

# Algorytmy rekurencyjne

wszelkie prawa zastrzeżone  
zakaz kopiowania, publikowania i przechowywania  
all rights reserved  
no copying, publishing or storing

Maciej Hojda

**Uwaga:** Słowa „dany”, „zadany”, „podany”, „wybrany”, „ustalony” itd. w kontekście parametrów (zmiennych) oznacza parametr zadany przez użytkownika (a nie na stałe, przez programistę), a implementacja wykorzystująca taki parametr powinna obsługiwać jego różne wartości.

## 1 Zadanie nr 1

Dla następujących ciągów liczbowych

1.  $\forall n \in \{1, 2, \dots\} : x(n) = 3^n + x(n-1)$   
 $x(0) = 1$
2.  $\forall n \in \{1, 2, \dots\} : x(n) = n + x(n-2)$   
 $x(-1) = x(0) = 0$
3.  $\forall n \in \{2, 3, \dots\} : x(n) = x(n-1) + x(n-2)$   
 $x(1) = 1; x(0) = 0$   
(ciąg Fibonacciego)

wykonaj:

1. zaimplementuj (rekurencyjny) algorytm wyliczający wartość  $n$ tego elementu ciągu,
2. analitycznie wyznacz wzór na wartość  $n$ tego elementu ciągu (np. indukcyjnie),
3. napisz procedurę weryfikującą poprawność zaimplementowanej rekurencji (wyświetlającą i porównującą wynik numeryczny i analityczny) dla  $N$  pierwszych elementów ciągu ( $N$  zadane przez użytkownika).

Wejście:  $N$ .

Wyjście: zestawienie wartości wyliczanych algorytmem rekurencyjnym i ze wzoru.

## 2 Zadanie nr 2

Dla listy liczb zadanej przez użytkownika, zaimplementuj algorytm (algorytmy) rekurencyjne

1. znajdujący największy element na liście,
2. znajdujący drugi największy element na liście,
3. obliczający średnią elementów na liście.

Algorytmy dzielą listę na dwie (w przybliżeniu) połowy i uruchamiają się rekurencyjnie na utworzonych połowach.

Wejście: lista liczb.

Wyjście: jak w 1, 2, 3.

Oszacuj złożoność czasową algorytmów.

### 3 Zadanie nr 3

Zaimplementuj algorytm sortowania przez scalanie wykonywany na liście liczb zadanej przez użytkownika.

Wejście: lista liczb

Wyjście: posortowana lista liczb.

### 4 Zadanie nr 4

Dany jest graf nieskierowany, nieważony. Niech długość ścieżki będzie liczbą krawędzi na niej.

Zaimplementuj rekurencyjną metodę wyznaczania najkrótszej ścieżki między dwoma zadanymi wierzchołkami.

Jeśli wierzchołki nie są bezpośrednimi sąsiadami, metoda wywołuje się rekurencyjnie, na każdym z bezpośrednich sąsiadów pierwszego wierzchołka i na drugim wierzchołku.

Jeśli wierzchołki są bezpośrednimi sąsiadami zwracana odległość to 1.

Uwaga: zadbaj o to, żeby wierzchołków nie odwiedzać wielokrotnie i o to, żeby algorytm nie zapętlał się w nieskończoność (a dokładniej, do przepełnienia stosu).

Wejście: graf zadany w pliku (zaproponuj format pliku).

Wyjście: najkrótsza ścieżka i jej długość.

Oszacuj złożoność czasową algorytmu.

### 5 Zadanie nr 5

Wyświetl wykresy czasu działania programów z zadań nr 2 i 3 w zależności od rozmiaru wejścia.

Dane wejściowe (listy liczb) generuj losowo.

Wyjście: wykresy czasu działania.