**Politechnika Warszawska**

**Algorytmy i struktury danych**

**Laboratorium 2**

**Projekt "Sortowanie"**

Informatyka – Inteligentne systemy

Paweł Sarnacki 305290

Piotr Niedziałek

Prowadzący: dr inż. Łukasz Skonieczny

Warszawa 2023

1. **Wstęp**

Program zostały napisany w języku python z wykorzystaniem aplikacji Visual Studio Code. Program tworzy posortowane tablice o rozmiarze n, składającej się z n wyrazów z wybranego pliku tekstowego. Dodatkowo tworzony wykresy dla każdego algorytmu sortującego. Program był pisany na wersji python 3.9.13, a wykorzystane biblioteki to: time, matplotlib.pyplot.

1. **Struktura projektu**

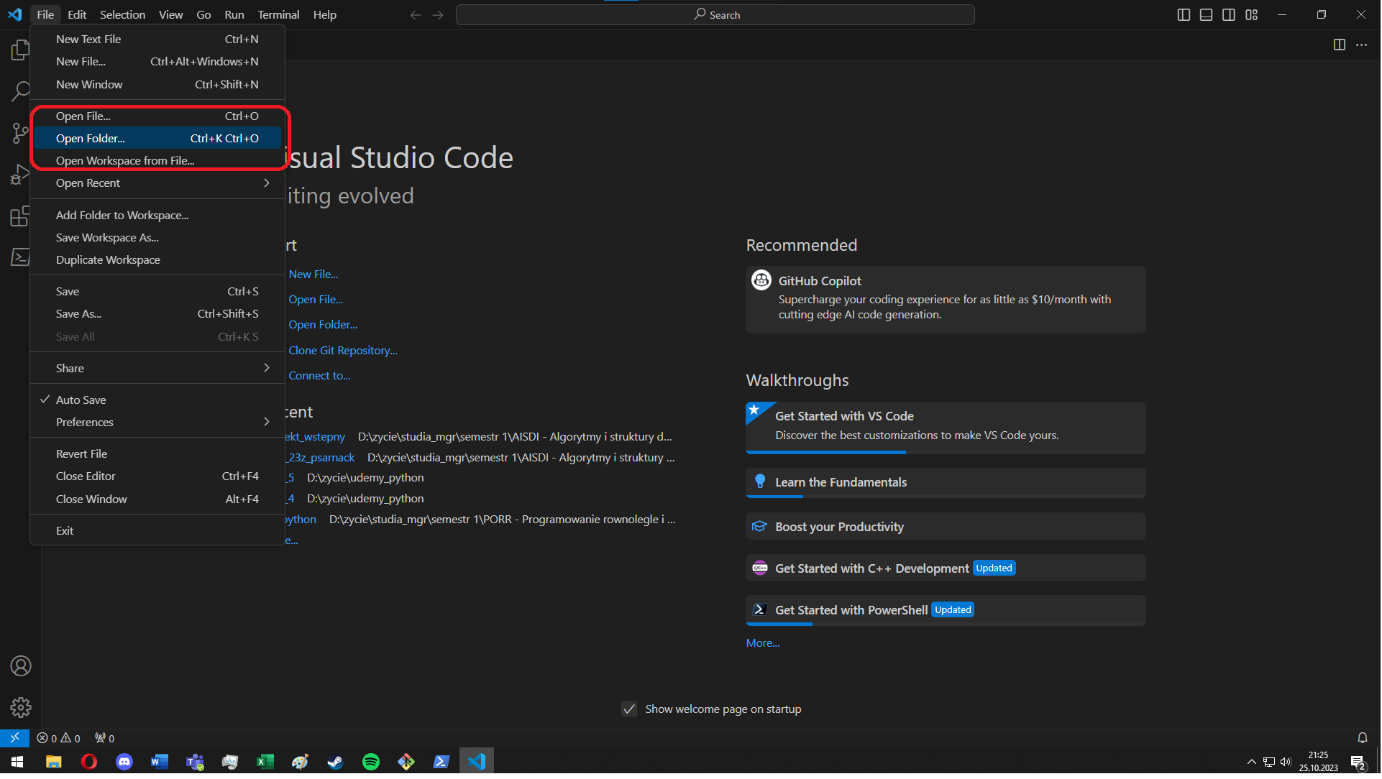
Projekt składał się z:

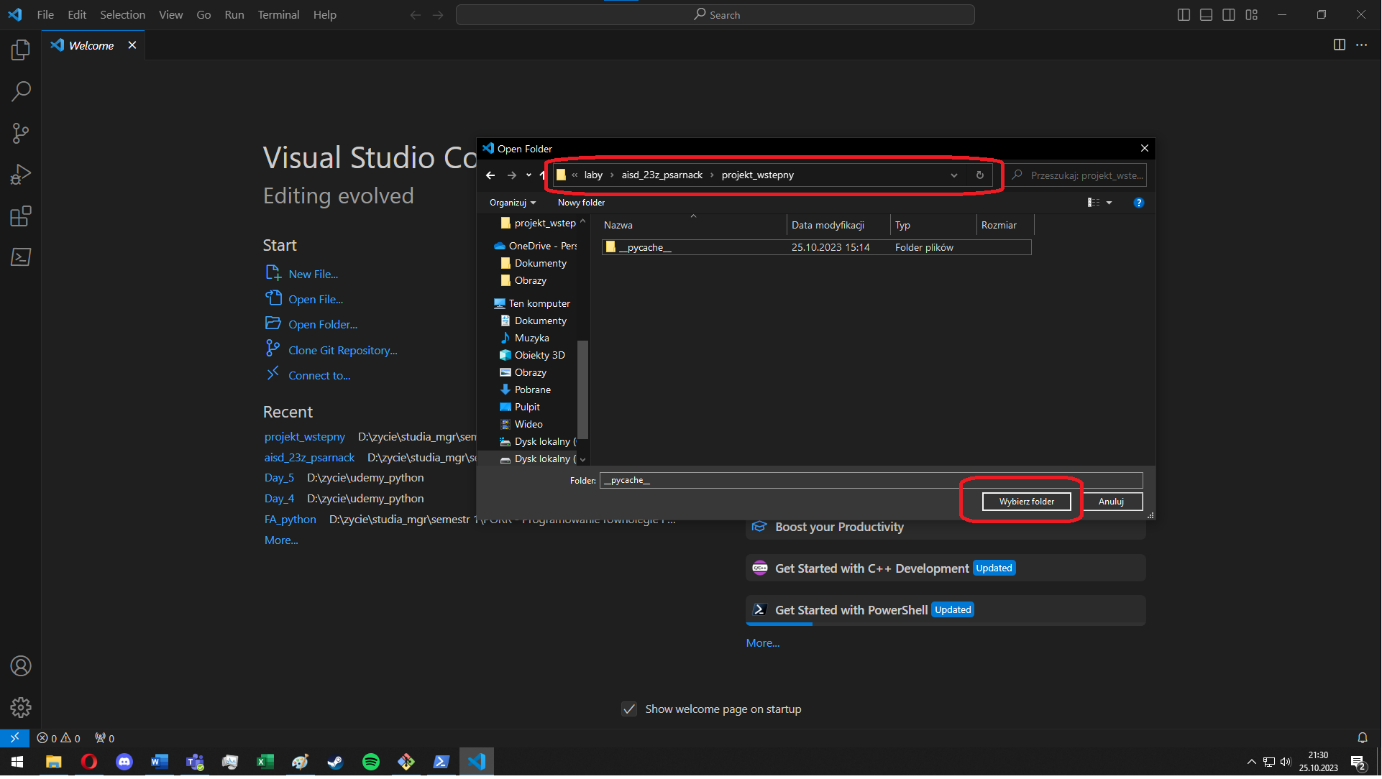
* Pliku main.py, który zawierał główny program obsługujący wszystkie algorytmy.
* Pliku sorty.py, który zawierał implementację czterech wybranych algorytmów.
* Pliku file.py, który zawierał implementację do wczytania n wyrazów i stworzenia tablicy.
* Pliku test.py, który zawierał implementację do sprawdzania czy algorytmy posortowały wyrazy w ten sam sposób co wbudowywana funkcja sorted().
* Pliku pan-tadeusz.txt, który zawierał w sobie tekst do sortowania.
* Katalogu plots, który zawierał wygenerowane wykresy.

Obraz zawierający tekst, zrzut ekranu, Czcionka, design

Opis wygenerowany automatycznie

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

****

****

* 1. **Wybranie termianalu 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

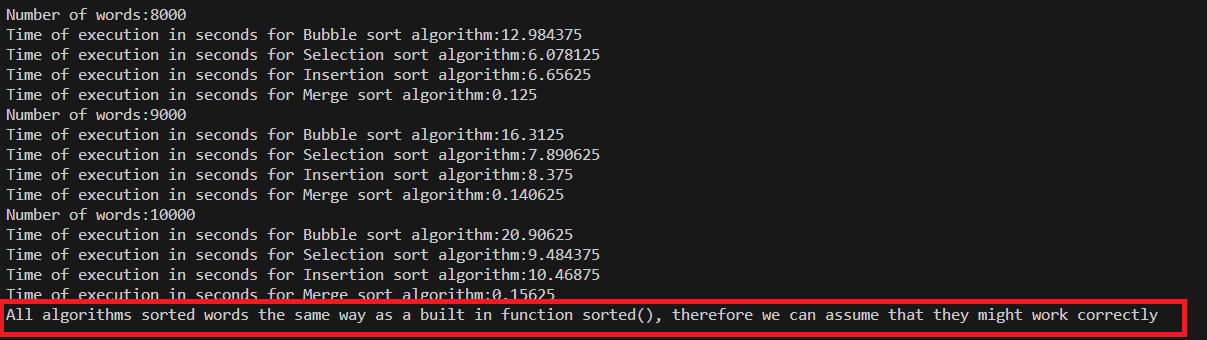
Opis wygenerowany automatycznie

* 1. **Wszystkie**

Obraz zawierający tekst, zrzut ekranu, linia, Wykres

Opis wygenerowany automatycznie

* 1. **Sprawdzenie**



1. **Wnioski**

- Najlepsze wyniki uzyskał algorytm przez scalanie, takiego wyniku można było się spodziewać ponieważ jego średnia złożoność wynosi O(nlog(n)), kiedy 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().