

# Sprawozdanie

Piotr Partyka

25.03.2016r.

## 1. Zadanie do wykonania:

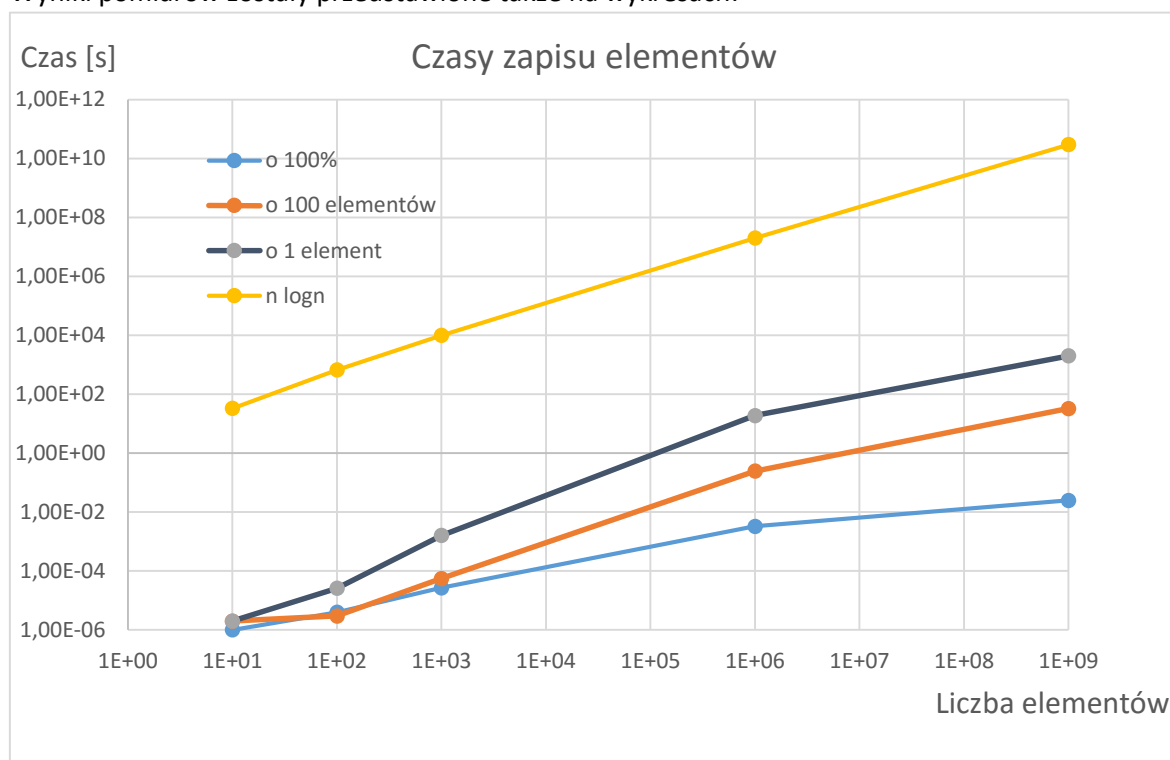
Zadanie było rozszerzeniem poprzedniego zadania. Poszczególne części programu miały zostać podzielone na klasy, a ich obsługa prowadzona poprzez interfejsy do nich stworzone. Ponownie miały też zostać wykonane pomiary czasu, tym razem jednak w ilości 10 pomiarów dla każdej ilości wpisywanych elementów, a z nich wyciągnięta średnia.

## 2. Wyniki pomiarów:

Poniżej została przedstawiona tabela zawierająca wyniki pomiarów dla poszczególnych danych:

Liczba elementów	Zwiększanie rozmiaru o 1	Zwiększanie rozmiaru o 100	Podwajanie rozmiaru
10	$2 * 10^{-6} s$	$2 * 10^{-6} s$	$1 * 10^{-6} s$
$10^2$	$2,6 * 10^{-5} s$	$3 * 10^{-6} s$	$4 * 10^{-6} s$
$10^3$	0,001619 s	$5,6 * 10^{-5} s$	$2,7 * 10^{-5} s$
$10^5$	18,7888 s	0,24728 s	0,003279 s
$10^6$	1996,67 s	32,9492 s	0,025028 s

Wyniki pomiarów zostały przedstawione także na wykresach:



### 3. Spostrzenia i wnioski

W ramach testów zauważony został spadek prędkości wypełniania tablicy przy dużych rozmiarach ( $10^5$ ,  $10^6$  elementów). Pamięć komputera jest zwalniana za każdym powiększeniem tablicy, przyczyna takiego stanu rzeczy nie została zidentyfikowana. Być może jest to normalne zachowanie komputera przy tak dużej liczbie danych, ponieważ testy były przeprowadzane na maszynie wirtualnej.

Tablice dynamiczne są bardzo dobrym rozwiązaniem na przechowywanie danych. Można się po nich wygodnie poruszać i nie wymagają od programisty wiedzy na temat liczby elementów. Przed zaimplementowaniem tablicy dynamicznej należy jednak poświęcić chwilę czasu na rozważenie kroku zwiększania jej rozmiaru. Zbyt mały krok przy dużej ilości danych będzie skutkował koniecznością wykonania wielu operacji kopiowania co pochłonie sporą ilość mocy obliczeniowej. Z kolei zbyt duży krok spowoduje alokację często zbyt dużej ilości pamięci, która mogłaby być spożytkowana w lepszy sposób.