



Rekurencja – ćwiczenia

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Schemat interaktywny](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Źródło: Pixabay, domena publiczna.

Rekurencja jest metodą rozwiązywania problemów, która korzysta z rozwiązań innych problemów. Jej wyjątkowość polega na tym, że za „innymi problemami” kryje się problem właściwy, najczęściej rozwiązywany dla mniejszych rozmiarów tych samych danych. Więcej informacji na ten temat znajdziesz w e-materiale [Rekurencja](#).

O tym, jak zagadnienie rekurencji wyjaśnia matematyka, przeczytasz w e-materiałach:

- [Ciąg określony rekurencyjnie](#),
- [Ciąg geometryczny określony rekurencyjnie](#),
- [Wzór ogólny ciągu określonego rekurencyjnie](#),
- [Ciąg arytmetyczny określony wzorem rekurencyjnym](#).

Implementację omawianego algorytmu w poszczególnych językach programowania przedstawiamy w e-materiałach:

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

Więcej zadań? Sięgnij do: [Rekurencja w zadaniach](#).

Twoje cele

- Przeanalizujesz kilka problemów, które zostaną rozwiązane za pomocą rekurencji.
- Napiszesz własne programy z zastosowaniem techniki rekurencyjnej, wykorzystując do tego pseudokod.
- Rozwiążesz zadania w celu sprawdzenia i utrwalenia wiadomości na temat rekurencji.

Przeczytaj

Silnia

Zacznijmy od algorytmu wyznaczania [silni](#) z liczby naturalnej. Zastosujemy w pseudokodzie rekurencję – funkcja będzie wywoływała samą siebie aż do momentu dojścia do [przypadku podstawowego](#). Przy pisaniu pseudokodu pomocna okaże się definicja rekurencyjna silni.

$$n! = \begin{cases} 1 & \text{gdy } n = 0 \\ (n-1)!n & \text{gdy } n > 0 \end{cases}$$

Zgodnie z tą definicją funkcja będzie wywoływała samą siebie z parametrem zmniejszonym o 1, aż do momentu, gdy n będzie równe 0 – wtedy zostanie zwrócona wartość 1. Pseudokod funkcji rekurencyjnej realizującej algorytm wyznaczania silni liczby naturalnej n wygląda następująco:

```
1 funkcja silnia(n)
2     jeżeli n == 0 wykonaj:
3         zwróć 1
4     w przeciwnym razie wykonaj:
5         zwróć silnia(n-1) * n
```

Ćwiczenie 1

Zastanówmy się, jaki będzie schemat kolejnych wywołań rekurencyjnych w przypadku wyznaczenia silni liczby 6.

Najpierw będą rekurencyjnie wywoływane funkcje z coraz mniejszymi wartościami, czyli 5, 4, 3, 2, 1, aż dojdziemy do 0. Wtedy nastąpi zwrócenie wartości 1 i zaczniemy po kolei wychodzić z zagnieżdżenia wywołań funkcji, za każdym razem zwracając wynik danej silni. Na koniec osiągniemy szukaną wartość 6!:

n	silnia(n)
6	720
5	120
4	24
3	6
2	2
1	1
0	1

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

Wyraz ciągu rekurencyjnego

Ćwiczenie 2

Napiszmy funkcję `funkcja(n)` przyjmującą parametr naturalny n , która w przypadku n parzystego zwróci `funkcja(n-2) - 3`, natomiast w przypadku n nieparzystego zwróci `funkcja(n-1) mod 9`. Dodatkowo dla $n < 2$ zawsze powinna zostać zwrócona wartość 122.

$$a_n = \begin{cases} 122 & \text{gdy } n < 2 \\ a_{n-2} - 3 & \text{gdy } n \bmod 2 = 0 \\ a_{n-1} \bmod 9 & \text{gdy } n \bmod 2 \neq 0 \end{cases}$$

Rozpocznijmy od warunku, który mówi o tym, że dla n mniejszego od 2 zwracana jest wartość 122.

```

1 funkcja funkcja(n)
2     jeżeli n < 2 wykonaj:
3         zwróć 122

```

Natomiast jeżeli n NIE JEST mniejsze od 2, musimy sprawdzić, czy n jest liczbą parzystą, i umieścić odpowiednie instrukcje w przypadku spełnienia tego warunku.

```

1 funkcja funkcja(n)
2     jeżeli n < 2 wykonaj:
3         zwróć 122
4     w przeciwnym razie wykonaj:
5         jeżeli n % 2 == 0 wykonaj:
6             zwróć funkcja(n-2) - 3

```

Jeżeli n nie jest liczbą parzystą, musimy zwrócić $\text{funkcja}(n-1) \bmod 9$.

```

1 funkcja funkcja(n)
2     jeżeli n < 2 wykonaj:
3         zwróć 122
4     w przeciwnym razie wykonaj:
5         jeżeli n % 2 == 0 wykonaj:
6             zwróć funkcja(n-2) - 3
7         w przeciwnym razie wykonaj:
8             zwróć funkcja(n-1) mod 9

```

Zapisaaliśmy za pomocą pseudokodu funkcję zwracającą n -ty wyraz ciągu rekurencyjnego.

