**Politechnika Warszawska**

**Algorytmy i struktury danych**

**Laboratorium 2**

**Projekt "Sortowanie"**

Informatyka – Inteligentne systemy

Paweł Sarnacki 305290

Piotr Niedziałek 304474

Prowadzący: dr inż. Łukasz Skonieczny

Warszawa 2023

1. **Wstęp**

Program został napisany w języku python z wykorzystaniem aplikacji Visual Studio Code. Program tworzy posortowane tablice o rozmiarze n, składające się z n wyrazów z wybranego pliku tekstowego. Dodatkowo tworzone są wykresy czasu sortowania w zależności od ilości elementów n dla każdego algorytmu sortującego. Program był pisany w wersji python 3.9.13, a wykorzystane biblioteki to: time, matplotlib.pyplot, unidecode i re.

1. **Struktura projektu**

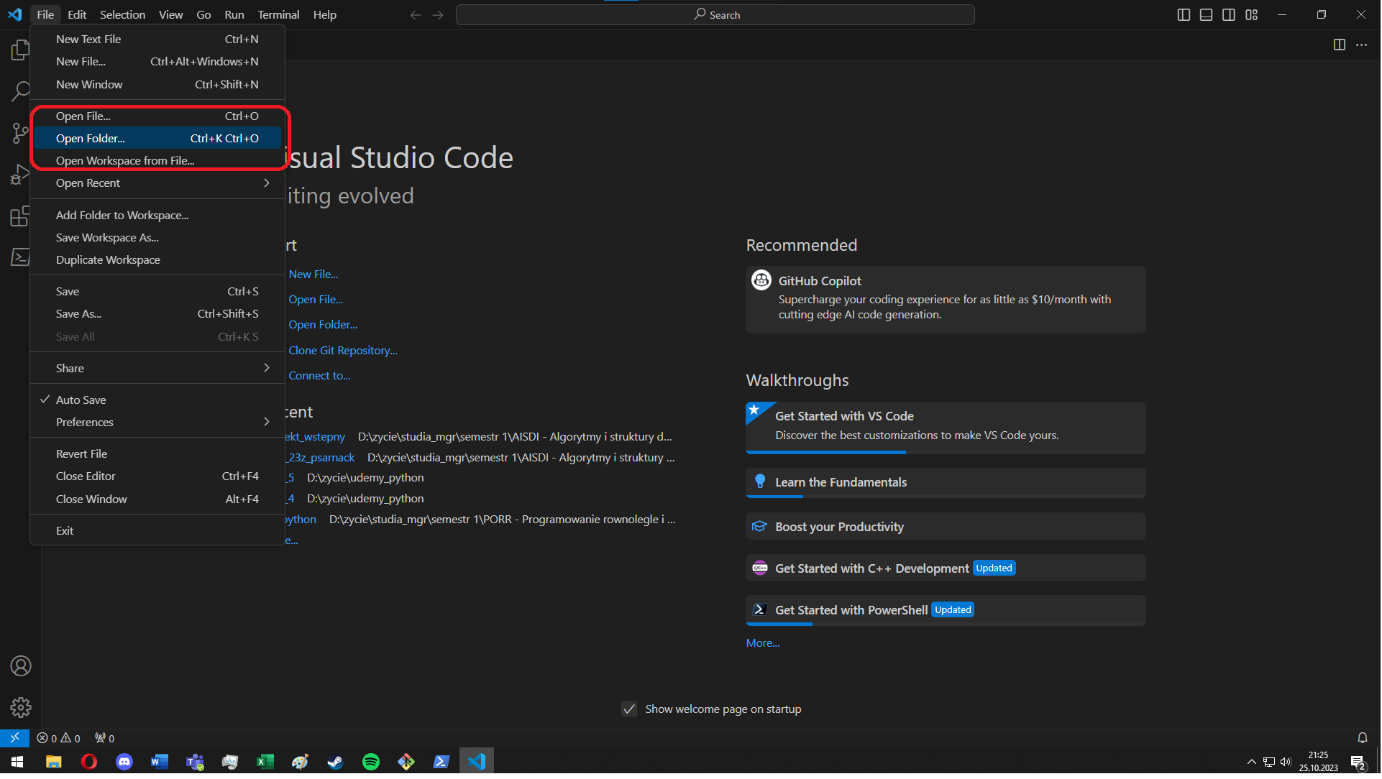
Projekt składa się z:

