

<b>Autor:</b> <b>Hubert Kowalczyk 259550</b>	<b>Struktury Danych i</b> <b>złożoność obliczeniowa</b> <b>Semestr letni 2022/2023</b>	<b>Termin:</b> <b>Wtorek NP: 17:05</b>
<b>Prowadzący:</b> <b>Dr. Inż. Tomasz Kapłon</b>	<b>Ćwiczenie 1</b>	<b>Data wykonania</b> <b>ćwiczenia:</b> <b>14.03.2023</b>
		<b>Data oddania</b> <b>sprawozdania</b> <b>28.03.2023</b>

## 1. Cel ćwiczenia

Celem ćwiczenia było napisanie szkieletu programu, który zostanie użyty do realizacji kolejnych zadań. Głównymi funkcjami szkieletu programu był odczyt danych z pliku/plików oraz zapis wyników do pliku. W przypadku zadania pierwszego program miał odczytywać odpowiednie parametry z pliku sterującego „config.cfg”. Plik sterujący zawiera pola, które definiują: nazwę pliku wejściowego, nazwę pliku wyjściowego, nazwę badanego algorytmu, nazwę pliku podsumowującego oraz rozmiar instancji. Oprócz wyżej wymienionych wymagań program miał także wyświetlać informację o tym, że program jest w trakcie działania.

## 2. Realizacja ćwiczenia

Program odczytuje plik config.cfg za pomocą obiektu ifstream z biblioteki fstream. Program odczytuje plik i po natrafieniu na linie, która pasuje do wzorca zapisuje ją do odpowiedniego pola w strukturze config w programie. Program odczytuje następujące parametry:

1. Plik z danymi - Plik, w którym znajdują się zestaw danych badawczych. W przypadku pierwszego i drugiego zadania jest to zestaw liczb, które należy posortować.
2. Posortowane elementy – Jest to pole potrzebne do realizacji ćwiczenia drugiego. Jest to ścieżka do pliku w którym znajdują się posortowane liczby. Plik ten znajduje się w folderze z projektem a same posortowane liczby zapisywane są w nim w formacie .csv według klucza <nazwa\_algorytmu><wielkość\_instancji>.csv przykładowa nazwa pliku: „bubbleSort2137.csv”
3. Algorytm – Jest to pole które definiuje jaki algorytm ma zostać zastosowany.
4. Rozmiar danych – Jest to pole które definiuje wielkość instancji potrzebnej do realizacji badania.
5. Wyniki czasowe – To pole to ścieżka w którym zapisywane są wyniki badania algorytmu. Same wyniki zapisywane są w pliku „TimeAndIteration.csv” według klucza
6. <nazwa\_algorytmu> <wielkość\_instancji> <czas\_trwania\_algorytmu>ns <ilość\_iteracji> iteracji.  
Przykładowy wynik algorytmu to : BubbleSort 10000 771440200ns 50004997 iteracji.

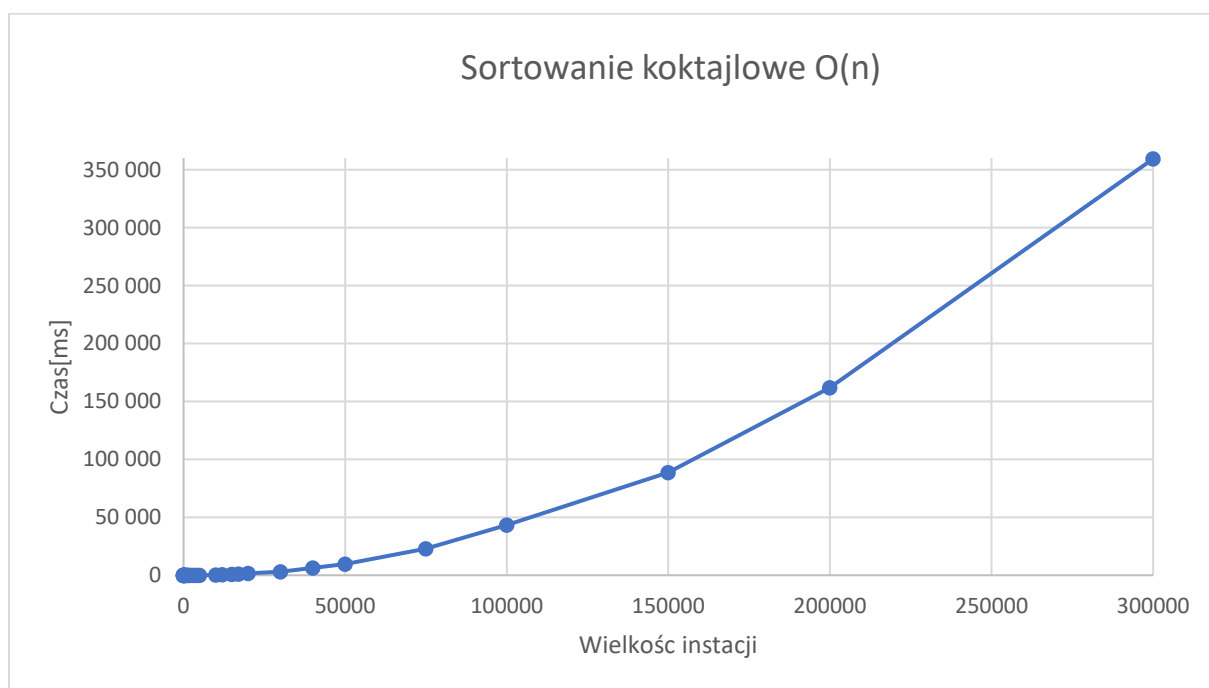
Program podczas realizacji przykładowego algorytmu wyświetla w konsoli wisielca. Po wykonaniu zadania program wypisuje w konsoli wyniki oraz zapisuje do pliku. W przypadku kiedy program nie zakończy zadania w ciągu 10 minut jest on przerywany a w konsoli zostaje wypisana odpowiednia informacja dla użytkownika. Zarówno wyświetlanie informacji o trwaniu algorytmu jak i zakończenie programu po upływie 10 minut realizowane są na osobnych wątkach za pomocą biblioteki thread

### 3.Realizacja przykładowego sortowania.

W ramach realizacji zadania pierwszego oprócz napisania szkieletu programu należało również zaimplementować oraz zbadać przykładowe sortowanie. W ramach realizacji ćwiczenia zastosowano sortowanie koktajlowe.

Sortowanie koktajlowe polega na obustronnym sortowaniu bąbelkowo zestawu danych.

Każda instancję której wielkości znajdują się w instrukcji powtórzono 10 razy oraz obliczono średnią. Następnie sporządzono wykres czasu wykonania w funkcji wielkości instancji. Wykres ten przedstawiono poniżej



Zgodnie z oczekiwaniami wykres notacji  $O$  przypomina parabolę.

### 4.Wnioski

W ramach realizacji ćwiczenia zaimplementowano sortowanie koktajlowe. Na podstawie wykresu wielkości czasu wykonania w funkcji wielkości instancji spostrzeżono, że wykres ten przypomina parabolę.

