Agnieszka Wiśniewska, nr albumu: 200466

Sprawozdanie z laboratorium nr 4
ALGORYTMY SORTUJĄCE

23.03.2014

1. Opis zadania

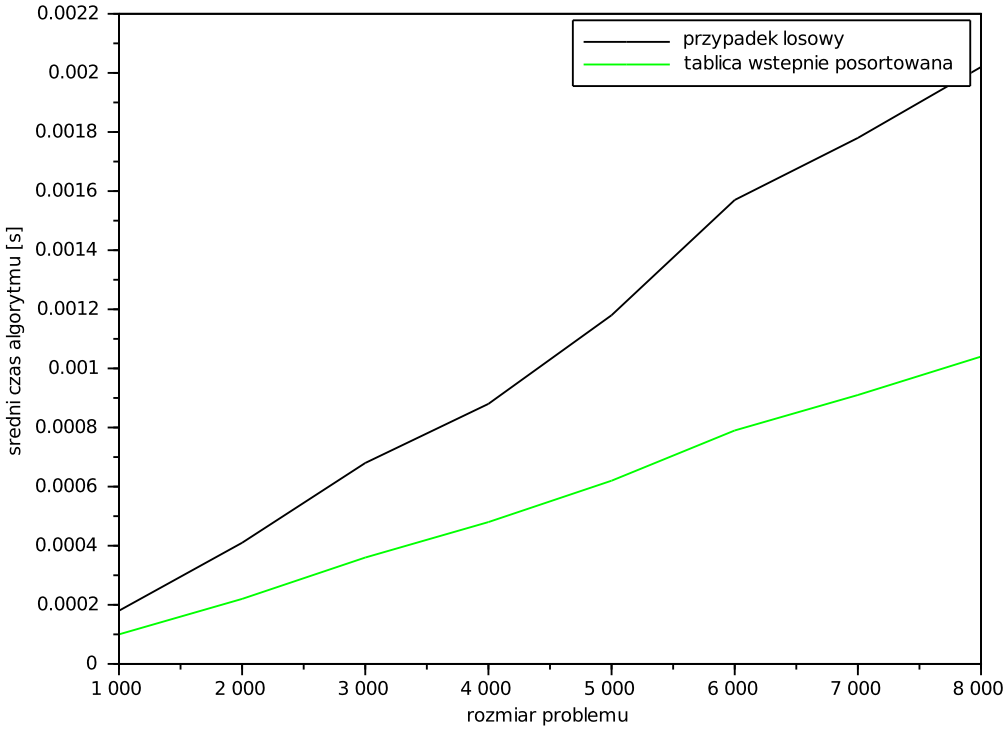
Zadaniem do wykonania było przygotowanie programu mierzącego czas wykonywania wybranych algorytmów sortowania. Wyniki jego działań można zobaczyć na wykresach w punkcie 2.

2. Wyniki działania programu — wykresy

2.1. Sortowanie szybkie (quicksort)

Jest to wydajny algorytm sortowania działający na zasadzie "Dziel i zwyciężaj- z tablicy wybiera się element rozdzielający, po czym jest ona dzielona na dwa fragmenty: do początkowego przenoszone są wszystkie elementy nie większe od rozdzielającego, do końcowego wszystkie większe. Następnie sortuje się osobno początkową i końcową część pierwszej tablicy. Rekurencja kończy się, gdy kolejny fragment uzyskany z podziału zawiera pojedynczy element, jako że jednoelementowa tablica nie wymaga sortowania.

Średnia czasowa złożoność obliczeniowa: O(nlogn)

