

Algorytmy sortowania (uproszczone)

wszelkie prawa zastrzeżone
zakaz kopiowania, publikowania i przechowywania
all rights reserved
no copying, publishing or storing

Maciej Hojda

Uwaga: Słowa „dany”, „zadany”, „podany”, „wybrany” itd. w kontekście parametrów (zmiennych) oznacza parametr zadany przez użytkownika (a nie na stałe, przez programistę), a implementacja wykorzystująca taki parametr powinna obsługiwać jego różne wartości.

Uwaga: Zadania uzupełnij o tryb "krok-po-kroku"przechodzenia przez proces sortowania.

1 Zadanie nr 1 – podstawy sortowania

Zaimplementuj algorytm sortowania wektora liczb całkowitych względem kolejnych cyfr (od najbardziej znaczącej) tak, aby kolejne cyfry sortować raz malejąco, raz rosnąco (naprzemiennie). Dokładniej, posortuj tak, aby kolejne liczby o takim samej pierwszej cyfrze zostały dodatkowo posortowane względem drugiej cyfry, a liczby o takich samych: pierwszej i drugiej cyfrze zostały posortowane względem trzeciej cyfry, itd.

Zadanie wykonaj wielokrotnie korzystając z jednego (wybranego przez siebie) stabilnego algorytmu sortowania.

2 Zadanie nr 2 – ograniczony zbiór wartości

Dany jest wektor robotów (z poprzedniej listy). Wykorzystaj algorytm sortowania przez zliczanie do sortowania względem rozdzielczości kamery.

3 Zadanie nr 3 – dziel i zwyciężaj

- Zaimplementuj algorytm szybkiego sortowania (quicksort) wektora robotów (z poprzedniej listy) względem zadanego parametru.
- Proces sortowania przedstaw graficznie (np. korzystając z wykresu kolumnowego `matplotlib`) dla parametru zasięg. Wizualizacja ma wyświetlać kolejne kroki algorytmu (stan listy), gdzie kolejny krok zachodzi po wykonaniu elementarnej operacji (porównywania liczb, zmiany położenia liczb w ciągu, itd.).
- Wykonaj analogiczną wizualizację dla innego algorytmu sortowania. Obie wizualizacje przedstaw (wyświetl) równolegle.