(opcjonalnie)

Struktury danych i złożoność obliczeniowa

**Badanie Efektywności Algorytmów Sortowania**

• przez wybór

• przez wstawianie (liniowe, binarne)

• bąbelkowe

• przez scalanie

• szybkie

Pomiary byli wykonane dla takiej ilości elementów:10 000,40 000,80 000

Spszęt:

Procesor: Intel Core i5-8250U

Operacyjna pamięć: 8Gb

Dysk: SSD 256Gb

Program napisałem w C++

Używane biblioteki:



Algorym sortujący i funkcja sprawdzająca znajdują się w nagłówku <algorytmy.h>

* **Obliczenia czasu sortowania**

Tablica czasu sortowania: 10 000 elementów



Wykres czasu sortowania



Tablica czasu sortowania: 40 000 elementów



Wykres czasu sortowania



Tablica czasu sortowania: 80 000 elementów



Wykres czasu sortowania



* **Wykresy i opisy sposobów sortowania**

Wykresy liniowe czasu dla sortowania (10000,40000,80000)

Klasa czasowa dla sortowania:przez wybór,przez wstawianie i bąbelkowego)-**O(n2)**

**Sortowanie przez wybór - jedna z prostszych metod sortowania o złożoności O(n2). Polega na wyszukaniu elementu mającego się znaleźć na żądanej pozycji i zamianie miejscami z tym, który jest tam obecnie. Operacja jest wykonywana dla wszystkich indeksów sortowanej tablicy.**



**Sortowanie przez wstawinie– jeden z najprostszych algorytmów sortowania, którego zasada działania odzwierciedla sposób w jaki ludzie ustawiają karty – kolejne elementy wejściowe są ustawiane na odpowiednie miejsca docelowe. Jest efektywny dla niewielkiej liczby elementów, jego złożoność wynosi O(n2).**



**Sortowanie bąbelkowe – prosta metoda sortowania o małej złożoności czasowej i pamięciowej . Polega na porównywaniu dwóch sąsiadujących elementów i zamianie ich kolejności, jeżeli zaburza ona porządek (element po lewej stronie jest większy od elementu po prawej stronie lub odwrotnie (sortowanie rosnące)), w jakim się sortuje tablicę. Sortowanie kończy się, gdy podczas ostatniego przejścia nie dokonano żadnej zmiany.**



Klasa czasowa dla sortowania: szybkiego i przez scalanie-**O(n\*logn)**

**Sortowanie przez scalanie : Podział n-elementowy dzieli ciąg na dwa ciągi. Sortuje otrzymane ciągi, używając rekurencyjnego sortowania przez scalanie. Następnie łączy je w jeden posortowany ciąg.**



**Sortowanie szybkie: z tablicy wybiera się element rozdzielający, po czym tablica jest dzielona na dwa fragmenty: do początkowego przenoszone są wszystkie elementy nie większe od rozdzielającego, do końcowego wszystkie większe. Potem sortuje się osobno początkową i końcową część tablicy.**



* **Wnioski:**

**Wyżej wymienione metody sortowania pokazują złożoność i efektywność każdego sposobu. Czas sortowania zależy od metody i podzespołów komputera. Aby wyniki były rzetelne, wszystkie procesy i aplikacje w tle zostały zamknięte. Z wykresów widać,że najszybszym algorytmem sortującym jest szybkie sortowanie. Różnica między sortowaniem przez scalanie ,a sortowaniem szybkim byla nie widoczna przy 10 000 elementach, ale wraz ze wzrostem ilości liczb do posortowania, różnica staje bardziej zauważalna. Sortowanie przez wstawianie znajduje się w pomiędzy. Czwarte miejsce otrzymuje sortowanie przez wybór, zaś ostatnie przypada dla sortowania bąbelkowego.**