

# Sprawozdanie Quicksort

Mateusz Król 226400

April 21, 2017

## 1 Wstęp oraz krótki opis programu

Celem obecnych zajęć było zapoznanie się z algorytmem szybkiego sortowania.

W programie zastosowano książkowy algorytm quicksort. Program dostaje na wejście tablice oraz jej pierwszy oraz ostatni element. Potem wybierany jest pivot, w poniższym sprawozdaniu rozpatrzono 3 przypadki doboru pivotu: początek tablicy, środek tablicy oraz koniec tablicy. Potem przesuwamy pivotu zostawiając elementy mniejsze od pivotu po lewej a większe od pivotu po prawej stronie względem aktualnej tablicy i wybiera następne pivoty w kolejnych częściach tablic.

Do badań przyjęto 3 rodzaje ułożenia elementów w tablicach: losowo, rosnąco, malejąco.

We wnioskach zebrano wykresy złożoności obliczeniowej z naniesionymi liniami trendu

## 2 Uśrednione czasy dla 10 pomiarów, 3 przypadki wyboru pivotu

ilosc elementow/pivot	początek [ms]	środek[ms]	malejąco [ms]
100	0,0095	0,0077	0,083
1000	0,108	0,1075	0,1082
10000	1,3957	1,593	1,4542
100000	23,3531	18,086	18,836
1000000	185,5871	187,814	197,91
10000000	1822,827	1795,3	1919,865
100000000	196469,72	18787,86	199126,57

Table 1: Tablica nieposortowana

ilosc elementow/pivot	lpoczątek [ms]	środek [ms]	koniec [ms]
100	0,0161	0,002	0,0163
1000	1,9819	0,0282	0,18436
10000	168,9879	0,4161	206,3731
100000	16457,32	5,6332	16748,16
1000000	brak	72,065	brak
10000000	brak	759,4242	brak
100000000	brak	9460,442	brak

Table 2: Tablica posortowana rosnąco

ilosc elementow/pivot	początek [ms]	środek[ms]	koniec[ms]
100	0,0166	0,0023	0,0171
1000	1,62	0,0283	1,7406
10000	172,3316	0,5308	168,9556
100000	15593,73	6,8849	1578,17
1000000	brak	65,076	brak
10000000	brak	692,9536	brak
100000000	brak	7929,29	brak

Table 3: Tablica posortowana malejąco

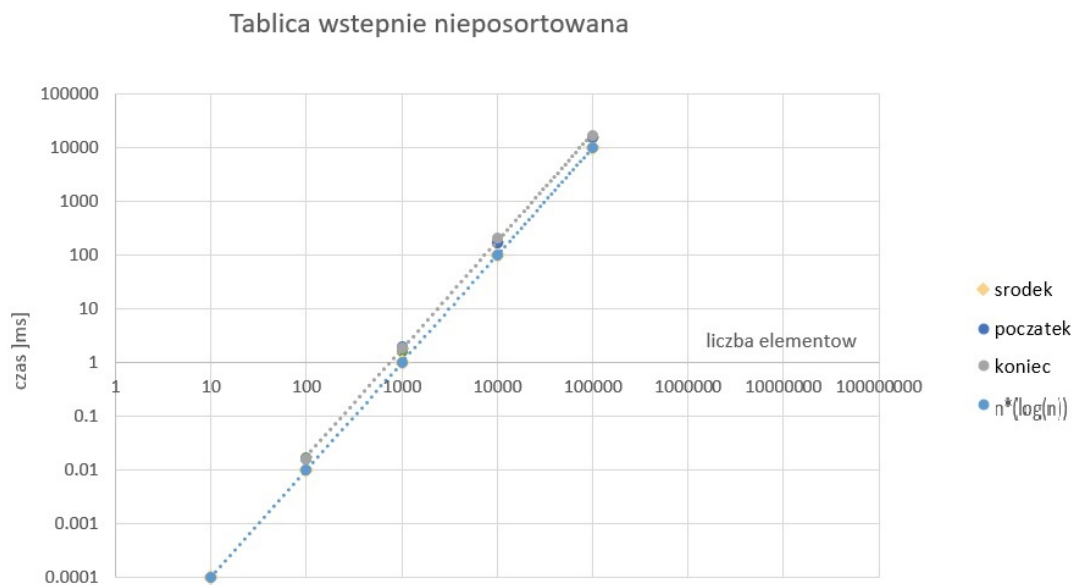


Figure 1: Działanie algorytmu przy wstępnie nieposortowanej tablicy.

