SORTOWANIE POZYCYJNE | JAVA

WSTĘP

Sortowanie pozycyjne jest kolejnym algorytmem sortowaniem niekorzystającym z porównywania. Radix sort polega na kilkukrotnym przesortowaniu elementów względem ich kolejnych znaków. Możemy go zastosować między innymi dla wielocyfrowych liczb, jak również i słów. Aby posortować rosnąco liczby najprościej wystarczy wykonywać algorytm od najmniej do najbardziej znaczących cyfr.

IMPLEMENTACJA W JAVIE

W załączonym przeze mnie kodzie, napisanym w języku Java, znajduję się prosty program, który po uruchomieniu umożliwia komunikację z użytkownikiem. Program pozwala na własnoręczne wpisanie tablicy znaków całkowitoliczbowych (int) oraz wypisuje już posortowaną tablicę. Wszystko dzieje się za sprawą algorytmu sortowania przez wstawianie, którego implementacja znajduje się   
(w liniach 48-53 przesłanego pliku Main.java) pod postacią funkcji *radixSort*, która wygląda następująco:

1    void radixSort(int arr[]) {

2        int max = get\_max(arr);

3        for(int e = 1; max/e > 0; e\*=10) {

4            countingSort(arr, 10, e);

5        }

6     }

OPIS ZAIMPLEMENTOWANEGO ALGORYTMU

Działanie poszczególnych linijek widocznego powyższego kodu:

2 Do zmiennej max zostaje przypisana największa wartość w tablicy otrzymanej do posortowania. Wykorzystuję tutaj prostą, napisaną przeze mnie funkcję, która przechodzi po wszystkich elementach w tablicy i szuka maksimum.

3-5 Wchodzimy teraz do pętli która sortuje nam liczby po ich kolejnych cyfrach. Ilość przejść pętli jest równa maksymalnej ilości cyfr w jednej liczbie naszej tablicy. Wykorzystujemy tutaj wcześniej zainicjalizowaną zmienną max oraz zmienną e, która przyjmuje wartości równe kolejnym potęg liczby 10. W ciele pętli korzystamy z już napisanej funkcji *countingSort.* Wartość e odpowiada tutaj miejscu względem, którego będą posortowane liczby (przykładowo e równe 100 oznacza, że arr[] zostanie posortowane względem liczby setek)

WAŻNE CECHY ALGORYTMU

* radix sort ma złożoność czasową równą **O(d \* (n + b))**, gdzie d jest maksymalną liczbą cyfr w liczbie z tablicy, n jest długością tablicy, natomiast b jest bazą używanego systemu liczbowego (dla systemu dziesiętnego b wynosi 10)
* ma liniową złożoność czasową co powoduje, że dla dużych wartości (liczb o wielu cyfrach) jest on optymalniejszy niż algorytmy wykorzystujące porównania
* sortowanie przez zliczanie wykorzystujące pomocniczy algorytm sortowania, który jest stabilny (tak jak counting sort) jest również stabilne