Algorithms and Data Structures / List No. 4

Wroclaw University of Science and Technology - Faculty of Computer Science and Management
Marcin Stachowiak (marcin.stachowiak@pwr.edu.pl) - kurs powtórkowy

Keywords—podstawowe operacje niezbędne przy algorytmach sortujących, sortowanie przez wybieranie, sortowanie przez wstawianie

I. Zadanie 1 (Operacje pomocnicze)

Zaimplementuj dwie najczęściej używane operacje w algorytmach sortujących, tj. *less(...)*, która sprawdza, czy obiekty są identyczne oraz *exch(...)*, która zamienia podane dwa obiekty w liście miejscami [1] [2]. Zaimplementowane operacje za pomocą testów jednostkowych.

- Obie operacje zostały już zdefiniowane w klasie *com.asid.algorithms.sorting.AbstractSortService*. Należy tylko zaimplementować ciała metod.
- Należy zauważyć, że operacja *exch(...)* występuje w dwóch wariantach. W pierwszej z nich należy podać listę i dokładny numery pozycji zamienianych obiektów, natomiast w drugiej wymagane jest podanie listy i bezpośrednio zamienianych obiektów.
- Testy jednostkowe (po jednym dla każdej metody) należy napisać w klasie *com.asid.algorithms.sorting.SelectionSortServiceTest*

II. Zadanie 2 (Sortowanie przez wybieranie)

Zaimplementuj algorytm **sortowania przez wybieranie** [1] [2]. Przetestuj zaimplementowaną funkcjonalność za pomocą testów jednostkowych.

- Jako argumenty wejściowe algorytm powinien przyjmować listę dowolnych obiektów do posortowania oraz zdefiniowany komparator determinujący w jaki sposób obiekty powinny być ze sobą porównywane.
- Algorytm należy zaimplementować w metodzie *com.asid.algorithms.sorting.SelectionSortService#sort*.
- Po zakończeniu obliczeń metoda *sort(...)* powinna zwrócić obiekt *SortResultDs*, który będzie zawierał posortowaną listę obiektów (przekazaną w argumencie konstruktora serwisu) oraz czas, w jakim działał algorytm (w milisekundach). Obiekt *SortResultDs* został już zaimplementowany.
- Test(y) jednostkowe należy umieścić w stworzonej już klasie *com.asid.algorithms.sorting.SelectionSortServiceTest*.

III. Zadanie 3 (Sortowanie przez wstawianie)

Zaimplementuj algorytm **sortowania przez wstawianie** [1] [2]. Przetestuj zaimplementowaną funkcjonalność za pomocą testów jednostkowych.

- Jako argumenty wejściowe algorytm powinien przyjmować listę dowolnych obiektów do posortowania oraz zdefiniowany komparator determinujący w jaki sposób obiekty powinny być ze sobą porównywane.
- Algorytm należy zaimplementować w metodzie *com.asid.algorithms.sorting.InsertionSortService#sort*.
- Po zakończeniu obliczeń metoda *sort(...)* powinna zwrócić obiekt *SortResultDs*, który będzie zawierał posortowaną rosnąco listę obiektów (przekazaną w argumencie konstruktora serwisu) oraz czas, w jakim działał algorytm (w milisekundach). Obiekt *SortResultDs* został już zaimplementowany.
- Test(y) jednostkowe należy umieścić w stworzonej już klasie *com.asid.algorithms.sorting.InsertionSortServiceTest*.

Wyniki przeprowadzonego badania powinny zostać wyświetlone na konsolę. Formatowanie tekstu jest dowolne, ale powinno być przejrzyste.

IV. Pytania

1) Sortowanie szybkie (posortować ciąg liczb całkowitoliczbowych).

Literatura

[1] Robert Sedgewick and Kevin Wayne. *Algorithms, 4th Edition. [in Polish: Algorytmy]*. 2011. 1
[2] James Ross Simon Harris. *Algorytmy. Od podstaw*. 2006. 1