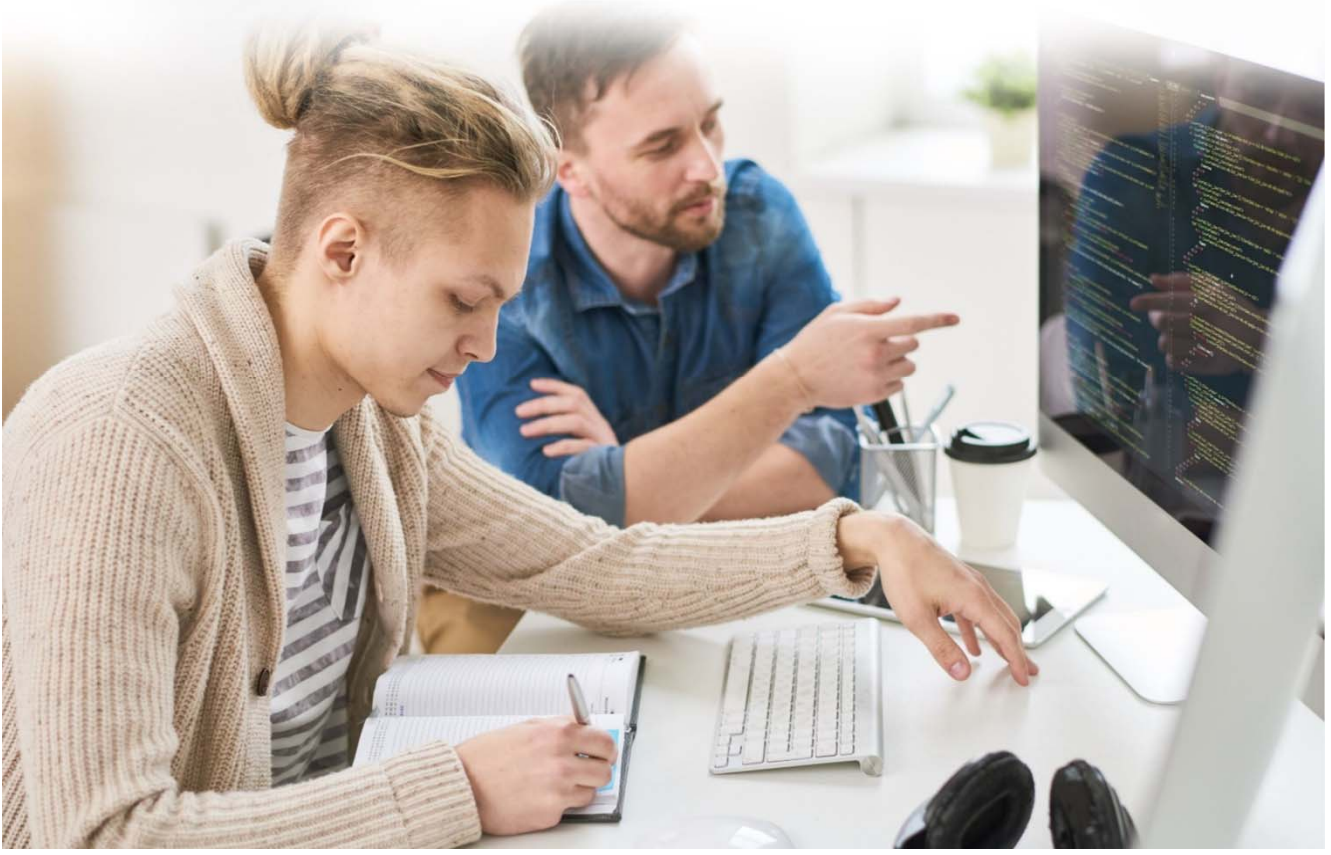
ROZDZIAŁ 4

KLASYFIKACJA ALGORYTMÓW



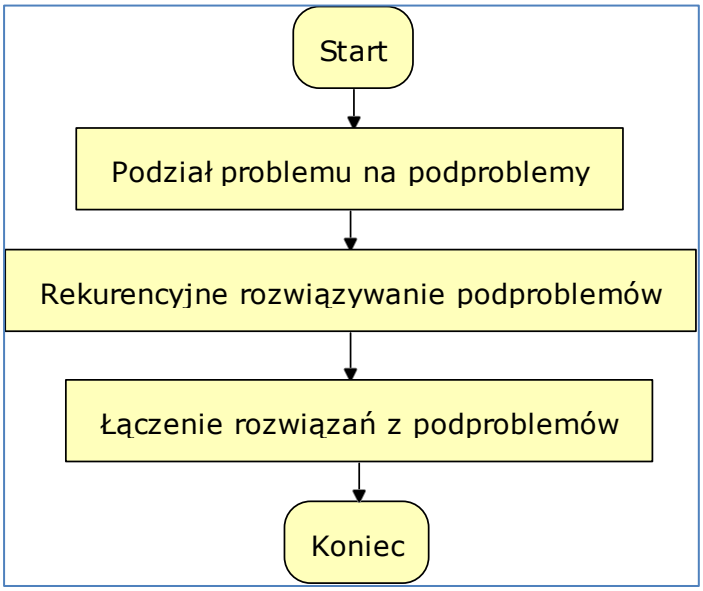
4 Klasyfikacja algorytmów

4.1 Klasyfikacja algorytmów ze względu na ich cel oraz cechy implementacyjne

Algorytmy można sklasyfikować ze względu na ich przeznaczenie, cechy implementacyjne oraz zakres przetwarzanych danych. Są to m.in. następujące klasy:

- dziel i zwyciężaj,
- programowanie liniowe,
- programowanie dynamiczne,
- algorytmy siłowe,
- algorytmy zachłanne,
- algorytmy probabilistyczne.

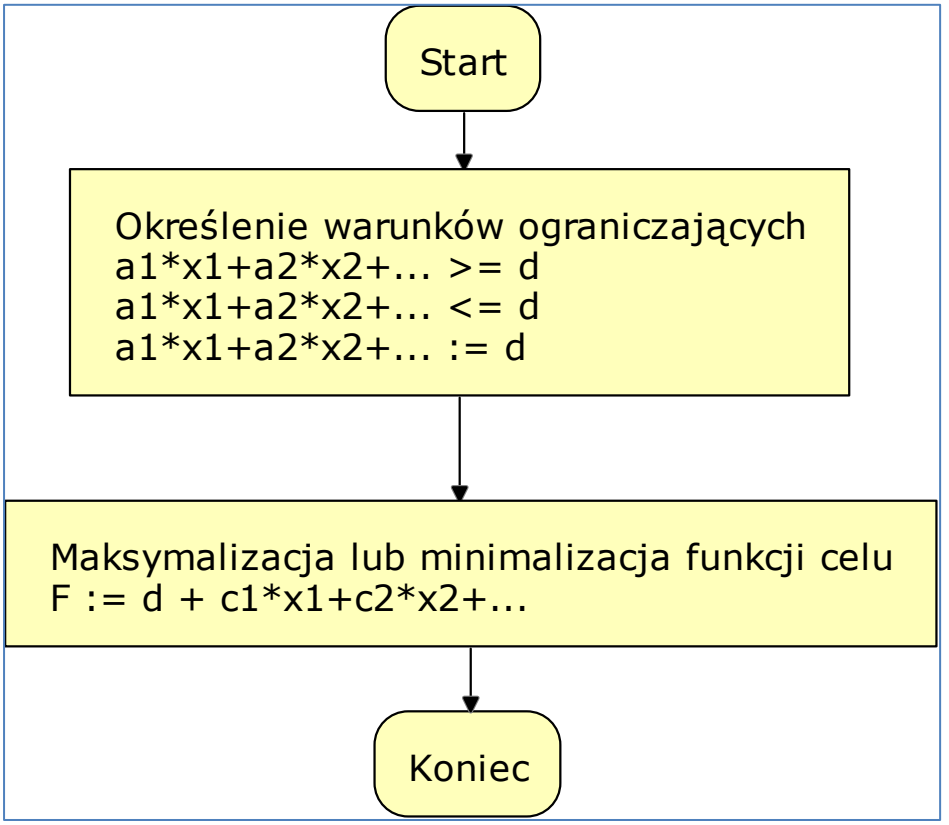
Algorytm dziel i zwyciężaj (łac. *divide et impera*), dzieli problem na mniejsze podproblemy, aż otrzyma problem łatwy do rozwiązania.



Rysunek 4.1. Strategia algorytmu dziel i zwyciężaj

Przykład: algorytm wyszukiwania binarnego.

