

AISDI - ćwiczenie 1

17 marca 2026

1 Sprawy organizacyjne

1. Ćwiczenie realizowane jest samodzielnie.
2. Ćwiczenie wykonywane jest w języku Python.
3. Ćwiczenie powinno zostać oddane najpóźniej do 08.04.2026. W ramach oddawania ćwiczenia należy zademonstrować prowadzącemu działanie kodu oraz utworzyć pull request (z kodem oraz raportem) który prowadzący będzie mógł komentować.
4. Rozwiązanie powinno wykorzystywać optymalne dla tego problemu struktury danych oraz algorytmy.
5. Kod oraz raport powinien być umieszczony na repozytorium **gitlab-stud.elka.pw.edu.pl**, a pull request utworzony do konta prowadzącego: **@djagodzi**
6. Raport powinien być w postaci pliku .pdf, .html albo być częścią noteboka jupyterowego. Powinien zawierać koncepcję rozwiązania i testowania, opis eksperymentów, uzyskane wyniki wraz z komentarzem oraz wnioski.
7. Na ocenę wpływa poprawność oraz jakość kodu i raportu.
8. Implementacja algorytmu powinna być ogólna. W szczególności powinna być możliwa do zastosowania dla dowolnego zbioru danych oraz dowolnej wielkości pamięci.

2 Ćwiczenie

Celem ćwiczenia jest implementacja algorytmu sortowania danych, które nie mieszczą się w pamięci operacyjnej. Dane znajdują się w dużym nieposortowanym pliku tekstowym. W pamięci RAM można jednocześnie przechowywać maksymalnie *MAX_MEMORY* elementów. Program może tworzyć dowolną liczbę plików tymczasowych.

- W żadnym momencie nie wolno załadować do pamięci więcej niż słów niż *MAX_MEMORY*.
- Nie wolno użyć `sorted()` ani żadnej funkcji sortującej na całym zbiorze danych naraz.

2.1 Dane

Celem ćwiczenia jest implementacja algorytmu, który posortuje alfabetycznie plik tekstowy zawierający zbiór 160 tysięcy słów z języka polskiego (jedno słowo w każdej linii): `input.txt`.

2.2 Program

Program powinien składać się z optymalnie zaimplementowanych części, które:

1. Wczytują dane z `input.txt`
2. Tworzą dowolną liczbę plików tymczasowych
3. Sortują wszystkie dane rosnąco (alfabetycznie)
4. Zapisują wynik do pliku `output.txt` (każde słowo w osobnym wierszu)
5. Czyszczą wszystkie pliki tymczasowe po zakończeniu

2.3 Badania

Po zaimplementowaniu algorytmu sortowania danych, które nie mieszczą się w pamięci operacyjnej oraz przetestowaniu poprawności działania, należy przeprowadzić badania wpływu maksymalnej wielkości pamięci *MAX_MEMORY* na czas wykonania sortowania pliku `input.txt`