Ćwiczenie 3

Dla powyższego pseudokodu obliczmy zwracane wartości dla n z zakresu od 0 do 7.

$$\text{funkcja}(0) = 122$$

$$\text{funkcja}(1) = 122$$

$$\text{funkcja}(2) = \text{funkcja}(0) - 3 = 122 - 3 = 119$$

$$\text{funkcja}(3) = \text{funkcja}(2) \bmod 9 = 119 \bmod 9 = 2$$

$$\text{funkcja}(4) = \text{funkcja}(2) - 3 = 119 - 3 = 116$$

$$funkcja(5) = funkcja(4) \bmod 9 = 116 \bmod 9 = 8$$

$$funkcja(6) = funkcja(4) - 3 = 116 - 3 = 113$$

$$funkcja(7) = funkcja(6) \bmod 9 = 113 \bmod 9 = 5$$

Funkcja wywołuje samą siebie we wszystkich przypadkach poza tymi, kiedy $n = 0$ lub $n = 1$. Na przykład `funkcja(5)` wywołuje funkcję `funkcja(4)`. Nie musimy jednak ponownie wyznaczać wartości tej funkcji, ponieważ obliczyliśmy ją w poprzednim kroku.

Ciąg rekurencyjny

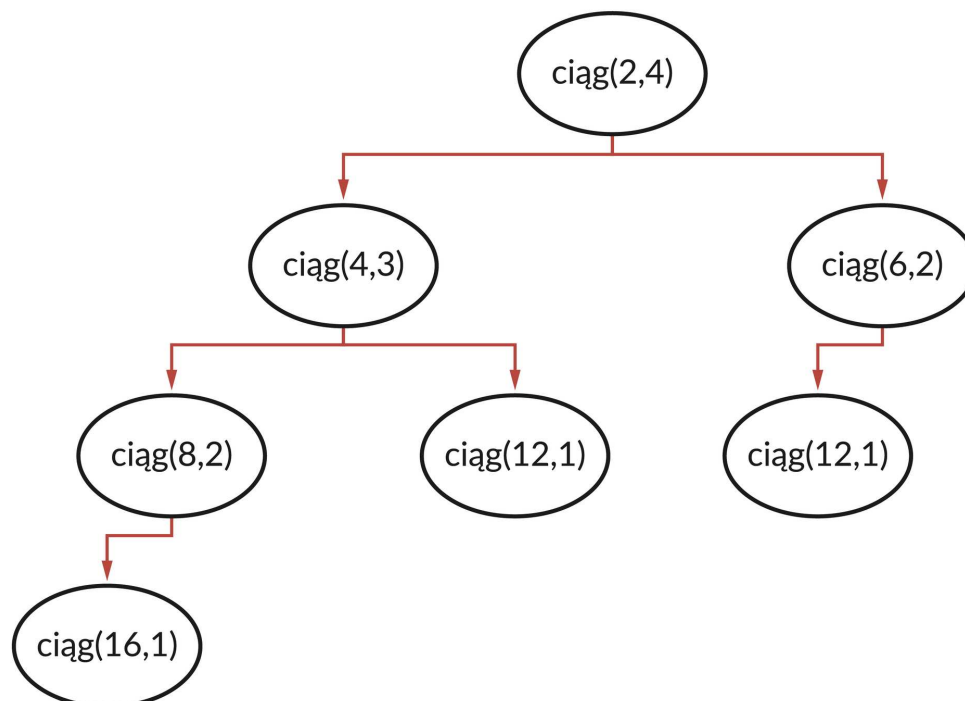
Przeanalizujmy przedstawioną za pomocą pseudokodu funkcję `ciąg(x, y)` przyjmującą parametry x i y typu całkowitego, której wywołanie spowoduje wypisanie ciągu liczb całkowitych:

```
1 funkcja ciąg(x, y)
2     jeżeli y > 0 wykonaj:
3         ciąg(2*x, y-1)
4         ciąg(3*x, y-2)
5         wypisz x - y
```

Ćwiczenie 4

Wypiszmy wszystkie wywołania rekurencyjne funkcji `ciąg(2, 4)`.

