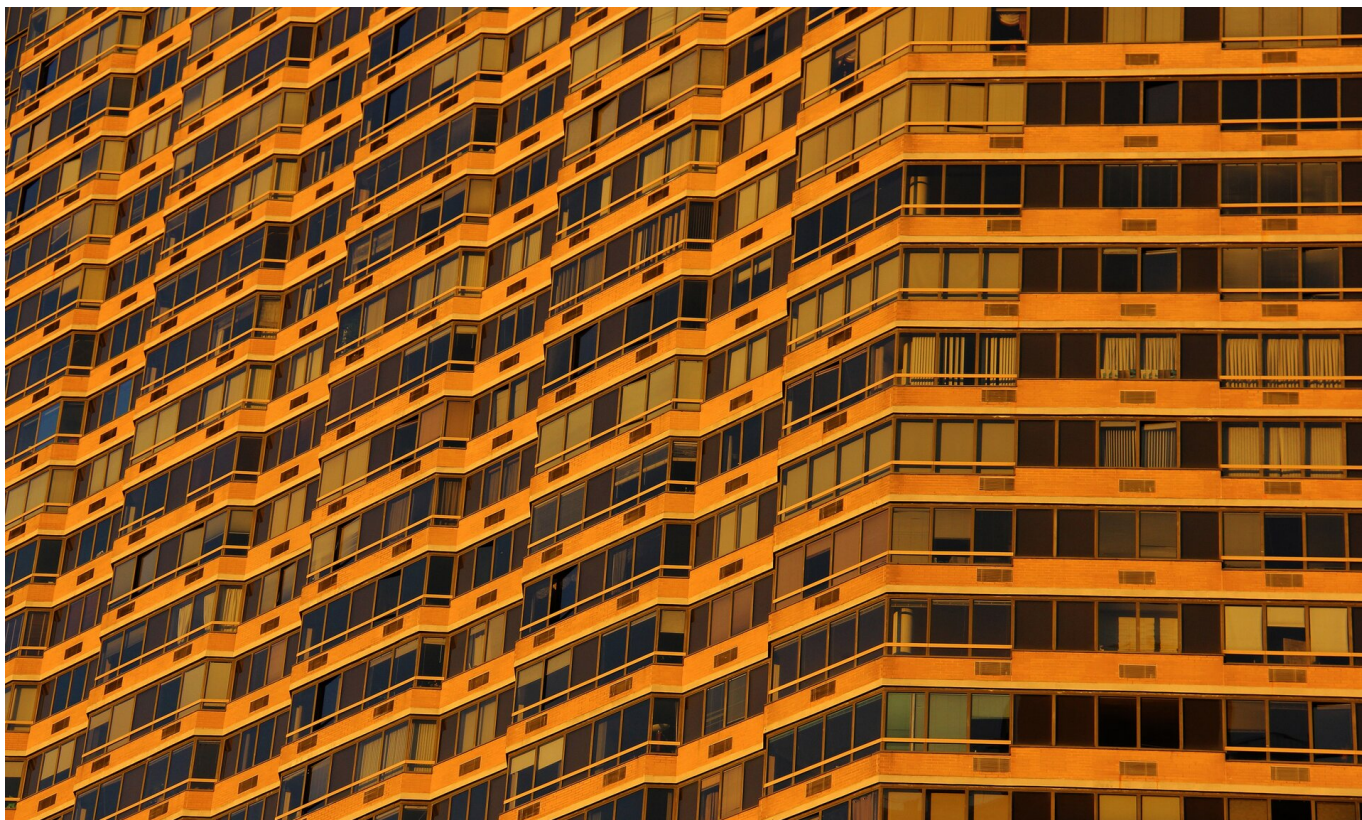




Algorytmy iteracyjne i liczbowe – potęgowanie liczb

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Gra edukacyjna](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Algorytmy iteracyjne i liczbowe – potęgowanie liczb

Źródło: Alan Weiner, domena publiczna.

Problem potęgowania liczb jesteśmy w stanie rozwiązać, stosując algorytm iteracyjny lub rekurencyjny. Potęgę a^n możemy obliczyć zgodnie z definicją, którą najczęściej używamy na lekcji matematyki, a także za pomocą algorytmu tzw. szybkiego potęgowania liczb.

Obliczanie potęgi a^n zgodnie z definicją wymaga od nas wykonania $n - 1$ mnożeń. Dzięki wykorzystaniu schematu Hornera w algorytmie szybkiego potęgowania, liczba wykonywanych mnożeń zależy od liczby bitów w rozwinięciu binarnym wykładnika. Wykonamy zatem maksymalnie $2 \log_2 n$ mnożeń.