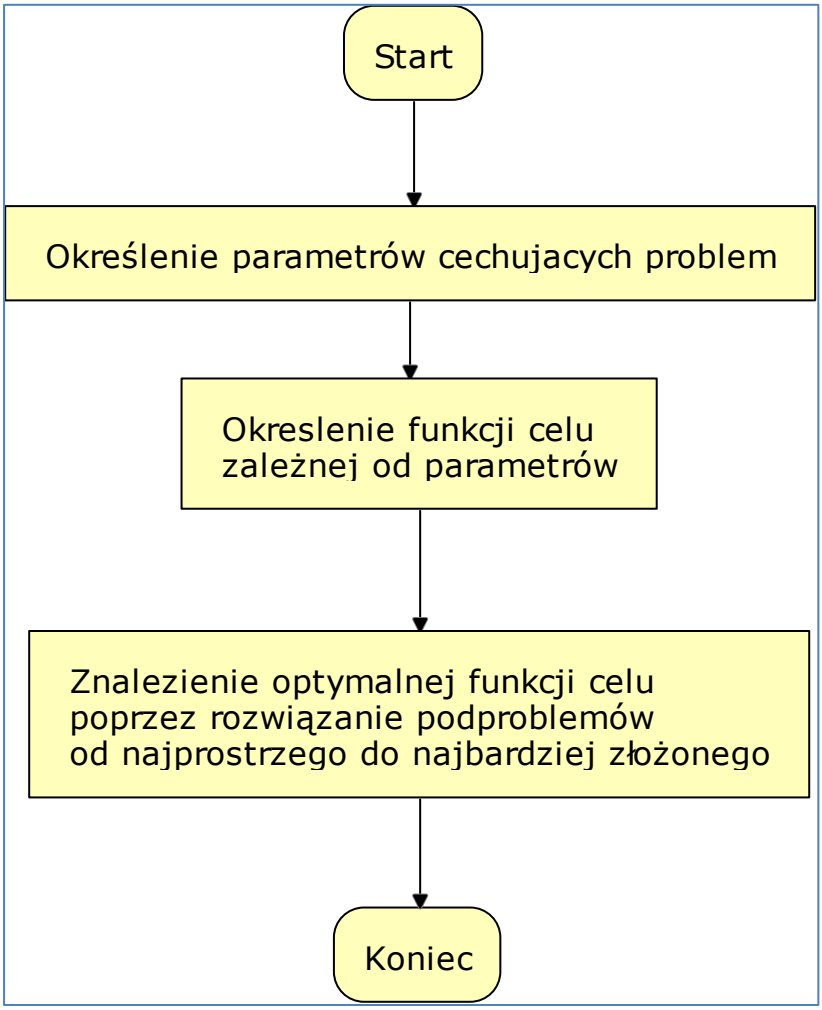
W programowaniu liniowym, do rozwiązania algorytmu stosuje funkcje liniowe.



Rysunek 4.2. Strategia programowania liniowego

Przykład: optymalizacja procesów produkcyjnych.

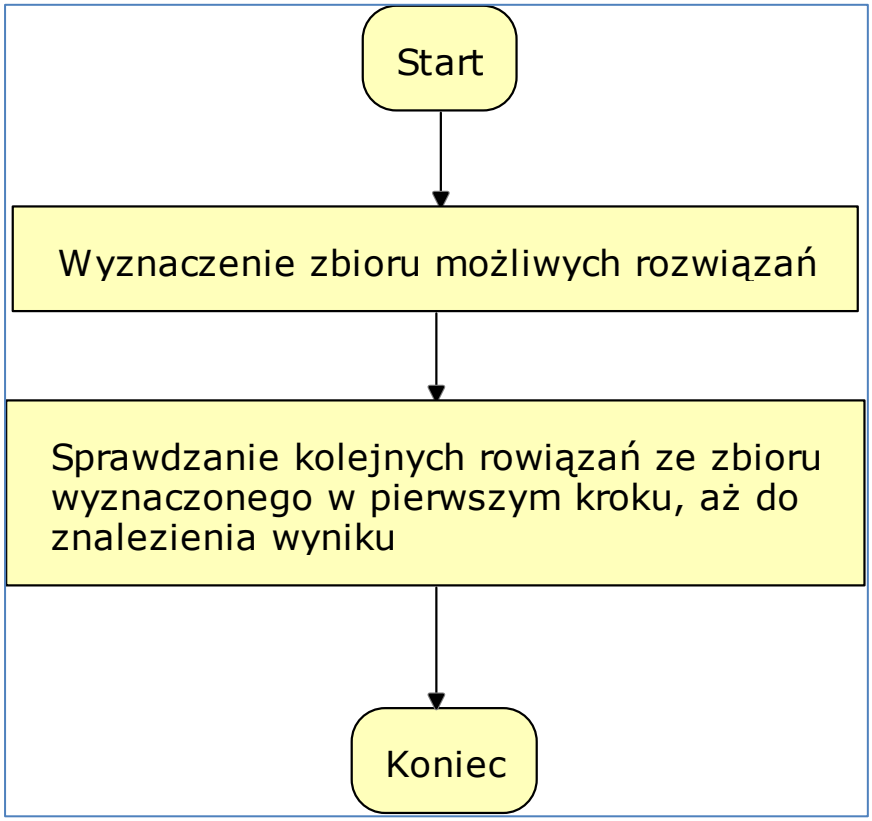
W programowaniu dynamicznym rozwiązywany problem dzieli się na problemy podrzędne względem kilku parametrów, które cechuje nierozłączność.



Rysunek 4.3. Strategia programowania dynamicznego

Przykład: algorytm pakowania plecaka.

