

# Badanie złożoności obliczeniowej algorytmów sortowania szybkiego oraz przez scalanie

Szymon Leśniak

31 marca 2014

## 1 Wstęp

Sortowanie przez scalanie (*merge sort*) opiera się na rekurencyjnym rozłożeniu sortowanej tablicy aż do pojedynczych elementów. Następnie dochodzi do scalenia tablic, już w uporządkowany sposób. Algorytm zarówno w średnim, jak i w najgorszym przypadku ma złożoność  $O(n \log n)$ .

W algorytmie sortowania szybkiego (*quicksort*) wybiera się jeden z elementów tablicy (nazywany *pivot*), który jest umieszczany na właściwe jemu miejsce. Na powstałych po jego bokach podtablicach wywołuje się rekurencyjnie ten algorytm. W średnim przypadku jego złożoność wynosi  $O(n \log n)$ , jednak działa szybciej od innych algorytmów tego samego rzędu. Algorytm działa najwolniej, gdy pivotem każdorazowo jest największa lub najmniejsza liczba z tablicy. Wówczas mamy  $O(n^2)$ .

## 2 Opis badania

Sortowanym pojemnikiem użytym w badaniu był `vector` liczb typu `int`, pochodzący z STL C++. Implementacji dokonano przy użyciu systemu operacyjnego Ubuntu 13.10 oraz kompilatora g++ 4.8.1.

Sortowane tablice były wczytywane z pliku oraz porównywane ze wzorcem zawartym w drugim pliku. W plikach wykorzystanych do badania każdy element tablicy z osobna był losowany, co powinno zapewnić realizację przypadku średniego sortowania. Ostateczny wynik badania był średnią z czasu wielokrotnego sortowania tej samej tablicy.

## 3 Wyniki badania

Wyniki są przedstawione w poniżej tabeli oraz na wykresie (skala log-log)

Tablica 1: Wyniki działania algorytmów sortowania

Ilość liczb	Ilość powtórzeń	MergeSort	QuickSort
10	500	$2,51 \cdot 10^{-5}$	$1,84 \cdot 10^{-6}$
100	100	0,000165963	$1,36 \cdot 10^{-5}$
1000	50	0,00183497	0,000194129
10000	25	0,02109	0,00222775
100000	15	0,231004	0,0270007
1000000	10	2,54498	0,318073

Średni czas sortowania losowego ciągu liczb