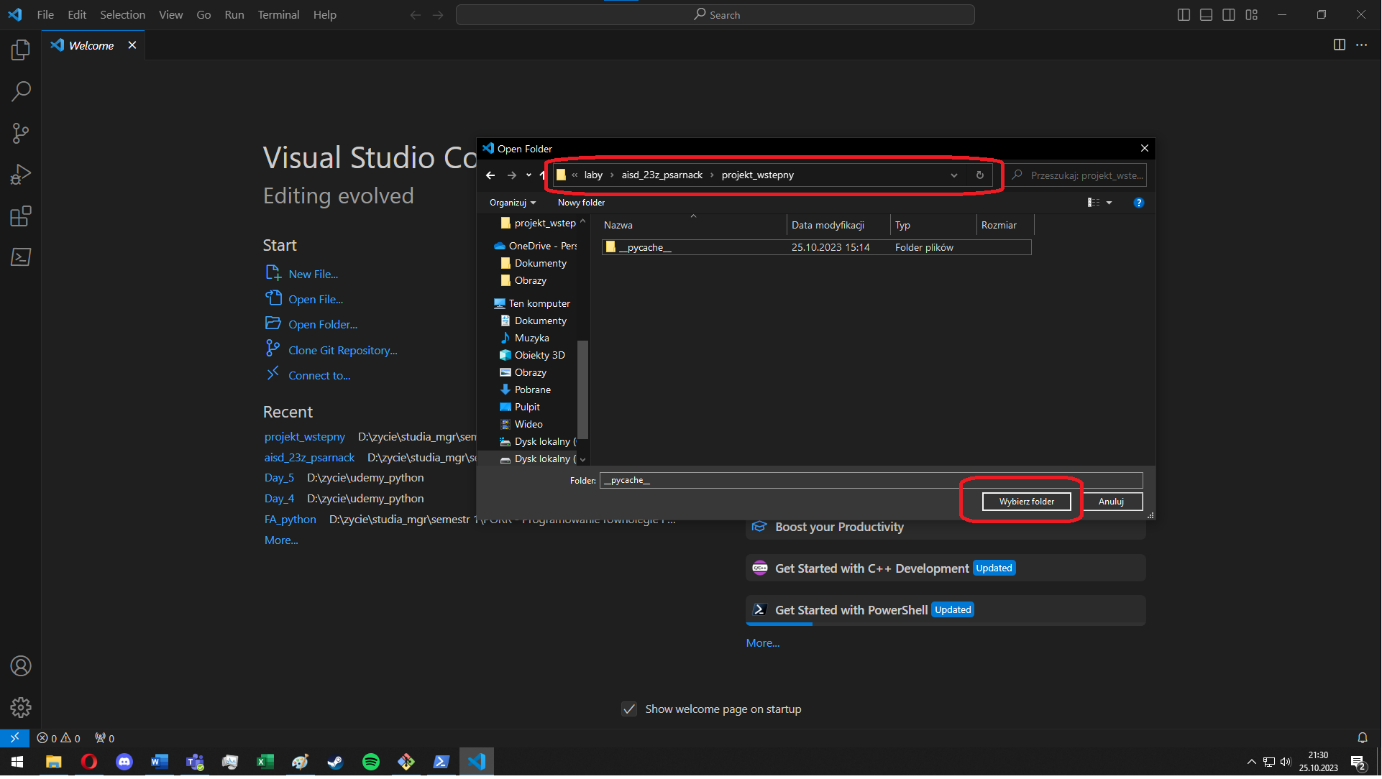
* Pliku main.py, który zawiera główny program obsługujący wszystkie algorytmy.
* Pliku insertion\_sort.py, który zawiera implementację algorytmu sortowania przez wstawianie.
* Pliku bubble\_sort.py, który zawiera implementację algorytmu sortowania bąbelkowego.
* Pliku merge\_sort.py, który zawiera implementację algorytmy sortowania przez scalanie.
* Pliku selection\_sort.py, który zawiera implementację algorytmu sortowania przez wybieranie.
* Pliku file.py, który zawiera implementację wczytywania n wyrazów i tworzenia listy.
* Pliku test.py, który zawiera implementację sprawdzania czy algorytmy posortowały wyrazy w ten sam sposób co wbudowana funkcja sorted().
* Pliku pan-tadeusz.txt, który zawiera w sobie tekst do sortowania.
* Katalogu plots, który zawiera wygenerowane wykresy.