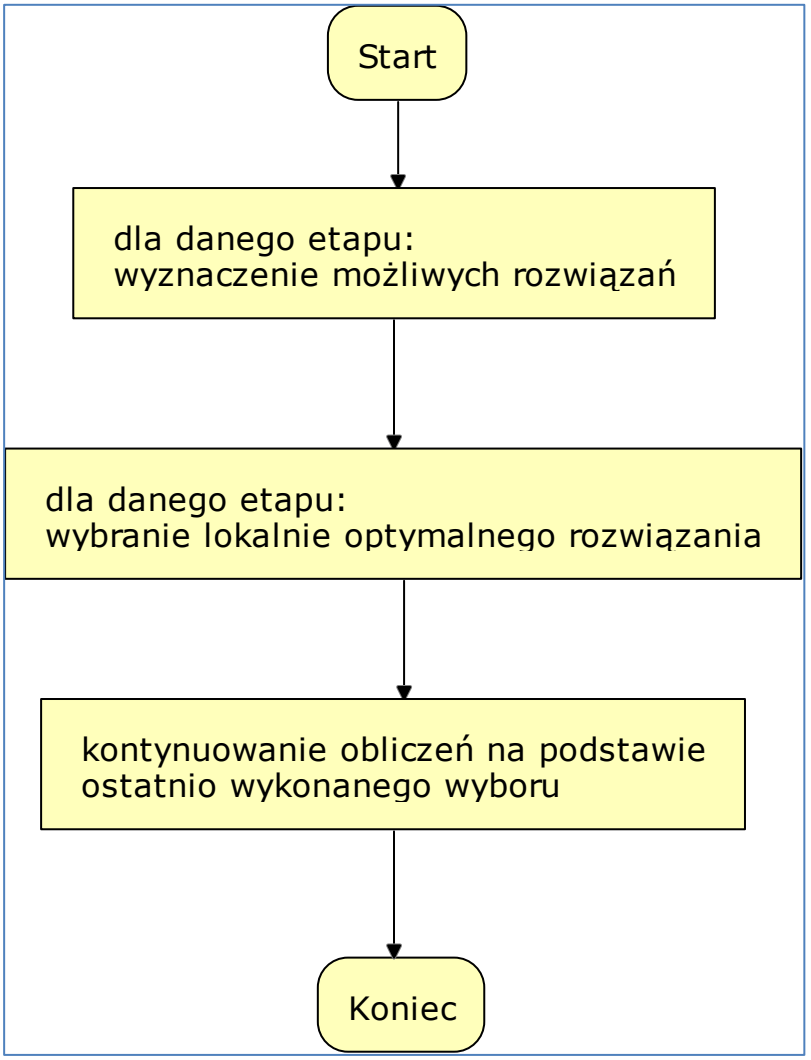
Algorytm siłowy analizuje wszystkie możliwe rozwiązania aż do znalezienia rozwiązania.



Rysunek 4.4. Strategia algorytmu siłowego

Przykład: Znalezienie zaszyfrowanego hasła, za pomocą generowania wszystkich możliwych kombinacji znaków.

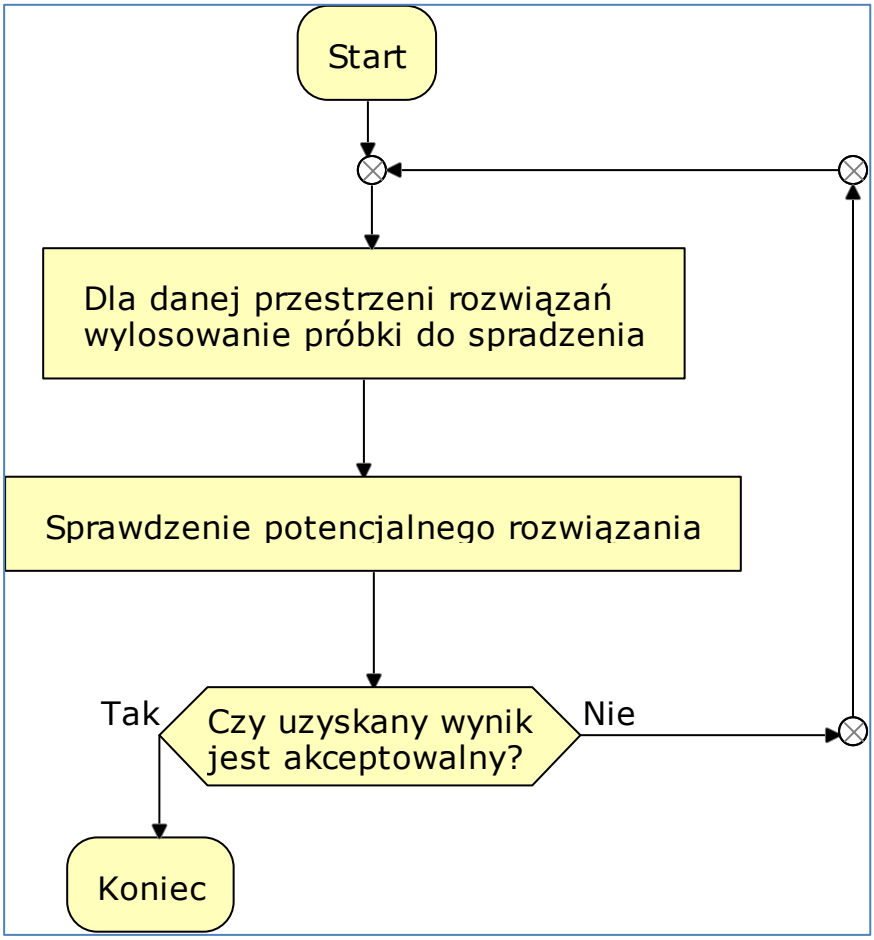
Algorytm zachłanny w każdym kroku wyznacza najlepsze rozwiązanie częściowe.