Więcej informacji o algorytmach iteracyjnych znajdziesz w e-materiałach:

- [Algorytmy iteracyjne](#),
- [Algorytmy liczbowe](#),
- [Algorytmy iteracyjne – obliczanie silni](#).

Więcej zadań? Sięgnij do e-materiału [Algorytmy iteracyjne i liczbowe – zadania maturalne](#).

Twoje cele

- Przeanalizujesz algorytm iteracyjnego potęgowania liczb.
- Zaimplementujesz program potęgujący liczby wykorzystujący algorytm iteracyjny.
- Scharakteryzujesz i zaimplementujesz algorytm szybkiego potęgowania liczb.

Przeczytaj

Potęgowanie liczb według definicji

Założmy, że chcemy podnieść liczbę a do potęgi n . Aby wykonać to działanie, należy obliczyć iloczyn n czynników o wartości podstawy, czyli a .

$$a^n = \underbrace{a \cdot a \cdot \dots \cdot a}_n$$

Na przykład:

$$2^4 = 2 \cdot 2 \cdot 2 \cdot 2.$$

Iloczyn tych liczb można obliczyć za pomocą algorytmu [iteracyjnego](#) oraz [rekurencyjnego](#).

Specyfikacja problemu:

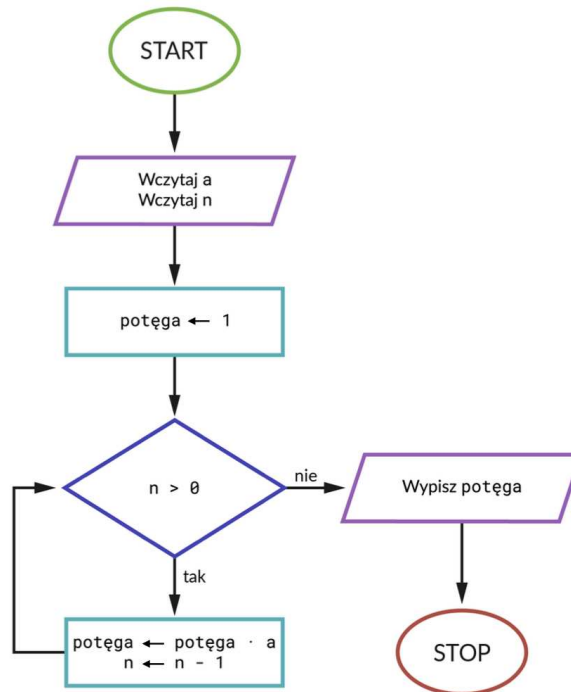
Dane:

- n – liczba naturalna; wykładnik potęgi
- a – liczba całkowita różna od zera; podstawa potęgi

Wynik:

Algorytm wypisuje wynik potęgowania a^n .

Na początek przeanalizujemy schemat blokowy algorytmu w wersji iteracyjnej.



Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY 3.0.

Wykonanych zostanie n iteracji pętli, w których będą przeprowadzane operacje mnożenia aktualnej potęgi przez podstawę a . Zatem złożoność czasowa algorytmu potęgowania liczb według definicji wyrażona jako liczba wykonanych mnożeń wynosi $O(n)$.

Algorytm ten możemy przedstawić także za pomocą pseudokodu. W tym celu wykorzystamy instrukcję iteracyjną dla $\dots$ wykonuj. Wewnątrz pętli umieścimy operację mnożenia aktualnej potęgi przez wartość podstawy potęgi.

```

1 potęga ← 1
2 dla i = n, n - 1, ... 1 wykonuj
3     potęga ← potęga * a
4 wypisz potęga
  
```

Szybkie potęgowanie liczb

Istnieje szybszy sposób potęgowania liczb od potęgowania według definicji.

Wersja iteracyjna [szybkiego potęgowania](#) oparta jest na dwójkowej reprezentacji wykładnika. Weźmy działanie 3^{14} . Liczba 14 w [systemie dwójkowym](#) to:

$$14 = 1110_{(2)}.$$

Możemy zapisać wartość tego wykładnika, korzystając ze schematu Hornera:

$$3^{14} = 3^{((1 \cdot 2 + 1) \cdot 2 + 1) \cdot 2} = \left((3^2 \cdot 3)^2 \cdot 3 \right)^2.$$

Metoda polega na pobieraniu kolejnych cyfr wykładnika zapisanego w systemie dwójkowym, zaczynając od lewej strony. Jeżeli ta cyfra ma wartość:

- 0 – to aktualny wynik podnoszony jest do potęgi drugiej,
- 1 – to aktualny wynik podnoszony jest do potęgi drugiej, a następnie mnożony jest przez podstawę.

