

II INFORMATYKA

Algorytmy i struktury danych

Lista nr 2

Zaproponuj "własny" algorytm rozwiązujący problem sortowania - zapisz ten algorytm w pseudokodzie. **Rozbuduj** program z Listy nr 1, uzupełniając go o funkcję działającą według tego algorytmu. Podobnie jak poprzednio, należy przewidzieć możliwość sortowania długich (np. 300 000 000 elementów) ciągów (o wartościach losowych i specyficznych - np. podawanych z klawiatury). Dla obu zaimplementowanych algorytmów należy obliczać t - liczbę wykonanych porównań pomiędzy elementami sortowanego ciągu (typu $a_i < a_j$) oraz wartość ilorazu $a = \frac{t}{n^2}$.

Program powinien dać możliwość racjonalnego sprawdzenia poprawności działania algorytmów poprzez wydrukowanie fragmentów tablicy (np. 50 początkowych elementów) przed i po sortowaniu.

Przetestuj działanie programu dla ciągów różnych długości. Sprawdzić doświadczalnie (sortując wiele ciągów losowych różnych długości), jaka jest średnia wartość czynnika a .

Uzupełnij program o pomiar rzeczywistego czasu działania obu algorytmów (w przeliczeniu na sekundy z dokładnością do drugiego miejsca po przecinku). Który z tych algorytmów (sortowanie przez wstawianie oraz własną metodą) jest szybszy? Odpowiedź uzasadnij porównując czasy działania obu algorytmów oraz liczbę wykonanych porównań pomiędzy elementami sortowanych ciągów. Czy istnieje jakiś związek pomiędzy wartościami czynnika a w obu algorytmach oraz różnicą w czasie ich działania?

Na podstawie wiedzy o czasie działania algorytmów (1 lub 2) dla ciągu długości 10000 o wartościach losowych, spróbuj przewidzieć czas działania tego algorytmu dla ciągów długości 100000 o wartościach losowych oraz dla ciągów o innych długościach.