A screenshot of a computer

Description automatically generated

1. **Uruchomienie projektu z środowiska Visual Studio Code**
   1. **Otworzenie folderu, wybór ścieżki i kliknięcie wybierz folder**

****

****

* 1. **Wybranie terminalu bash, wpisanie „py main.py”, lub kliknięcie F5, na dole zostanie wyświetlony wynik**

**Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie**

1. **Uruchomienie projektu bez środowiska Visual Studio Code**

Przejście do odpowiedniego folderu z plikami projektu i wpisanie w cmd **„py main.py”** **Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie**

1. **Otrzymane wyniki**
   1. **Bubble sort**

Obraz zawierający tekst, linia, diagram, zrzut ekranu

Opis wygenerowany automatycznie

* 1. **Insertion sort**

Obraz zawierający tekst, linia, diagram, Wykres

Opis wygenerowany automatycznie

* 1. **Merge sort**

Obraz zawierający tekst, diagram, Wykres, linia

Opis wygenerowany automatycznie

* 1. **Selection sort**

Obraz zawierający tekst, linia, diagram, Wykres

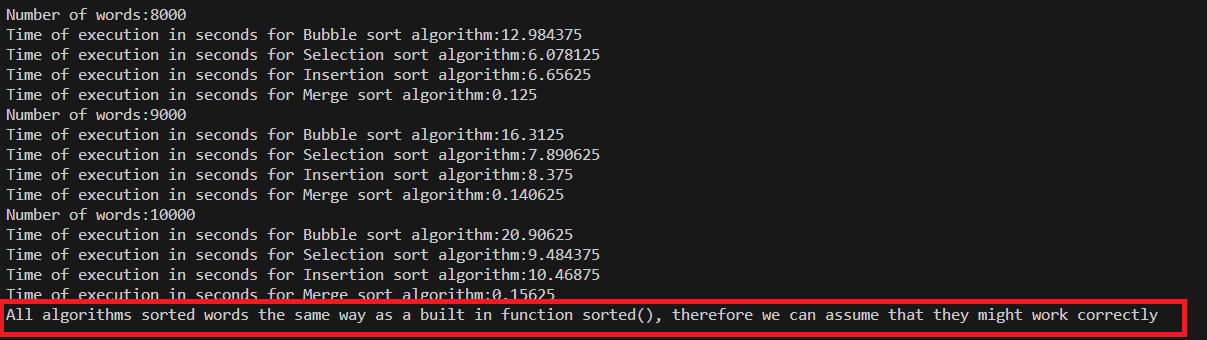
Opis wygenerowany automatycznie

* 1. **Wszystkie**

Obraz zawierający tekst, zrzut ekranu, linia, Wykres

Opis wygenerowany automatycznie

* 1. **Sprawdzenie**



1. **Wnioski**

- Najlepsze wyniki uzyskał algorytm sortowania przez scalanie (merge sort) Takiego rezultatu można było się spodziewać, ponieważ jego złożoność w najgorszym wypadku wynosi O(nlog(n)), natomiast pozostałych wynosi O(n2).

- Najgorsze wyniki uzyskał algorytm bąbelkowy.

- Wyniki dla algorytmu przez wstawianie i wybieranie były zbliżone do siebie.

- Zaimplementowane algorytmy działały poprawnie, to znaczy zwracały taką samą listę słów co wbudowana funkcja sorted().