Specyfikacja problemu:

Dane:

- n – liczba naturalna w postaci binarnej $n_x n_{x-1} n_{x-2} \dots n_1$; wykładnik potęgi
- x – liczba naturalna; liczba bitów w reprezentacji binarnej liczby n
- a – liczba całkowita; podstawa potęgi

Wynik:

Algorytm zwraca wynik potęgowania a^n .

```
1 potęga ← 1
2 dla i = x, x - 1, ..., 1 wykonuj
3     bit ← cyfra liczby n na pozycji i
4     jeżeli bit = 0
5         potęga ← potęga * potęga
6     w przeciwnym razie
7         potęga ← potęga * potęga * a
8 wypisz potęga
```

Powyższa metoda oparta jest na dwójkowej reprezentacji wykładnika. Liczby w systemie dwójkowym tworzone są od strony prawej (od najmniej znaczącego bitu), zatem jeżeli chcemy dotrzeć do poszczególnych bitów wykładnika zaczynając od lewej strony, należy najpierw użyć algorytmu konwersji liczby z systemu dziesiętnego na system binarny.

Analizując powyższy pseudokod algorytmu szybkiego potęgowania od lewej do prawej, możemy zauważyć, że liczba mnożeń zależy od liczby bitów w binarnej reprezentacji wykładnika. Złożoność algorytmu wynosi $O(\log_2 n)$.

Ważne!

Obie zaprezentowane metody mają swoje wersje rekurencyjne. Ich omówienie oraz porównanie z wersjami iteracyjnymi znajdziesz w e-materiale [Analiza podejścia rekurencyjnego i iteracyjnego](#).

Praca domowa

Zapisz zaprezentowane w e-materiale algorytmy, wykorzystując wybrany język programowania.

Słownik

rekurencja

technika programowania polegająca na odwoływaniu się procedur lub funkcji do samych siebie, aż do momentu spełnienia warunku podstawowego

iteracja

technika programowania polegająca na powtarzaniu tych samych operacji w pętli określoną liczbę razy lub do momentu, aż zostanie spełniony zadany warunek

schemat Hornera

sposób obliczania wartości wielomianu dla danej wartości argumentu wykorzystujący minimalną liczbę mnożeń

system dwójkowy (system binarny)

pozycyjny system liczbowy; sposób zapisywania liczb, w którym potrzebne są tylko dwie cyfry: 0 i 1, a podstawą tego systemu jest liczba 2

szybkie potęgowanie

algorytm pozwalający na szybkie obliczenie potęgi liczby; wykorzystuje wykładnik zapisany w systemie dwójkowym

Gra edukacyjna

Polecenie 1

Sprawdź swoją wiedzę na temat algorytmów iteracyjnych i liczbowych, biorąc udział w grze edukacyjnej.



Test

Sprawdź swoją wiedzę, biorąc udział w grze

Poziom trudności:

łatwy

Limit czasu:

7 min

Twój ostatni wynik:

—

Uruchom

Polecenie 2

Zastanów się, które pytania sprawiły ci trudność. Wróć do fragmentów, których dotyczą.

Polecenie 3

Zapisz trzy pytania, które mogłyby zostać dołączone do testu. Pytania omówicie na forum klasy, a następnie wspólnie na nie udzielcie odpowiedzi.

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Podaj definicję iteracji.

Ćwiczenie 2



Wskaż, które twierdzenia na temat szybkiego potęgowania liczb są prawdziwe.

☐

Wykorzystuje wykładnik w systemie binarnym.

☐

Jest sposobem obliczania wartości wielomianu.

☐

Jest to algorytm istniejący jedynie w wersji rekurencyjnej.

☐

Jest szybszy od sposobu potęgowania według definicji.

Ćwiczenie 3



Połącz w pary pojęcia z ich definicjami.

schemat Hornera

pozycyjny system liczbowy, w którym zapisujemy każdą liczbę za pomocą dwóch cyfr: zer oraz jedynek

szybkie potęgowanie

metoda pozwalająca na szybkie obliczenie potęgi o wykładniku naturalnym

system dwójkowy

technika programowania polegająca na odwoływaniu się procedur lub funkcji do samych siebie, aż do momentu spełnienia warunku podstawowego

rekurencja

sposób obliczania wartości wielomianu, który zmniejsza liczbę potrzebnych mnożeń

Ćwiczenie 4



Wskaż, ile wynosi złożoność obliczeniowa algorytmu szybkiego potęgowania liczb.