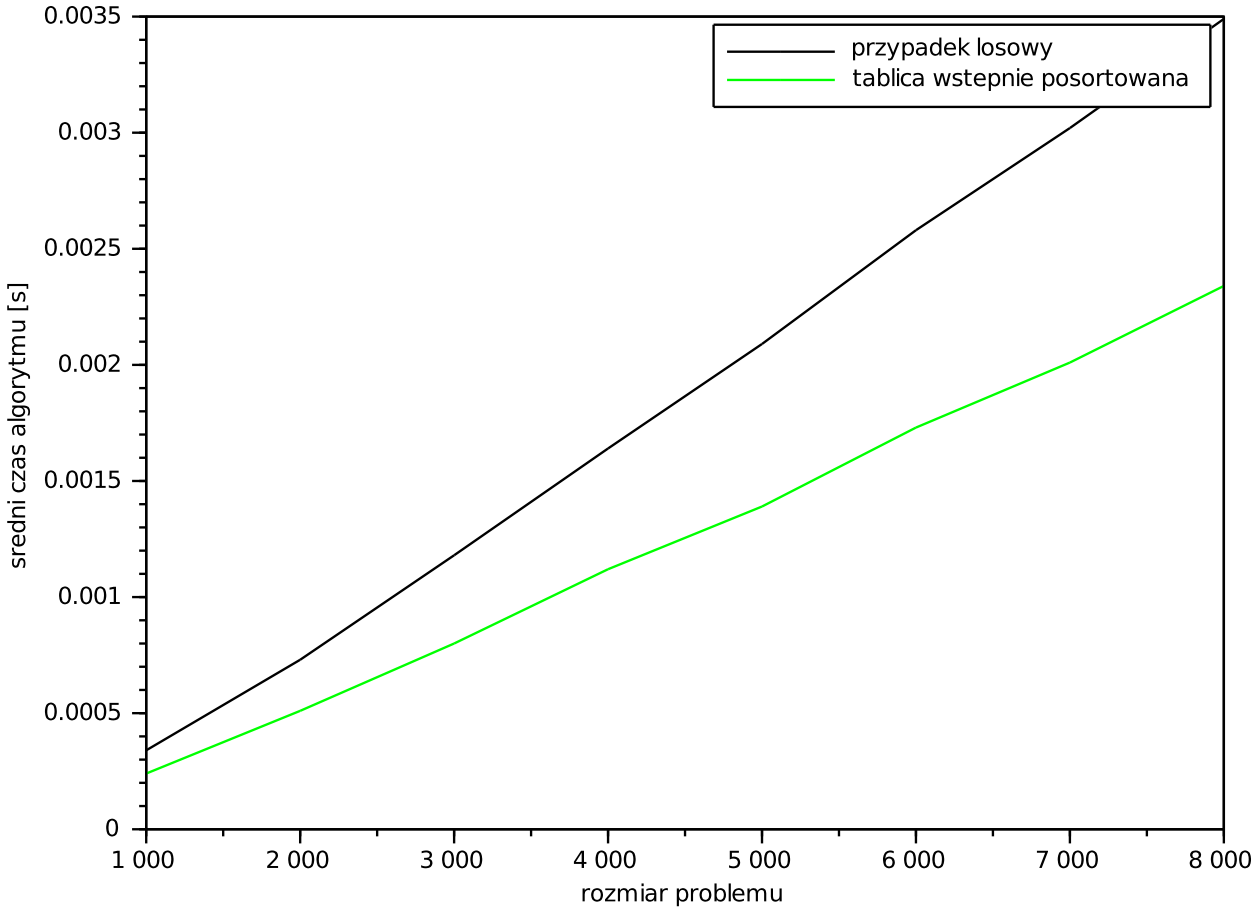


Rysunek 1. Zależność czasu wykonywania algorytmu od rozmiaru problemu dla sortowania szybkiego.

2.2. Sortowanie przez scalanie (mergesort)

Ten algorytm również działa na zasadzie "dziel i zwyciężaj". Na początku rozdziela się dane na dwie części, sortuje je przez scalanie (chyba, że pozostał już tylko jeden element), aby w końcu połączyć wszystkie podciągi w jeden posortowany ciąg.

Średnia czasowa złożoność obliczeniowa: O(nlogn)