Do wykonania tego zadania pomocne okaże się drzewo wywołań rekurencyjnych, w którym umieścimy wszystkie wywołania funkcji `ciąg(2, 4)`.



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

W drzewie zostały przedstawione wywołania tylko tych funkcji, których wartość drugiego przyjmowanego argumentu, czyli y , jest większa od 0. Z tego względu wykonania niektórych funkcji powodują wywołanie tylko jednej rekurencyjnej funkcji. Lewa strzałka każdej z funkcji prowadzi do wywołania funkcji $\text{ciąg}(2 \cdot x, y - 1)$, natomiast prawa prowadzi do $\text{ciąg}(3 \cdot x, y - 2)$.

Ćwiczenie 5

Podajmy teraz wynik działania funkcji $\text{ciąg}(2, 4)$ i uzupełnijmy wygenerowany ciąg rekurencyjny:

15 6

Ponownie skorzystamy z drzewa wywołań rekurencyjnych. Wiemy, że nasz ciąg to uzyskane wartości działań $x - y$. Jednak w jakiej kolejności wypisywane są te wartości? Spójrzmy ponownie na zapisaną za pomocą pseudokodu funkcję $\text{ciąg}(x, y)$. Najpierw zostanie wywołana rekurencyjnie funkcja $\text{ciąg}(2 \cdot x, y - 1)$, czyli lewa skierowana krawędź, następnie $\text{ciąg}(3 \cdot x, y - 2)$, czyli prawa krawędź, a na końcu wypisana zostanie wartość $x - y$.

A zatem początek wygenerowanego ciągu rekurencyjnego zgadza się z naszymi wnioskami. W pierwszej kolejności wykonywane jest najniżej położone, lewe poddrzewo, zatem wywołana zostanie funkcja $\text{ciąg}(16, 1)$.

$$16 - 1 = 15$$

Następnie miałyby zostać wykonane prawe poddrzewo, jednak takie nie istnieje, przechodzimy więc do wyższego poziomu, czyli $\text{ciąg}(8, 2)$. Stosując powyższe obserwacje, uzyskamy następujący ciąg:

15 6 11 1 11 4 - 2

Słownik

przypadek podstawowy

przypadek, dla którego funkcja rekurencyjna nie wywołuje samej siebie (warunek zakończenia), zwraca za to konkretną, wyliczoną wartość; jest to pewien założony stan początkowy

silnia

$n!$ – iloczyn kolejnych liczb naturalnych, mniejszych lub równych n

Schemat interaktywny

Polecenie 1

Stwórz algorytm znajdujący wartość n -tego wyrazu ciągu zdefiniowanego wzorem rekurencyjnym:

$$a_n = \begin{cases} 1 & \text{dla } n = 1 \\ 7 & \text{dla } n = 2 \\ 10 & \text{dla } n = 3 \\ a_{n-1} + a_{n-2} & \text{dla parzystych } n > 3 \\ a_{n-1} \bmod 7 & \text{dla nieparzystych } n > 3 \end{cases}$$

Specyfikacja problemu:

Dane:

- n – liczba naturalna

Wynik:

Program wypisuje wartość n -tego wyrazu ciągu.

Przetestuj działanie programu dla 12. wyrazu podanego ciągu.

Jeśli chcesz przećwiczyć umiejętności ze szkoły podstawowej, przygotuj schemat blokowy programu. W innym wypadku napisz program w wybranym języku programowania.

Polecenie 2

Dodaj do swojego programu komentarze tak, żeby był zrozumiały dla osoby, która nie potrafi programować.

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Zapisz za pomocą pseudokodu algorytm wyznaczający metodą rekurencyjną sumę n kolejnych liczb naturalnych.

Specyfikacja problemu:

Dane:

- n – liczba naturalna

Wynik:

Program wypisuje sumę n kolejnych liczb naturalnych.

1

Ćwiczenie 2



Wskaż, ile nastąpi wywołań funkcji `Silnia()` w programie wyznaczania silni dla $n=6$.

☐ 7

☐ 5

☐ 1

☐ 6

Ćwiczenie 3



Zapisz algorytm wyznaczający metodą rekurencyjną wartość potęgi o wykładniku całkowitym za pomocą pseudokodu.

Specyfikacja problemu:

Dane:

- a – podstawa potęgi
- n – liczba całkowita; wykładnik

Wynik:

Program wypisuje wartość potęgi o wykładniku całkowitym.