☐ $O(\sqrt{n})$

☐ $O(n^2)$

☐ $O(\log_2 n)$

☐ $O(n)$

Ćwiczenie 5



Uzupełnij brakujący fragment kodu.

```
1 potęga ← 1
2 dla i = n, n - 1, ... 1 wykonuj
3     (...)
4 wypisz potęga
```

☐ 1 potęga ← potęga ** a

☐ 1 potęga ← a ** a

☐ 1 potęga ← potęga * a

☐ 1 potęga ← a * a

Ćwiczenie 6



Uzupełnij wartości w działaniach kolejnych iteracji pętli w algorytmie szybkiego potęgowania 3^{1001} . Wykładnik zapisany został w systemie dwójkowym. Wynik będzie przechowywany w zmiennej potęga. Zmienna ta została już zainicjowana wartością 1.

• potęga ← · 1 ·

• potęga ← ·

• potęga ← ·

• potęga ← · · 3



Napisz pseudokod algorytmu wyświetlającego liczbę wszystkich możliwych kodów, które możemy ustawić, aby zabezpieczyć pewną kłódkę. Kod ten składa się z n cyfr z zakresu $\langle 0, 9 \rangle$ i z m liter ze zbioru $\{A, B, C, D\}$.

Litery i cyfry w szyfrze mogą się powtarzać.

Przykład:

Jeżeli kod składa się z dwóch cyfr całkowitych z przedziału $\langle 0, 7 \rangle$ i trzech liter ze zbioru $\{A, B, C, D\}$, to liczba możliwych kodów to $8 \cdot 8 \cdot 4 \cdot 4 \cdot 4$. Ósemki w kodzie oznaczają liczbę kombinacji cyfr (od 0 do 7 mamy siedem cyfr). Są dwie, ponieważ kod składa się z dwóch cyfr. Analogicznie jest w przypadku liter – do wyboru mamy cztery litery (A, B, C lub D), każda z nich może pojawić się w jednym z czterech miejsc kodu.

Wykorzystaj algorytm potęgowania liczb według definicji.

Specyfikacja problemu:

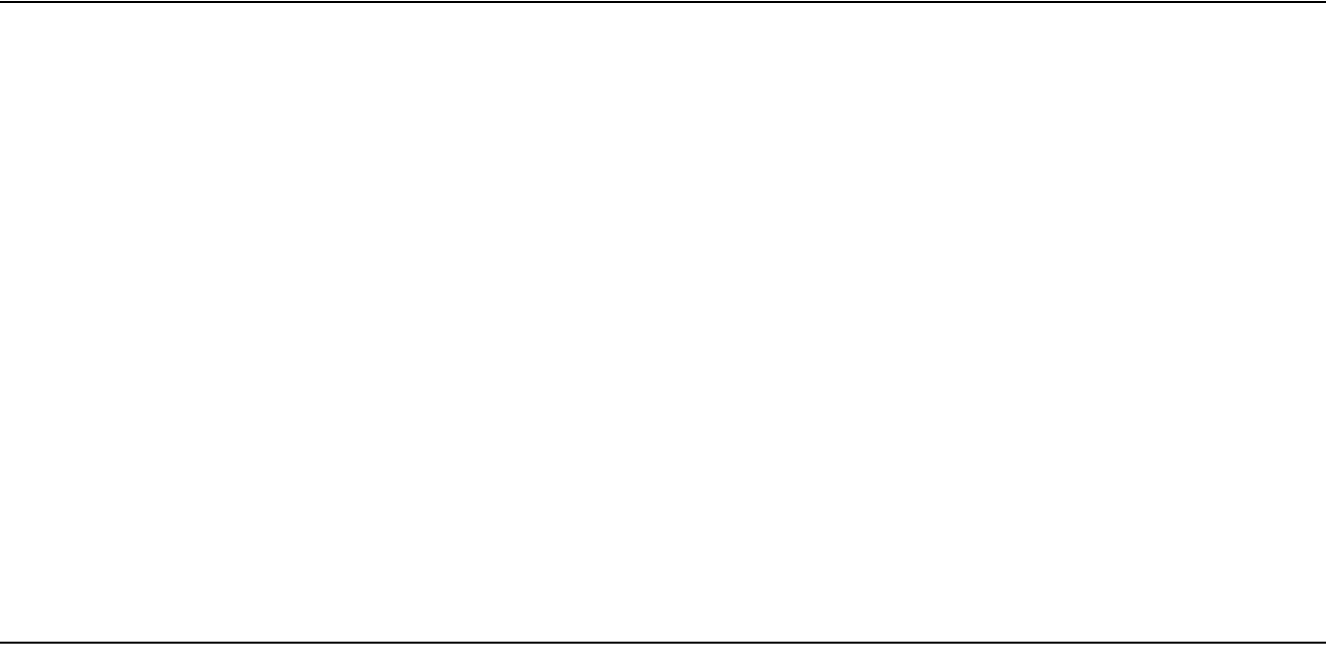
Dane:

- n – liczba naturalna; liczba cyfr zawartych w kodzie
- m – liczba naturalna; liczba liter zawartych w kodzie

Wynik:

Algorytm wypisuje liczbę możliwych kodów zabezpieczających kłódkę.

