



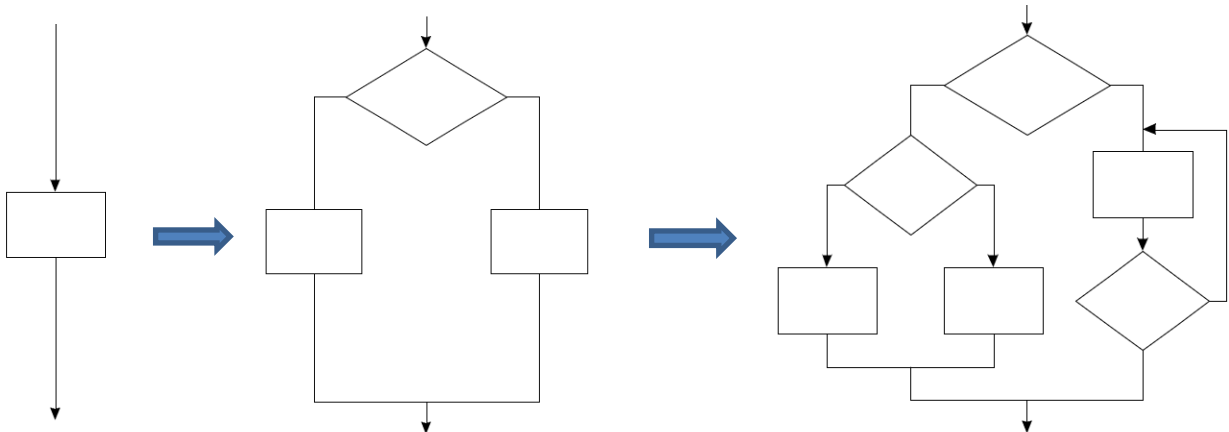
ALGORYTMY I STRUKTURY DANYCH

LABORATORIUM 1: ALGORYTMY WYSZUKIWANIA I SORTOWANIA

Programowanie strukturalne polega na budowaniu algorytmów (i programów) z użyciem tylko trzech konstrukcji:

- sekwencyjnej
- warunkowej
- iteracyjnej (ograniczonej i warunkowej)

instrukcja1 instrukcja2	if warunek: instrukcja1 else: instrukcja2	while warunek: instrukcja
x = 5 x = x + 10	if a % 2 == 0: flaga = True else: flaga = False	while i < 10: i+=1



Aplikacja do rysowania schematów blokowych: <https://app.diagrams.net/>

Zadanie 1. Narysuj schemat blokowy algorytmu sprawdzającego, czy podana przez użytkownika wartość występuje liście, a następnie napisz program w oparciu o ten schemat.

(Pliki do wykorzystania: [zadanie1.py](#))

Zadanie 2. Zaprojektuj algorytm wyszukiwania minimalnej wartości w liście.



Zadanie 2*. Zaprojektuj algorytm wyszukiwania minimalnej wartości w każdym wierszu listy dwuwymiarowej. Po znalezieniu minimalnej wartości wstaw ją na początek danego wiersza, zamieniając ją z wartością, która znajdowała się dotychczas na początku wiersza.

Zadanie 3. Wykonaj analizę działania algorytmu sortowania bąbelkowego na przykładowych danych.
(Pliki do wykorzystania: [Lab1_zadania.xlsx](#), arkusz [zadanie_3](#))

