

Laboratorium AiSD

Lista 5

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 4 zaimplementuj i przetestuj następujące warianty poznanych algorytmów:
 - a. **Sortowanie przez scalanie** zoptymalizowane pod kątem tablic (patrz wykład 5 slajd 8, „bez stosów/kolejek (...)"),
 - b. **Sortowanie szybkie**:
 - i. z wyborem pivota jako **pierwszego** elementu,
 - ii. z wyborem pivota jako **losowego** elementu.

Dla sortowania szybkiego sposób wyboru **pivota** dostarczać poprzez **konstruktor** obiektu algorytmu. Zastosuj **wzorzec projektowy strategia**.

Algorytmy przetestuj na wariantach kolekcji z poprzedniej listy. Sprawdź, czy implementacja **sortowania przez scalanie** jest tak samo efektywna dla **list** i **tablic**. Przygotuj wykresy metryk jak dla listy 5.

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

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

Aby stworzyć generator list dwiżazaniowych skorzystaj z klasy opakowuywujacej ***LinkedListGenerator<T>***.