Napisz pseudokod algorytmu wyświetlającego liczbę wszystkich możliwych kodów, które możemy ustawić, aby zabezpieczyć pewną kłódkę.





Okres połowicznego rozpadu jest to czas, po którym liczebność danej próbki maleje dwukrotnie. Na przykład po trzech okresach pozostanie $\frac{1}{8}$ początkowej ilości próbki. Ilość próbki N_t pozostającej po danej liczbie okresów t rozpadu początkowej próbki N_0 możemy wyrazić wzorem:

$$N_t = N_0 \cdot \frac{1}{2^t}$$

Zapisz w postaci pseudokodu algorytm wyliczania ilości pozostającej próbki N_t po liczbie okresów t zapisanych w systemie dwójkowym. Wykorzystaj algorytm szybkiego potęgowania liczb w wersji iteracyjnej.

Specyfikacja problemu:

Dane:

- t – liczba okresów, liczba naturalna zapisana w systemie binarnym
- N_0 – początkowa ilość próbki, liczba naturalna

Wynik:

Algorytm wypisuje ilość pozostającej próbki po liczbie okresów t :

1



Dla nauczyciela

Autor: Zespół autorski Contentplus.pl sp. z o.o.

Przedmiot: Informatyka

Temat: Algorytmy iteracyjne i liczbowe – potęgowanie liczb

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 rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

2) do realizacji rozwiązania problemu dobiera odpowiednią metodę lub technikę algorytmiczną i struktury danych;

3) objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność;

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:

h) obliczania wartości wielomianu za pomocą schematu Hornera,

i) szybkiego potęgowania liczb w wersji iteracyjnej i rekurencyjnej,

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 algorytm iteracyjnego potęgowania liczb.
- Zaimplementujesz program potęgujący liczby wykorzystujący algorytm iteracyjny.
- Scharakteryzujesz i zaimplementujesz algorytm szybkiego potęgowania liczb.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

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

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.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Algorytmy iteracyjne i liczbowe – potęgowanie liczb”. Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Przeczytaj”.

Faza wstępna:

1. Nauczyciel wyświetla temat i cele zajęć zawarte w sekcji „Wprowadzenie”. Prosi, by na ich podstawie uczniowie sformułowali kryteria sukcesu.
2. **Rozpoznanie wiedzy uczniów.** Uczniowie tworzą pytania dotyczące tematu zajęć, na które odpowiedzą w trakcie lekcji.

Faza realizacyjna:

1. **Praca z tekstem.** Jeżeli przygotowanie uczniów do lekcji jest niewystarczające, nauczyciel prosi o indywidualne zapoznanie się z treścią zawartą w sekcji „Przeczytaj”. Każdy uczestnik zajęć podczas cichego czytania wynotowuje najważniejsze kwestie poruszane w tekście.
2. **Praca z multimediami.** Nauczyciel wyświetla zawartość sekcji „Gra edukacyjna”. Uczniowie indywidualnie odpowiadają na 10 pytań.
3. W kolejnym etapie uczniowie dobierają się w pary i wykonują ćwiczenia nr 1-6 z sekcji „Sprawdź się”. Następnie konsultują swoje rozwiązania z inną parą uczniów i ustalają jedną wersję odpowiedzi.

Faza podsumowująca:

1. Nauczyciel wyświetla na tablicy temat lekcji i cele zawarte w sekcji „Wprowadzenie”. W kontekście ich realizacji podsumowuje przebieg zajęć, a także wskazuje mocne i słabe strony pracy uczniów. Nauczyciel zadaje pytania podsumowujące, np.
 - na czym polega szybkie potęgowanie?
 - jak nazywamy odwoływanie się definicji lub funkcji do samej siebie?
 - na czym polega dwójkowy system zapisywania liczb?
 - do czego służy schemat Hornera?

Praca domowa:

1. Uczniowie wykonują ćwiczenia 7-8 z sekcji „Sprawdź się”.

Wskazówki metodyczne:

- Uczniowie mogą wykorzystać treści w sekcjach: „Przeczytaj”, „Gra edukacyjna”, „Sprawdź się” jako materiał do lekcji powtórkowej.