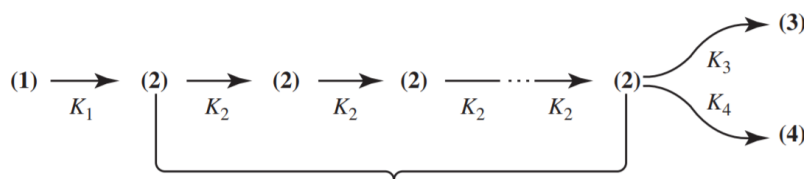
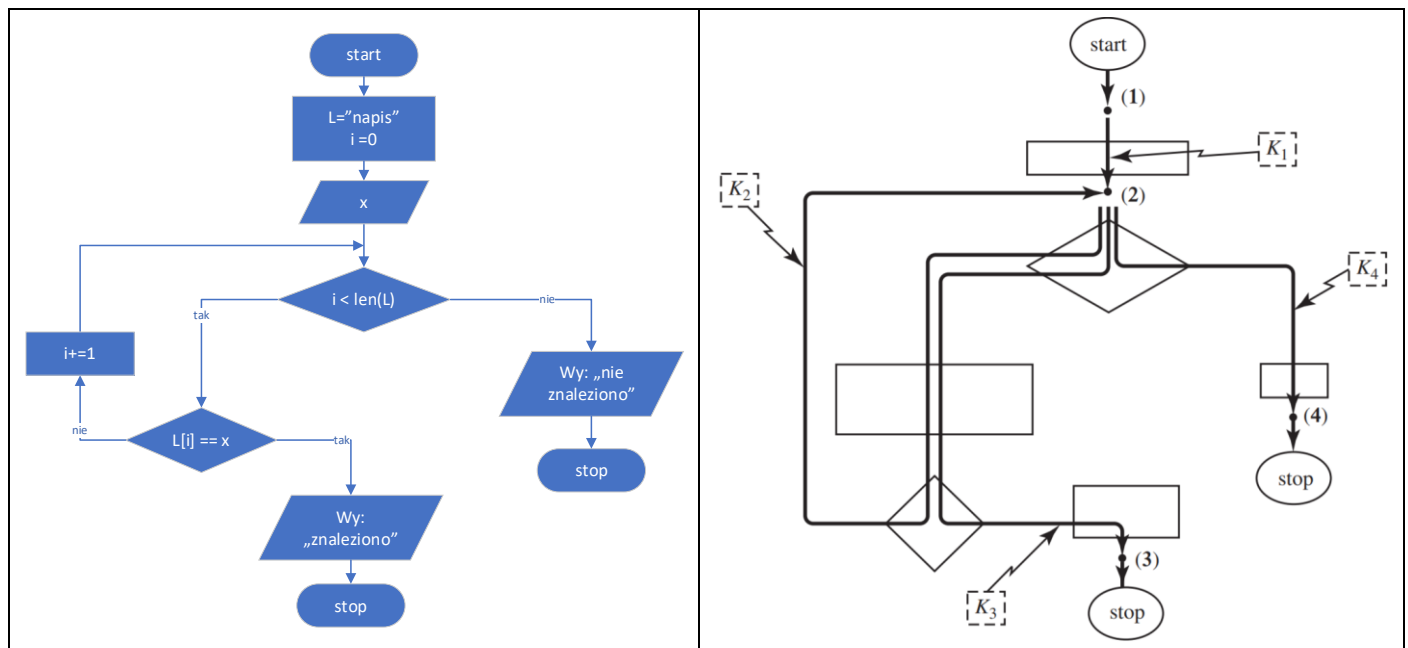
Dodatkowo*: Zaimplementuj algorytm sortowania bąbelkowego, który kończy działanie, jeśli w jednym przejściu całej tablicy nie dokonał żadnych zamian elementów.

(Pliki do wykorzystania: [zadanie3_bubbleSort.py](#))

ZŁOŻONOŚĆ OBLICZENIOWA

Złożoność obliczeniowa określa liczbę instrukcji wykonywaną przez program dla zadanego zbioru danych wejściowych.

Złożoność pamięciowa jest to liczba komórek pamięci potrzebnych do rozwiązania dowolnego zadania rozmiaru n .



D. Harel „Rzecz o istocie informatyki”

Na rysunku stałe K_1 do K_4 są związane z czterema możliwymi drogami. Zakłada się, iż każde wykonanie lokalnego odcinka obejmuje nie więcej niż K_i instrukcji.



Zadanie 4. Wykonaj analizę złożoności obliczeniowej algorytmu sortowania bąbelkowego.

(Pliki do wykorzystania: [bubble_sort_complexity.pdf](#))

Zadanie 5. Wykonaj analizę działania algorytmu sortowania przez wybór na przykładowych danych.

(Pliki do wykorzystania: [Lab1_zadania.xlsx](#), arkusz [zadanie_5](#))

Zadanie 6. Wykonaj analizę działania algorytmu sortowania przez wstawianie na przykładowych danych.

(Pliki do wykorzystania: [Lab1_zadania.xlsx](#), arkusz [zadanie_6](#))

Zadanie 7. Wykonaj analizę złożoności obliczeniowej algorytmu sortowania przez wstawianie.

(Pliki do wykorzystania: [insert_sort_complexity.pdf](#))

Zadanie 8*. Napisz program sortowania **liczb** naturalnych „leksykograficznie” względem rozwinięć dziesiętnych.

Przykład: Wejście: 1, 2, 3, 11, 21, 111, 231
 Wyjście: 1, 11, 111, 2, 21, 231, 3

Zadanie 9*. Napisz program sortowania wierszy macierzy „leksykograficznie”. Zastosuj algorytm sortowania przez wybór (selectionSort) lub przez wstawianie (insertSort).