bez stosu).

Przykład 2. **Na ile sposobów można wejść na 5 stopni, robiąc kroki o 1 lub o 3 stopnie? Narysuj drzewo wywołań funkcji s(5).**

```
def s(n):
    if n < 3:
        return 1
    return s(n - 1) + s(n - 3)
```

Ostatni krok to 1 stopień (zostaje s(n − 1) sposobów) albo 3 stopnie (s(n − 3)). Dla n < 3 jest jeden sposób (same kroki o 1). Drzewo wywołań s(5):



s(3) = s(2) + s(0) = 2, s(4) = s(3) + s(1) = 3, s(5) = s(4) + s(2) = 4. Wywołań jest 7, a s(2) liczone jest 2 razy. Dla n = 30 naiwna funkcja wywoła się 116 849 razy, a metoda wstępująca obliczy 28 sum.

Przykład 3. **Znajdź maksimum ciągu 4, 9, 2, 7, 5, 1, 8, 3 metodą dziel i zwyciężaj. Ile porównań wykona funkcja?**

```
def maks(t, l, p):          # największy element t[l..p]
    if l == p:
        return t[l]
    s = (l + p) // 2
    m1 = maks(t, l, s)          # lewa połowa
    m2 = maks(t, s + 1, p)      # prawa połowa
    if m1 > m2:
        return m1
    return m2
```

Podział: [4, 9, 2, 7] i [5, 1, 8, 3] → [4, 9], [2, 7], [5, 1], [8, 3] → pojedyncze liczby.
Łączenie: max(4, 9) = 9, max(2, 7) = 7, max(5, 1) = 5, max(8, 3) = 8 (4 porównania); 9 i 7 → 9, 5 i 8 → 8 (2 porównania); 9 i 8 → 9 (1 porównanie). Razem 7 = 8 − 1 porównań, głębokość podziału 3 = $\log_2 8$.

Przykład 4. **Wypisz odcinki zbioru Cantora stopnia 2 na odcinku [0, 1]. Ile odcinków i jaka łączna długość będzie w stopniu 4?**

```
def cantor(a, b, n):        # odcinki zbioru Cantora stopnia n na [a, b]
    if n == 0:
        print(a, b)
        return
    d = (b - a) / 3
    cantor(a, a + d, n - 1)     # lewa tercja
    cantor(b - d, b, n - 1)     # prawa tercja
```



Stopień 1: [0, 1/3], [2/3, 1]. Stopień 2: [0, 1/9], [2/9, 1/3], [2/3, 7/9], [8/9, 1] (program wypisze przybliżenia dziesiętne).

Stopień 4: $2^4 = 16$ odcinków, każdy długości $1/3^4 = 1/81$, razem 16/81. Procedura wywoła się 31 razy (1 + 2 + 4 + 8 + 16).

Zadania

Zadanie 1. (0–3 pkt) ..... / 3

Ciąg jest określony wzorem: b(1) = 1, b(n) = 2 · b(n − 1) + n dla n ≥ 2. Uzupełnij.

1. b(2) = _____ 2. b(3) = _____ 3. b(4) = _____

Zadanie 2. (0–1 pkt) ..... / 1

Funkcja f(n) zwraca 1 dla n = 0, a dla n > 0 zwraca 2 · f(n − 1). Wynikiem f(5) jest

A. 32 B. 16 C. 64 D. 10

PIERWSZA ŁAWKA

Rekurencja i dziel i zwyciężaj, metoda wstępująca i zstępująca, fraktale – ćwiczenia podstawowe

ZR </> KARTA 2/8

Ćwiczę umiejętności z działu

Imię i nazwisko: Klasa: Data: Wynik: ..... / 18 pkt

Ćwiczenia · 18 pkt

Zadanie 1. (0–3 pkt) ..... / 3

Oblicz metodą wstępującą wartości funkcji s(n) (liczba sposobów wejścia na n stopni krokami 1 lub 3; s(0) = s(1) = s(2) = 1, s(n) = s(n − 1) + s(n − 3)). Uzupełnij tabelę.

n	0	1	2	3	4	5	6	7
s(n)	1	1	1					

Zadanie 2. (0–3 pkt) ..... / 3

Narysuj drzewo wywołań naiwnej funkcji s(6) z karty 1. Podaj, ile jest wszystkich wywołań (z pierwszym) i ile razy liczone jest s(2).

