Podstawy programowania (W11OPA-SI0072G)
Wstęp do programowania (W11FTE-SI0141W)
Wstęp do programowania (W11IKW-SI0080W)

Laboratorium 11

1. Zaimplementuj funkcję znajdującą najmniejszy element na liście. Funkcja powinna pobierać listę jako argument. Funkcja powinna zwracać wartość elementu najmniejszego. Aby zademonstrować działanie tej funkcji, napisz program, który wywołuje przygotowaną funkcję dla testowych danych (dane mogą być zapisane w programie lub mogą być podawane przez użytkownika).

2. Zaimplementuj funkcje działające na liście (np. zawierającej liczby całkowite):
   - wyznaczającą indeks najmniejszego elementu,
   - znajdującą największy element,
   - wyznaczającą indeks największego elementu.

   Zademonstruj działanie tych funkcji – napisz program, który wywołuje funkcje dla testowych danych.

3. Przygotuj plik sort.py (moduł) zawierający definicję funkcji z zadań 1 i 2. Napisz program – umieszczony w osobnym pliku, który wywołuje funkcje z modułu sort.

4. W pliku sort.py umieść i zdefiniuj funkcję **`swap`**, która w liście zamienia miejscami dwa elementy (o podanych indeksach).

5. Przeanalizuj algorytm sortowania przez wybór. Zauważ, że dysponujesz funkcjami:
   - znajdującą indeks elementu minimalnego w liście,
   - zamieniającą dwa elementy miejscami.

   Napisz funkcję **`selectsort`**, która porządkuje elementy listy, wykorzystując algorytm sortowania przez wybieranie. Zademonstruj działanie funkcji w programie.

6. Przeanalizuj algorytm sortowania bąbelkowego. Zaimplementuj funkcję **`bubblesort`**, która porządkuje elementy listy, wykorzystując ten algorytm. Zauważ, że dysponujesz funkcją zamieniającą miejscami dwa elementy tablicy. Zademonstruj działanie funkcji w programie.

7. Przeanalizuj algorytm sortowania przez scalanie. Zaimplementuj funkcję **`mergesort`**, która porządkuje elementy listy, wykorzystując ten algorytm. Zauważ, że przydatne będzie przygotowanie osobnej funkcji (**`merge`**) służącej do scalania dwóch posortowanych fragmentów listy. W funkcji **`merge`** konieczna będzie utworzenie kopii fragmentu listy. Zademonstruj działanie funkcji **`mergesort`** w programie.

8. Przeanalizuj algorytm sortowania szybkiego. Zaimplementuj funkcję **`quicksort`**, która porządkuje elementy listy, wykorzystując ten algorytm. Zauważ, że przydatne będzie przygotowanie osobnej funkcji (**`divide`**) służącej do podziału elementów listy na dwa fragmenty: elementów mniejszych oraz elementów większych/równych od wybranego elementu. W funkcji **`divide`** wykorzystaj funkcję **`swap`**. Zademonstruj działanie funkcji **`quicksort`** w programie.

Karol Tarnowski
Wrocław, 2025