

Laboratorium AiSD

Lista 6

Efektywne algorytmy sortowania

Proszę pamiętać, że **część rozwiązania** zadania stanowi również **zestaw testów** zaimplementowanych algorytmów i/lub struktur danych. Dodatkowo, proszę zwracać uwagę na **powtarzające się fragmenty** kodu i wydzielać je do osobnych funkcji/klas.

1. Wykorzystując paczkę kodu z listy 5 zaimplementuj i przetestuj następujące warianty poznanych algorytmów:
 - a. **Sortowanie przez scalanie** iteracyjne z kolejką (Wykład 5 slajd 8),
 - b. **Sortowanie szybkie** zoptymalizowane pod kątem list dwiuzłazaniowych:
 - i. z wyborem pivota jako **pierwszego** elementu,
 - ii. z wyborem pivota jako **losowego** elementu,

Algorytmy przetestuj na wariantach kolekcji z poprzedniej listy. Sprawdź, czy implementacje są tak samo efektywne dla **list** i **tablic**. Przygotuj wykresy metryk jak dla listy 5.

Dla sortowania przez scalanie, jako klasę bazową użyć ***AbstractSortingAlgorithm<T>***.

Jako kolejkę wykorzystaj klasę *java.util.LinkedList<T>* (metody *addFirst*, *addLast*, *removeFirst*, *removeLast*).

Pamiętaj, że dostęp do elementu listy dwiuzłazaniowej ma złożoność **$O(n)$** !

Implementując sortowanie szybkie, metodę wyboru pivota warto przekazywać do konstruktora w postaci obiektu z metodą.

Aby stworzyć generator list dwiuzłazaniowych skorzystaj z klasy opakowującej ***LinkedListGenerator<T>***.