Miejsce na odpowiedź

Zadanie 3. (0–3 pkt) ..... / 3

Uzupełnij luki w funkcji maks(t, l, p), która metodą dziel i zwyciężaj zwraca największy element listy t w przedziale indeksów od l do p.

```
def maks(t, l, p):          # największy element t[l..p]
    if  luka 1 :
        return t[l]
    s =  luka 2
    m1 = maks(t, l, s)          # lewa połowa
    m2 = maks(t,  luka 3 , p)      # prawa połowa
    if m1 > m2:
        return m1
    return m2
```

1. luka 1: _____
2. luka 2: _____
3. luka 3: _____

Zadanie 4. (0–2 pkt) ..... / 2

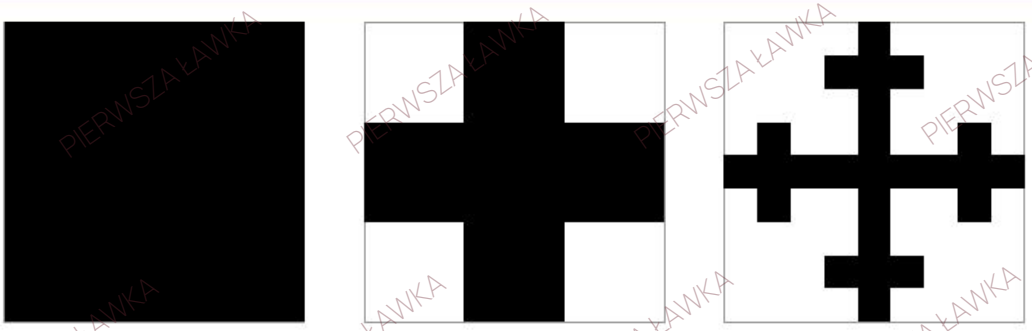
Uczeń napisał funkcję podaną niżej. Co się stanie po wywołaniu suma(3)? Popraw funkcję.

```
def suma(n):          # miała zwracać 1 + 2 + ... + n
    return n + suma(n - 1)
```

Miejsce na odpowiedź

Zadanie 5. (0–4 pkt) ..... / 4

Fraktal Vicsek (rysunek niżej) o boku a. Uzupełnij.



stopień 0 stopień 1 stopień 2

1. stopień 2 – liczba kwadratów: _____
2. stopień 2 – bok kwadratu: _____
3. stopień 3 – liczba kwadratów: _____
4. stopień 3 – bok kwadratu: _____

Zadanie 6. (0–3 pkt) ..... / 3

Zapisz wersję funkcji s(n) z zapamiętywaniem wyników (metoda zstępująca) albo metodą wstępującą (pętla). Wyjaśnij, dlaczego jest szybsza od naiwnej.

Miejsce na odpowiedź

Zadanie 3. (0–1 pkt) ..... / 1

Dokończ zdanie. Zaznacz właściwą odpowiedź spośród podanych.
Funkcja maks(t, l, p) z karty 1 (dziel i zwyciężaj) dla listy 16 liczb wykona porównań elementów

A. 8 B. 4 C. 16 D. 15

Zadanie 4. (0–1 pkt) ..... / 1

Dokończ zdanie. Zaznacz właściwą odpowiedź spośród podanych.
Łączna długość odcinków zbioru Cantora stopnia 4 zbudowanego na odcinku [0, 1] jest równa

A. 8/27 B. 1/81 C. 16/81 D. 4/81

Zadanie 5. (0–3 pkt) ..... / 3

Ciąg d jest określony rekurencyjnie: $d(0) = 1$, $d(1) = 1$, $d(n) = d(n - 1) + 2 \cdot d(n - 2)$ dla $n \geq 2$. Zapisz w pseudokodzie lub w wybranym języku programowania **nierekurencyjny** algorytm (metodą wstępującą), który dla danej liczby n obliczy d(n).