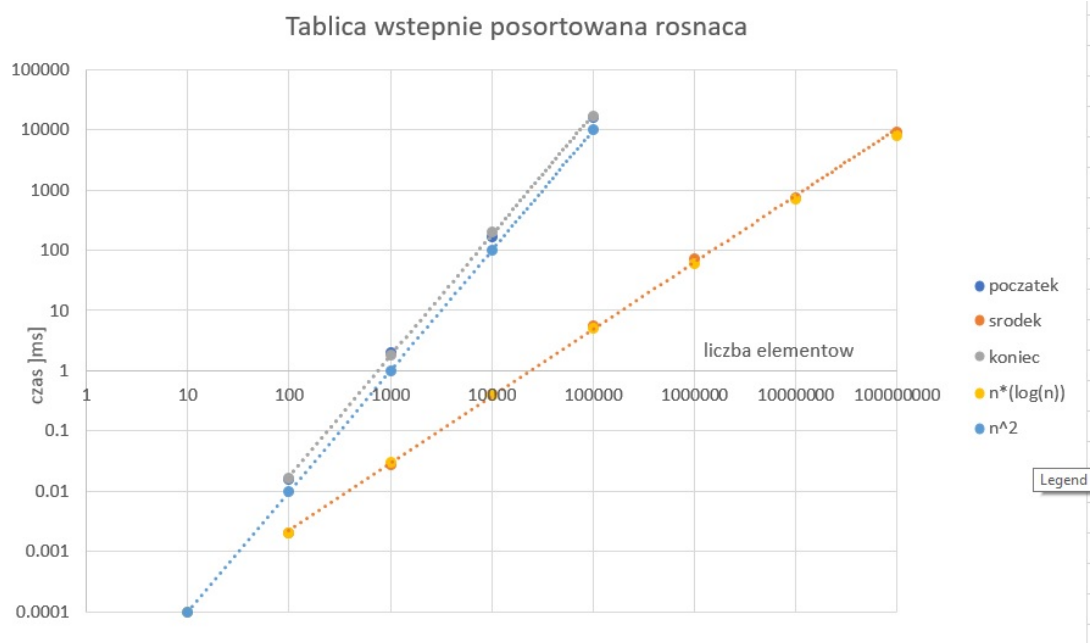


Figure 2: Działanie algorytmu przy wstępnie posortowanej tablicy.

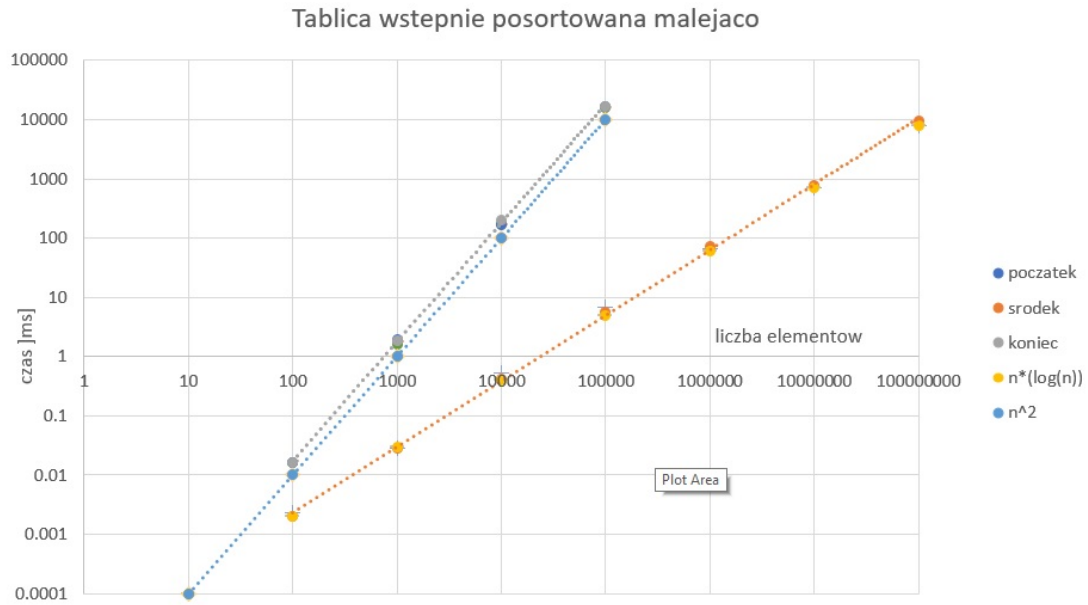


Figure 3: Działanie algorytmu przy wstępnie odwrotnie posortowanej tablicy.

### 3 Wnioski

Jak możemy zauważyć z tabeli powyżej i z wykresów poniżej (skala log-log) w przypadku nieposortowanej tablicy wybór piwota nie miał znaczącego wpływu na czas działania algorytmu i złożoność obliczeniowa utrzymała się idealnie na poziomie  $O(n \log(n))$ .

W przypadku tablicy, która była wstępnie posortowana rosnąco, złożoność obliczeniowa  $O(n \log(n))$  utrzymała się tylko w przypadku doboru piwota w środku tablicy. W przypadku gdy obrano za piwot pierwszy lub ostatni element tablicy, to okazała się, że złożoność obliczeniowa naszego algorytmu wynosiła już  $O(n^2)$ , czyli według literatury najgorszy z zakładanych przypadków algorytmu quicksort

W przypadku gdy program na wejście dostał tablice posortowaną malejąco, to zachował się podobnie do powyższego przypadku gdy tablica była posortowana rosnąco. Czasy są minimalnie lepsze, ale pomiary mogłybyć zakłócone, dlatego, że testy były przeprowadzane w środowisku VirtualBox z uroczomionymi innymi programami.