Rysunek 4.5. Strategia algorytmu zachłannego

Przykład: algorytm wyznaczający najkrótszą ścieżkę w grafie.

Algorytm probabilistyczny polega na pseudolosowym przeszukiwaniu zakresu rozwiązań.



Rysunek 4.6. Strategia algorytmu probabilistycznego

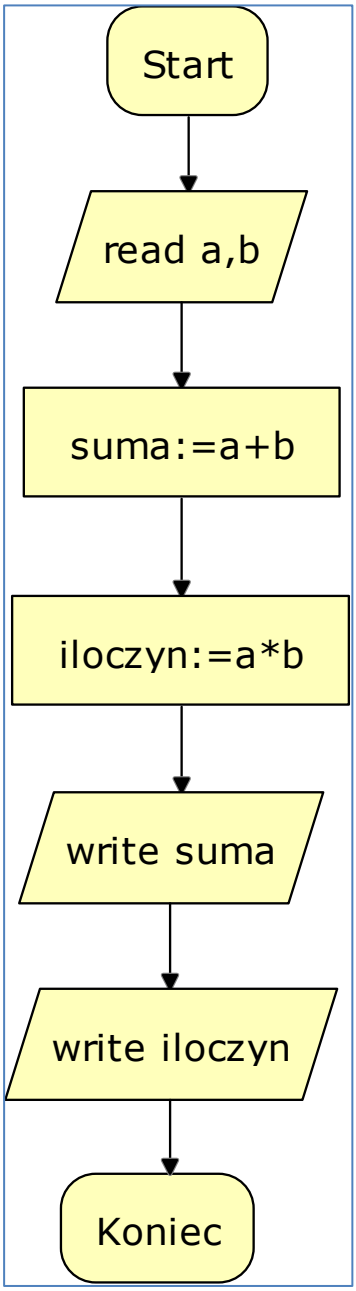
Przykład: Metoda Monte Carlo do obliczania pola figur.

4.2 Klasyfikacja ze względu na kolejność wykonywanych operacji

Algorytmy można podzielić ze względu na kolejność wykonywania operacji i są to:

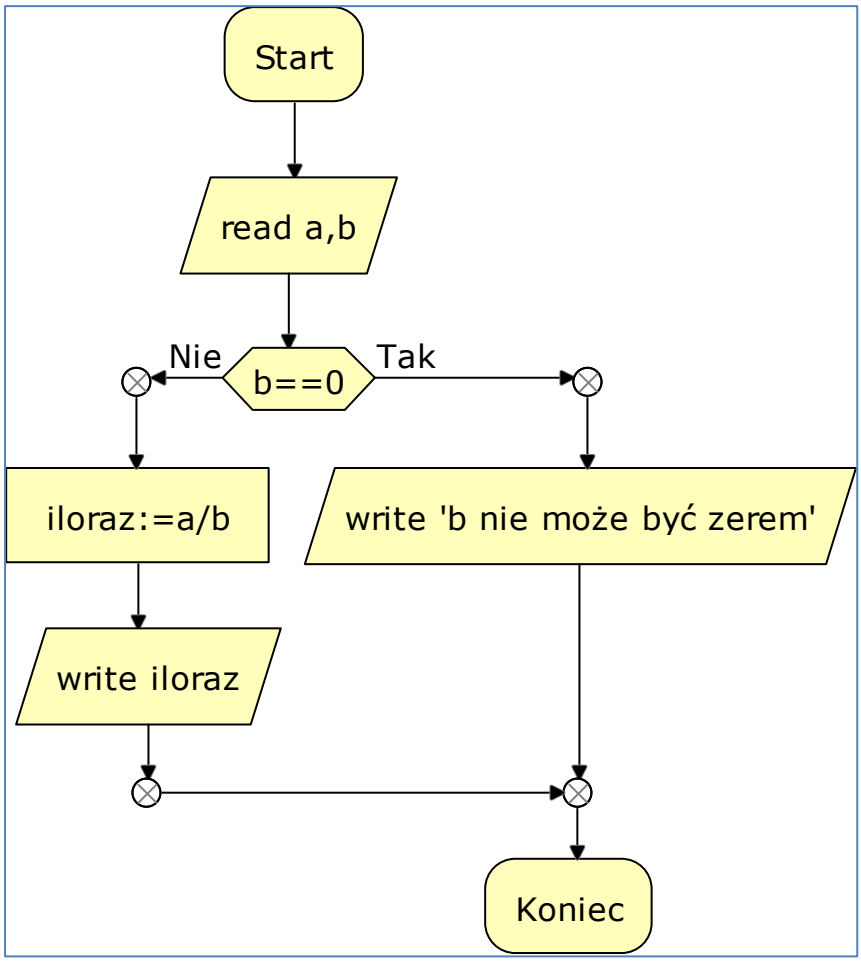
- algorytmy liniowe,
- algorytmy rozgałęzione,
- algorytmy iteracyjne.