Przykład: dla n = 5 wynikiem jest 21.

Uwaga: Twój algorytm może używać tylko zmiennych przechowujących pojedyncze liczby całkowite, operacji arytmetycznych, porównań, instrukcji warunkowych, pętli i przypisań. Nie wolno używać rekurencji ani list.

Specyfikacja
Dane: n – liczba całkowita, $n \geq 0$
Wynik: d(n) – wartość wyrazu ciągu

Miejsce na odpowiedź

Zadanie 6. (0–2 pkt) ..... / 2

Podaj, ile wywołań (łącznie z pierwszym) wykona naiwna funkcja s(n) z karty 1 dla n = 8, oraz zapisz wzór rekurencyjny na liczbę wywołań W(n).

Miejsce na odpowiedź

Rekurencja i dziel i zwyciężaj, metoda wstępująca i zstępująca, fraktale – w praktyce, przy komputerze

ZR </> KARTA 4/8

Pliki danych w archiwum ZIP · wyniki w plikach wynikiN.txt

Imię i nazwisko: ..... Klasa: ..... Data: ..... Wynik: ..... / 12 pkt

Ćwiczenia · 12 pkt

Zadanie 1. (0–6 pkt)

Kody kontrolne

Opis danych

Kody kontrolne paczek w magazynie liczy funkcja F z karty 3: F(n) = n dla n < 10, a dla n ≥ 10 F(n) = F(n div 10 + n mod 10). Plik **kody.txt** zawiera 400 wierszy – w każdym jedną liczbę całkowitą n ($1 \le n \le 10^9$). Pierwsze wiersze pliku:

```
139512
571879
414782
248261
```

Plik **kody_przyklad.txt** zawiera 15 liczb w tym samym formacie. Zaprogramuj F **rekurencyjnie**.

Do dyspozycji masz plik **kody.txt** oraz mniejszy plik **kody_przyklad.txt**. Odpowiedzi dla danych przykładowych podano przy zadaniach – najpierw sprawdź na nich swój program.

Napisz program (lub kilka programów), który(-e) znajdzie(-ą) i da(dzą) odpowiedzi do podanych zadań. Odpowiedzi zapisz w pliku **wyniki1.txt**, a każdą z nich poprzedź numerem odpowiedniego zadania.

Do oceny oddajesz: plik tekstowy **wyniki1.txt** z odpowiedziami do zadań 1.1–1.3 (każda poprzedzona numerem zadania) oraz pliki z kodem źródłowym programów, np. **zadanie1.py**.

Dane fikcyjne, przygotowane do zadania.

Zadanie 1.1. (0–2 pkt) ..... / 2

Podaj, dla ilu liczb z pliku F(n) = 7. *Dla danych z pliku przykładowego odpowiedź: 2.*

Odpowiedź zapisz w pliku wyniki1.txt

Zadanie 1.2. (0–2 pkt) ..... / 2

Podaj pierwszą w pliku liczbę, dla której obliczenie F(n) wymaga największej liczby wywołań funkcji F (łącznie z pierwszym), oraz tę liczbę wywołań. Obie wartości oddziel spacją. *Dla danych z pliku przykładowego odpowiedź: 877757511 10.*

Odpowiedź zapisz w pliku wyniki1.txt

Zadanie 1.3. (0–2 pkt) ..... / 2

Podaj, dla ilu liczb n z pliku zachodzi F(n + 1) = F(n) + 1. *Dla danych z pliku przykładowego odpowiedź: 14.*

Odpowiedź zapisz w pliku wyniki1.txt

Zadanie 2. (0–6 pkt)

Schody

Opis danych

Funkcja s(n) z karty 1 podaje liczbę sposobów wejścia na n stopni krokami o 1 lub o 3 stopnie (s(0) = s(1) = s(2) = 1, s(n) = s(n − 1) + s(n − 3)). Plik **schody.txt** zawiera 200 wierszy – w każdym liczbę stopni n (1 ≤ n ≤ 100) jednej z klatek schodowych. Pierwsze wiersze pliku:

```
65
31
70
62
```

Plik **schody_przyklad.txt** zawiera 12 wierszy. Uwaga: naiwna rekurencja dla n = 100 nie skończy się w rozsądnym czasie – zastosuj metodę wstępującą albo zapamiętywanie wyników. s(100) mieści się w typie 64-bitowym (long long w C++, long w Javie).

