AISDI 2024L
Projekt 1

Sortowanie

Maciej Bogusławski 331362
Hubert Kaczyński 331386

Opis projektu

W plikach bubble_sort.py, insert_sort.py, selection_sort.py, merge_sort.py znajdują się zaimplementowane przez nas wybrane algorytmy sortowania.

W pliku analize.py znajdują się funkcje służące do wczytywania słów z pliku tekstowego, inicjalizacji algorytmów sortowania, przeprowadzania pomiarów czasu działania algorytmów oraz zapisywania wyników do pliku.

W pliku create_plot.py znajdują się funkcje służące do rysowania wykresu zależności czasu sortowania od długości otrzymanych danych oraz zapisywania ich do pliku result.png.

W pliku main.py inicjalizowana jest analiza algorytmów sortowania.

W pliku test_sorts.py znajdują się testy sprawdzające poprawność działania algorytmów sortowania.

W folderze results w pliku result.png znajduje się wygenerowany wykres zależności czasu sortowania od długości danych.

W folderze results w pliku sorting_results.json znajdują się wyniki zawierające wartości pomiarów czasu działania algorytmów sortowania, nazwy wykonywanych algorytmów oraz długość sortowanych danych.

Opis środowiska

Implementacja i testowanie kodu przeprowadzone zostało, wykorzystując wersję 3.10.12 64-bit języka Python.

Projekt korzysta z następujących bibliotek:

- matplotlib
- gc
- time
- json
- copy

Procedura uruchomienia

W celu poprawnego działania projekt należy uruchomić poprzez plik main.py, znajdując się jednocześnie w katalogu Sortowanie.

Podział zadań

Zarówno na przestrzeni opracowania jak i testowania rozwiązania stale konsultowaliśmy się i współpracowaliśmy.

Maciej Bogusławski wykonał implementację plików merge_sort.py, analize.py i main.py, .gitignore.

Hubert Kaczyński wykonał implementację plików bubble_sort.py, selection_sort.py, insert_sort.py, test_sorts.py, create_plot.py.

Wspólnie konsultowaliśmy się w trakcie pracy nad projektem oraz konfigurowaliśmy repozytorium na Gitlab.

Podsumowanie wyników

Z tworzonych przez nasz projekt wykresów widać jasno efektywność poszczególnych algorytmów sortowania:

Bubble_sort: zdecydowanie najwolniejszy algorytm. Posiadał największy stosunek wzrostu czasu do wzrostu ilości sortowanych elementów. Dla każdej ilości elementów był wolniejszy od pozostałych metod.

Insert_sort i Selection_sort: dla obu algorytmów czas wykonania wzrastał porównywalnie szybko. Oba okazały się efektywniejsze od bubble sort oraz mniej efektywne od merge sort. Jednakże spośród ich dwóch to selection sort okazał się odrobinę efektywniejszy czasowo.

Merge_sort: dla każdej ilości elementów do posortowania (1 – 10000) czas wykonania utrzymywał się poniżej pięciu setnych sekundy. Merge sort okazał się zdecydowanie najefektywniejszy czasowo dla wszystkich ilości elementów powyżej kilkunastu. Przy bardzo małych ilościach (ok. 10) elementów Selection sort i Insert sort były porównywalnie efektywne