Algorytmy liniowe to algorytmy, w którym kolejność wykonywanych operacji jest taka sama i niezależna od danych wejściowych.



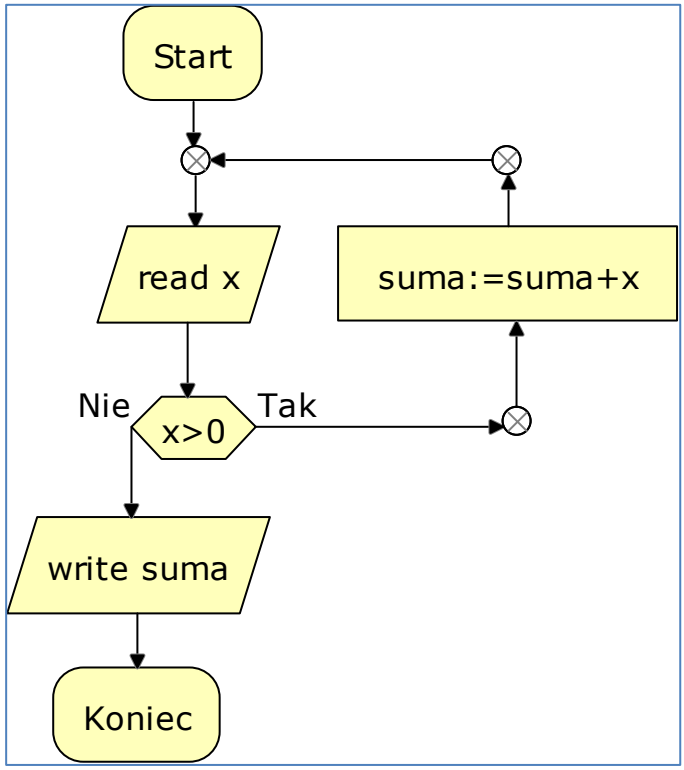
Rysunek 4.7. Przykład algorytmu liniowego

Algorytmy warunkowe (rozgałęzione) to algorytmy, w którym występuje przynajmniej jedna operacja warunkowa (kolejność wykonywanych operacji jest zależna od danych wejściowych).



Rysunek 4.8. Przykład algorytmu warunkowego

Algorytmy z pętlą (cykliczne) to algorytmy, w których występuje cykliczne powtarzanie pewnego ciągu operacji (stosują instrukcje pętli).



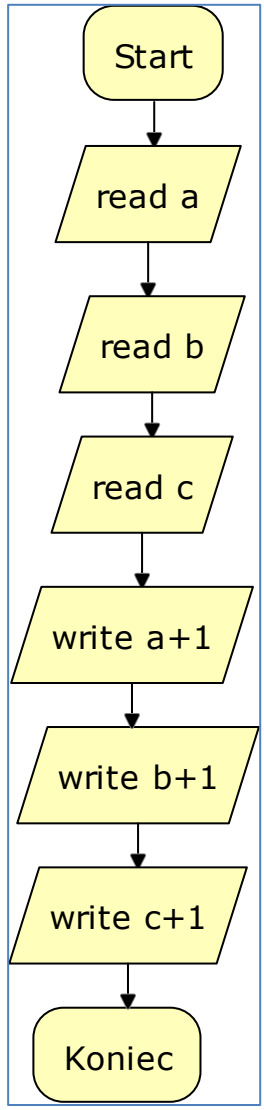
Rysunek 4.9. Przykład algorytmu z pętlą

4.3 Klasyfikacja ze względu na sposób wykonywania operacji

Algorytmy można podzielić ze względu na sposób wykonywania operacji i są to:

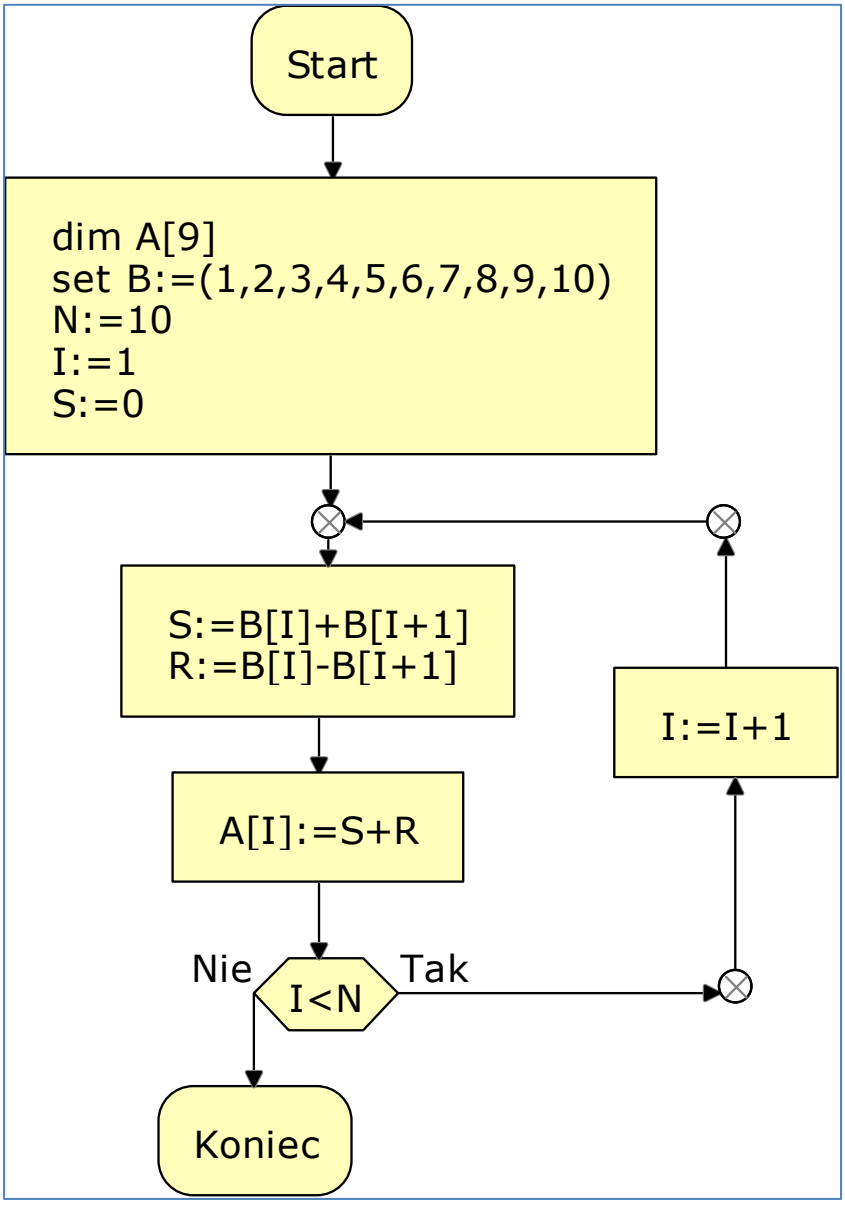
- algorytmy sekwencyjne,
- algorytmy iteracyjne,
- algorytmy rekurencyjne.

Algorytmy sekwencyjne to algorytmy, w których operacje są wykonywane w takiej kolejności, jak zostały napisane.



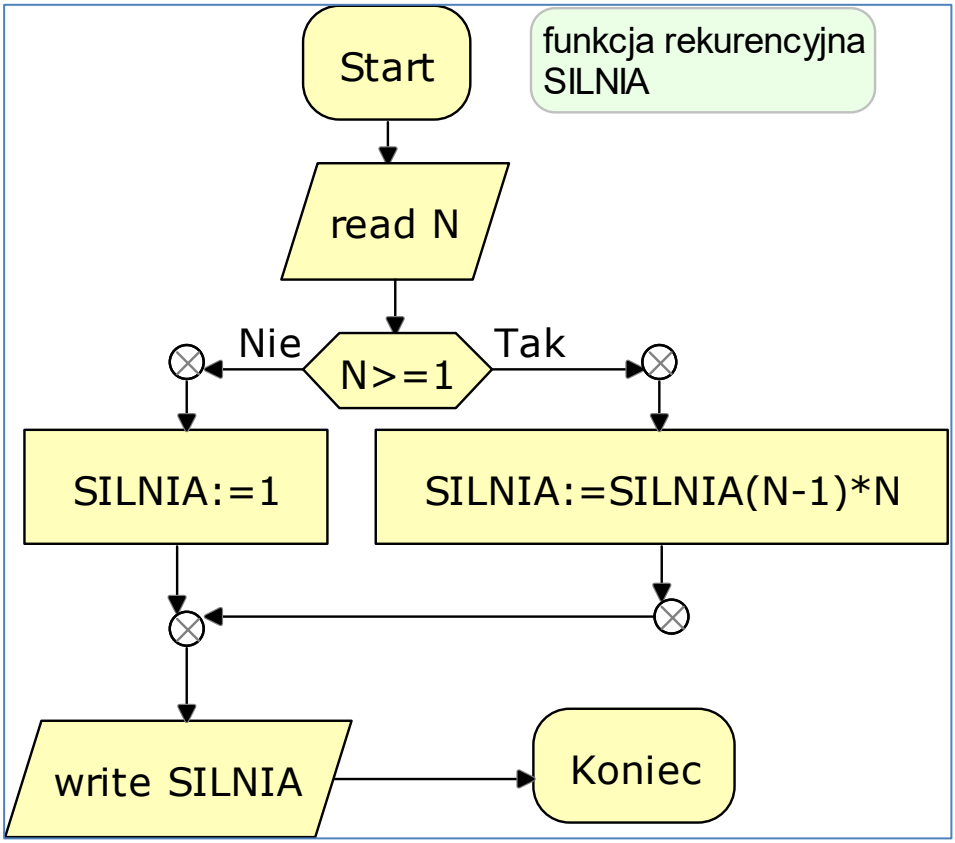
Rysunek 4.10. Przykład algorytmu sekwencyjnego