Do dyspozycji masz plik **schody.txt** oraz mniejszy plik **schody_przyklad.txt**. Odpowiedzi dla danych przykładowych podano przy zadaniach – najpierw sprawdź na nich swój program.

Napisz program (lub kilka programów), który(-e) znajdzie(-ą) i da(dzą) odpowiedzi do podanych zadań. Odpowiedzi zapisz w pliku **wyniki2.txt**, a każdą z nich poprzedź numerem odpowiedniego zadania.

Do oceny oddajesz: plik tekstowy **wyniki2.txt** z odpowiedziami do zadań 2.1–2.3 (każda poprzedzona numerem zadania) oraz pliki z kodem źródłowym programów, np. **zadanie2.py**.

Dane fikcyjne, przygotowane do zadania.

Zadanie 2.1. (0–2 pkt) ..... / 2

Podaj, dla ilu wierszy pliku liczba s(n) jest parzysta. *Dla danych z pliku przykładowego odpowiedź: 6.*

Odpowiedź zapisz w pliku wyniki2.txt

Zadanie 2.2. (0–2 pkt) ..... / 2

Podaj największe n w pliku i wartość s(n) dla tego n. Obie liczby oddziel spacją. *Dla danych z pliku przykładowego odpowiedź: 96 5285136390291172.*

Odpowiedź zapisz w pliku wyniki2.txt

Zadanie 2.3. (0–2 pkt) ..... / 2

Podaj, dla ilu wierszy pliku $s(n) > 10^{12}$. *Dla danych z pliku przykładowego odpowiedź: 5.*

Odpowiedź zapisz w pliku wyniki2.txt

CO JESZCZE JEST W ŚRODKU

Rekurencja i dziel i zwyciężaj, metoda wstępująca i zstępująca, fraktale – kartkówka, sprawdzian i karty pracy z powtórką, informatyka, klasa 3 LO i...

Podgląd pokazał 10 z 31 stron. Pełna wersja: 31 stron A4, PDF do druku i DOCX do edycji.

- Karta 1: Rekurencja i dziel i zwyciężaj, metoda wstępująca i zstępująca, fraktale – najważniejsze wiadomości – 4 zadania, 8 pkt
- Karta 2: Rekurencja i dziel i zwyciężaj, metoda wstępująca i zstępująca, fraktale – ćwiczenia podstawowe – 6 zadań, 18 pkt
- Karta 3: Rekurencja i dziel i zwyciężaj, metoda wstępująca i zstępująca, fraktale – utrwalenie – 6 zadań, 11 pkt
- Karta 4: Rekurencja i dziel i zwyciężaj, metoda wstępująca i zstępująca, fraktale – w praktyce, przy komputerze – 2 zadania, 12 pkt
- Karta 5: Kartkówka: rekurencja i fraktale – grupa A – 4 zadania, 8 pkt
- Karta 6: Kartkówka: rekurencja i fraktale – grupa B – 4 zadania, 8 pkt
- Karta 7: Sprawdzian: Rekurencja i dziel i zwyciężaj, metoda wstępująca i zstępująca, fraktale – grupa A – 8 zadań, 20 pkt
- Karta 8: Sprawdzian: Rekurencja i dziel i zwyciężaj, metoda wstępująca i zstępująca, fraktale – grupa B – 8 zadań, 20 pkt Karta 1 przypomina teorię i pokazuje przykłady rozwiązane

Kup pełną wersję – plik od razu na e-mail

pierwszalawka.pl/material/informatyka-liceum-klasa-3-zr-dzial-13-rekurencja-i-dziel-i-zwyciezaj-metoda-wstepujaca