1

Ćwiczenie 4



Zapoznaj się z pseudokodem i wykonaj ćwiczenie.

```
1 funkcja(n)
2     jeżeli n = 1
3         zwróć 1
4     jeżeli n = 2
5         zwróć 2
6     zwróć funkcja(n - 1) * funkcja(n - 2)
```

Wskaż, jaka wartość zostanie zwrócona, gdy wywołamy funkcję `funkcja(6)`.

☐ 8

☐ 16

☐ 32

☐ 256

Ćwiczenie 5



Zapoznaj się z pseudokodem i wykonaj ćwiczenie.

```
1 funkcja(n)
2     jeżeli n < 2
3         zwróć 1
4     w przeciwnym razie
5         jeżeli n mod 2 = 0
6             zwróć funkcja(n - 2) * funkcja(n - 1)
7         w przeciwnym razie
8             zwróć funkcja(n-1) + 8
```

Wskaż, jakie wartości nie zostaną zwrócone dla funkcji `funkcja(4)`.

☐ 153

☐ 17

☐ 18

☐ 9

Ćwiczenie 6



Uzupełnij tabelę dla funkcji przedstawionej w poprzednim ćwiczeniu.

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

Ćwiczenie 7



Zapoznaj się z pseudokodem i wykonaj ćwiczenie.

```
1 ciąg(x, y)
2   jeżeli y > 0
3       ciąg(2 + x, y - 1)
4       wypisz x
5       ciąg(x + 4, y - 1)
```

Wskaż długość ciągu liczbowego otrzymanego w wyniku wywołania funkcji `ciąg(3, 4)`.

☐ 15

☐ 16

☐ 7

☐ 8

Ćwiczenie 8



Zapoznaj się z pseudokodem i wykonaj ćwiczenie.

```
1 ciąg(x, y)
2   jeżeli y > 0
3       ciąg(2 + x, y - 1)
4       wypisz x
5       ciąg(x + 4, y - 1)
```

Wskaż, który wzór opisuje długość ciągu liczbowego otrzymanego w wyniku wywołania poniższej funkcji.

☐ $y^2 - 1$

☐ $2^y - 1$

☐ $x^2 + 1$

☐ $2^x - 1$

Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Rekurencja – ćwiczenia

Grupa docelowa:

Szkoła ponadpodstawowa, liceum ogólnokształcące, technikum, zakres 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 + II. Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

3) objaśnia, a także porównuje podstawowe metody i techniki algorytmiczne oraz struktury danych, wykorzystując przy tym przykłady problemów i algorytmów, w szczególności:

b) rekurencję (do generowania ciągów liczb, potęgowania, sortowania liczb, generowania fraktali),

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 kilka problemów, które zostaną rozwiązane za pomocą rekurencji.

- Napiszesz własne programy z zastosowaniem techniki rekurencyjnej, wykorzystując do tego pseudokod.
- Rozwiążesz zadania w celu sprawdzenia i utrwalenia wiadomości na temat rekurencji.

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: „Rekurencja – ćwiczenia”. Uczniowie mają zapoznać się z treściami w sekcji „Przeczytaj”. Nauczyciel prosi również o wykonanie ćwiczenia nr 1.

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.** Uczniowie tworzą pytania dotyczące tematu zajęć, na które odpowiedzą w trakcie lekcji.

Faza realizacyjna:

1. Uczniowie zapoznają się z treścią sekcji „Przeczytaj”. Następnie powtarzają zaprezentowane rozwiązania ćwiczeń 2-5 na swoich komputerach.

2. **Praca z multimediami** Nauczyciel wyświetla zawartość sekcji „Schemat interaktywny”, czyta treść polecenia nr 1 i omawia kolejne kroki rozwiązania.
3. **Ćwiczenie umiejętności.** Prowadzący zapowiada uczniom, że w kolejnym kroku będą rozwiązywać ćwiczenia nr 2-8 z sekcji „Sprawdź się”. Każdy z uczniów robi to samodzielnie. Po ustalonym czasie wybrani uczniowie przedstawiają rozwiązania. Nauczyciel w razie potrzeby koryguje odpowiedzi, dopowiada istotne informacje, udziela uczniom informacji zwrotnej.

Faza podsumowująca:

1. Na koniec zajęć nauczyciel prosi uczniów o rozwinięcie zdania: „Na dzisiejszych zajęciach nauczyłam/łem się jak...”.

Praca domowa:

1. Uczniowie wykonują ćwiczenie nr 1 z sekcji „Sprawdź się”.

Wskazówki metodyczne:

- Treści w sekcji „Schemat interaktywny” można wykorzystać na lekcji jako podsumowanie i utrwalenie wiedzy uczniów.