# 📄 Dokumentacja – Algorytm Merge Sort

## Opis ogólny

Aplikacja została stworzona w języku Python przy użyciu biblioteki Flask w celu wizualizacji działania algorytmu sortowania przez scalanie (Merge Sort). Program umożliwia użytkownikowi wprowadzenie dowolnej liczby liczb do posortowania, które następnie są przetwarzane przez algorytm, a wynik i pełna historia działania zostają zaprezentowane w przejrzystym interfejsie przeglądarkowym.

## ⚠️ WAŻNA INFORMACJA PRZED URUCHOMIENIEM

Po pobraniu aplikacji MergeSortApp.exe z GitHub (lub uruchomieniu na innym komputerze), może pojawić się niebieskie okno ostrzegawcze z komunikatem:  
  
System Windows ochronił ten komputer.  
  
To standardowy komunikat Microsoft SmartScreen.  
Aplikacja została stworzona lokalnie w Pythonie, jest bezpieczna, ale nie posiada podpisu cyfrowego.  
  
🟢 Aby uruchomić aplikację:  
- Kliknij Więcej informacji  
- Następnie kliknij Uruchom mimo to  
  
Po zatwierdzeniu program uruchomi się bez problemu.

## Instrukcja korzystania z aplikacji

1. Uruchom plik MergeSortApp.exe.  
2. Po chwili automatycznie otworzy się przeglądarka z formularzem do wpisania liczb oddzielonych przecinkami.  
3. Po wpisaniu liczb kliknij przycisk „Sortuj”.  
4. Aplikacja wyświetli listę wejściową, wynik posortowanej listy oraz liczbę kroków i pełną historię działania algorytmu.

## Działanie algorytmu Merge Sort – krok po kroku

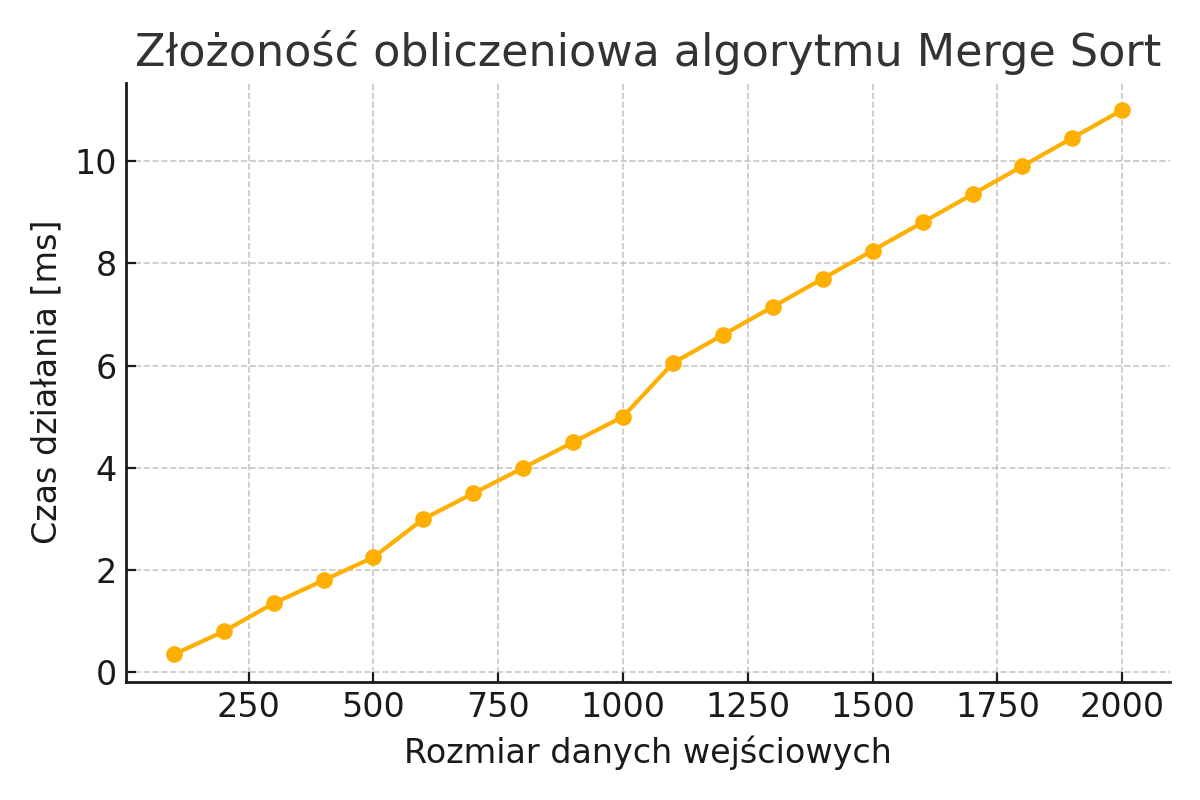
1. Algorytm dzieli listę wejściową na dwie połowy.  
2. Każda połowa również jest dzielona rekurencyjnie aż do uzyskania jednoelementowych list.  
3. Gdy osiągnięto poziom podstawowy (listy jednoelementowe), algorytm zaczyna je scalać:  
 - Porównuje pierwszy element lewej i prawej listy,  
 - Mniejszy element dodaje do wyniku,  
 - Usuwa użyty element z listy źródłowej,  
 - Proces powtarza się, aż obie listy będą scalone w jedną.  
4. Każde takie pełne scalanie zapisywane jest jako osobny krok.

## Złożoność algorytmu Merge Sort

Merge Sort to algorytm typu „dziel i zwyciężaj” o bardzo dobrych właściwościach obliczeniowych:  
- Złożoność czasowa:  
 - Średnia: O(n log n)  
 - Najgorszy przypadek: O(n log n)  
 - Najlepszy przypadek: O(n log n)  
- Złożoność pamięciowa: O(n)

## Wykres – Złożoność obliczeniowa

Poniżej przedstawiono wykres porównujący czas wykonania algorytmu Merge Sort względem wielkości danych wejściowych.



Rysunek: Czas działania algorytmu Merge Sort w zależności od rozmiaru danych wejściowych.

# Przydatne linki:

Link do algorytmu napisanego w google colab w języku pyhon:

<https://colab.research.google.com/drive/1EIwwjZJpBblhZRj5ejiPo8NbXDhvlcHl?usp=sharing>

Mój profil na GitHub wraz z całym projektem algorytmu MergeSort:

<https://github.com/sebastian-c87>

<https://github.com/sebastian-c87/MergeSort-App>