Algorytmy iteracyjne to algorytmy, w których jest powtarzanie pewnego ciągu operacji aż do spełnienia pewnego warunku.



Rysunek 4.11. Przykład algorytmu iteracyjnego

Algorytmy rekurencyjne to algorytmy, w których stosuje się funkcję odwołująca się do niej samej aż do spełnienia warunku zakończenia rekurencji.



Rysunek 4.12. Przykład algorytmu rekurencyjnego

```
wczytaj(n)
jeśli n = 0
        s <- 1
w przeciwnym razie
        dopóki n>=1 wykonuj
                s <- silnia(n-1)*n
wypisz(s)
```

Rysunek 4.13. Przykład algorytmu rekurencyjnego (pseudokod)

4.4 Klasyfikacja ze względu na obszar zastosowań

Algorytmy można podzielić ze względu na ich zastosowanie i są to:

- algorytmy matematyczne,
- algorytmy przeszukujące,
- algorytmy porządkujące,
- algorytmy szyfrujące.

Algorytmy matematyczne to algorytmy, w których wykonuje się obliczenia numeryczne,

Algorytmy przeszukujące to algorytmy, które przeszukują zbiór danych w celu znalezienia określonego elementu.

Algorytmy porządkujące to algorytmy, które zmieniają pozycje elementów w zbiorze danych w celu ich uporządkowania.

Algorytmy szyfrujące to algorytmy które kodują dane, tak aby ich odczyt nie był możliwy bez znajomości klucza szyfrującego.

4.5 Klasyfikacja algorytmów przyjęta w książce

W tej książce podzielono algorytmy na następujące kategorie tematyczne:

- algorytmy liniowe,
- algorytmy arytmetyczne,
- algorytmy znakowo-tekstowe,
- algorytmy sortujące,
- algorytmy szyfrujące,

operacje plikowe,

- algorytmy zaawansowane.

4.6 Wzór opisu algorytmu zastosowany w książce

Opis każdego algorytmu w tym rozdziale składa się z następujących części:

- specyfikacja algorytmu w postaci tabeli, zawierającej: poziom maturalny, klasę algorytmu, nazwę katalogu z kodami źródłowymi C# oraz schematem w wykonanym w Magicznych Bloczkach, opis danych

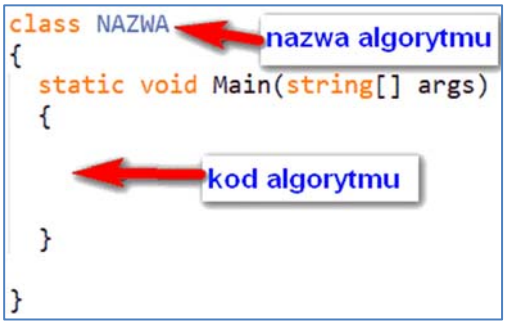
wejściowych, opis danych wyjściowych oraz ewentualnie dodatkowe założenia i uwagi.

Cel algorytmu	Np. Obliczenie średniej dla 3 liczb
Dane wejściowe	Np. a, b, c - liczby całkowite
Dane wyjściowe	Np. średnia - średnia arytmetyczna
Założenia i uwagi	Np. Brak
Nazwa katalogu źródłowego	Np. SUMOWANIE-CZTERECH-LICZB

Tabela 4.14. Wzór specyfikacji algorytmu

- przykładowa implementacja algorytmu:
  - Przykładowy schemat blokowy realizacji algorytmu, wykonany w programie Magiczne Bloczki, albo adekwatny pseudokod.
  - Przykładowy kod źródłowy realizacji algorytmu w języku C# (C Sharp) wykonany w środowisku Visual Studio.

Przyjęty ogólny schemat tworzenia kodów w C# w tej książce jest następujący:



Rysunek 4.15 Wzorzec programów w C#