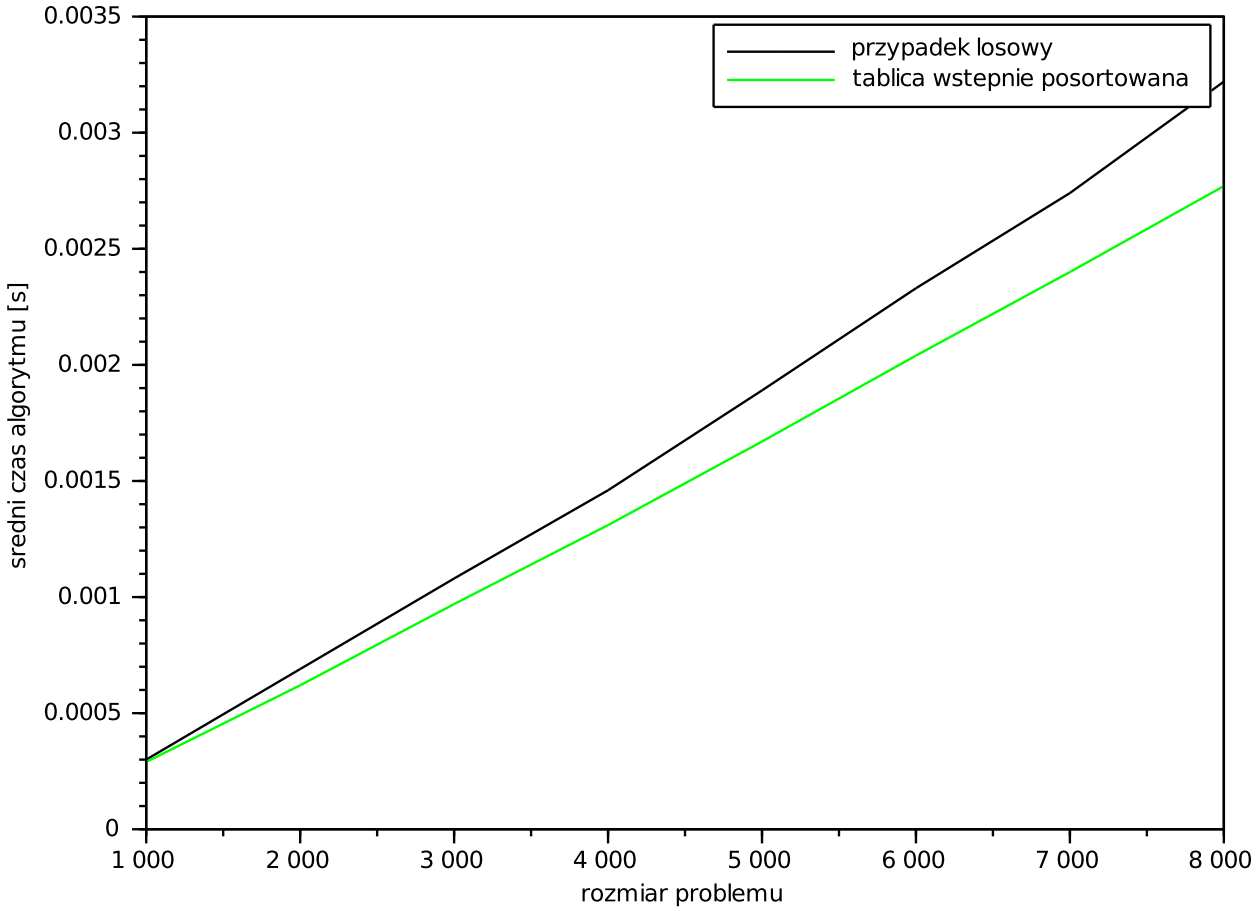


Rysunek 2. Zależność czasu wykonywania algorytmu od rozmiaru problemu dla sortowania przez scalanie.

2.3. Sortowanie przez kopcowanie (heapsort)

Niestabilny, lecz szybki i niepochłaniający dużo pamięci algorytm sortowania danych. Działa na podstawie użycia kolejki priorytetowej zaimplementowanej w postaci binarnego kopca zupełnego. Algorytm składa się z dwóch faz - w pierwszej fazie sortowane elementy są reorganizowane w celu utworzenia kopca, w drugiej dokonywane jest właściwe sortowanie.

Średnia czasowa złożoność obliczeniowa: O(nlogn)



Rysunek 3. Zależność czasu wykonywania algorytmu od rozmiaru problemu dla sortowania przez kopcowanie.

3. Wnioski

— Z powyższych wykresów wynika, że wszystkie trzy przetestowane algorytmy są wydajniejsze dla danych wstępnie posortowanych.
— Średnia czasowa złożoność obliczeniowa dla testowanych algorytmów wynosi O(nlogn)
— W najgorszym przypadku złożoność obliczeniowa algorytmu sortowania szybkiego wynosi O(n^2). Dzieje się tak, gdy np.: element rozdzielający (piwot) za każdym razem jest elementem podtablicy o skrajnej wartości (maksymalnej lub minimalnej) - w takim przypadku podtablica nie jest praktycznie dzielona, lecz 'odpada' od niej jedynie element osiowy, a resztę tablicy należy dzielić dalej.

4. Załączniki

— Dokumentacja z Doxygena "Dokumentacja.pdf"