

# Sprawozdanie tablice dynamiczne

Piotr Partyka

18.03.2016r.

## 1. Zadanie do wykonania:

Zadanie miało na celu zapoznanie się z funkcjonowaniem tablic dynamicznych w języku C++. Polegało na zaimplementowaniu klasy, w której znajdowała by się tablica dynamiczna. Następnie należało wpisać do tablicy  $10$ ,  $10^2$ ,  $10^3$ ,  $10^6$  oraz  $10^9$  elementów. Dla każdego wariantu należało przeprowadzić pomiar czasu operacji dla zwiększania rozmiaru tablicy o 1, 100 elementów lub podwajając jej rozmiar.

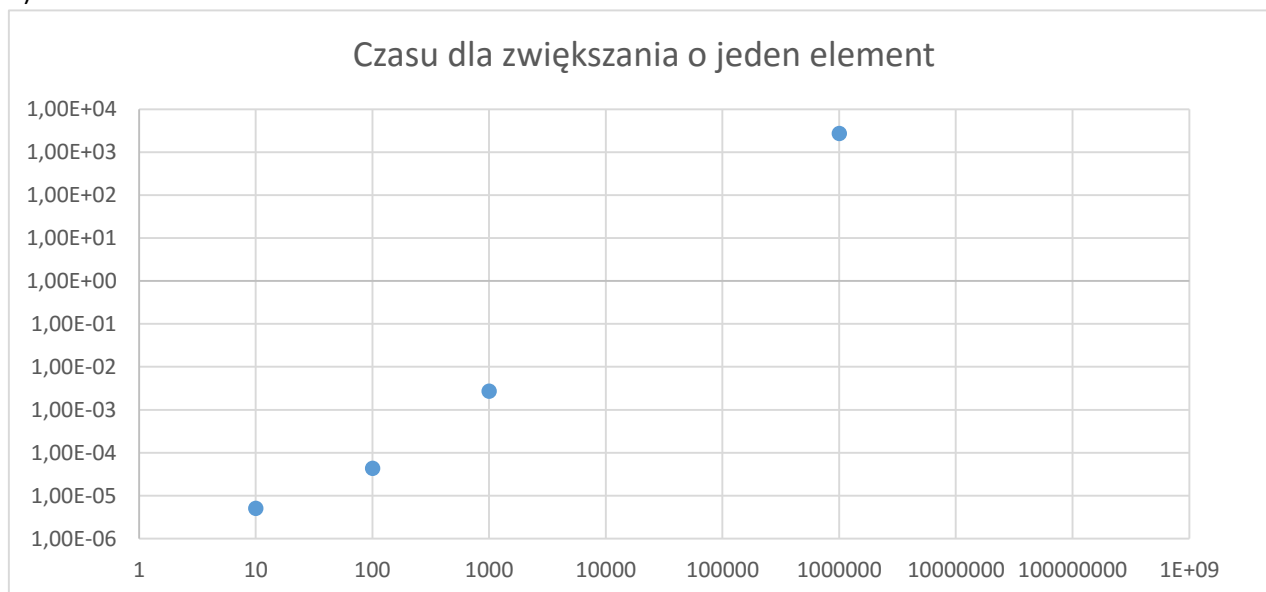
## 2. Wyniki pomiarów:

Poniżej została przedstawiona tabela zawierająca wyniki pomiarów dla poszczególnych danych:

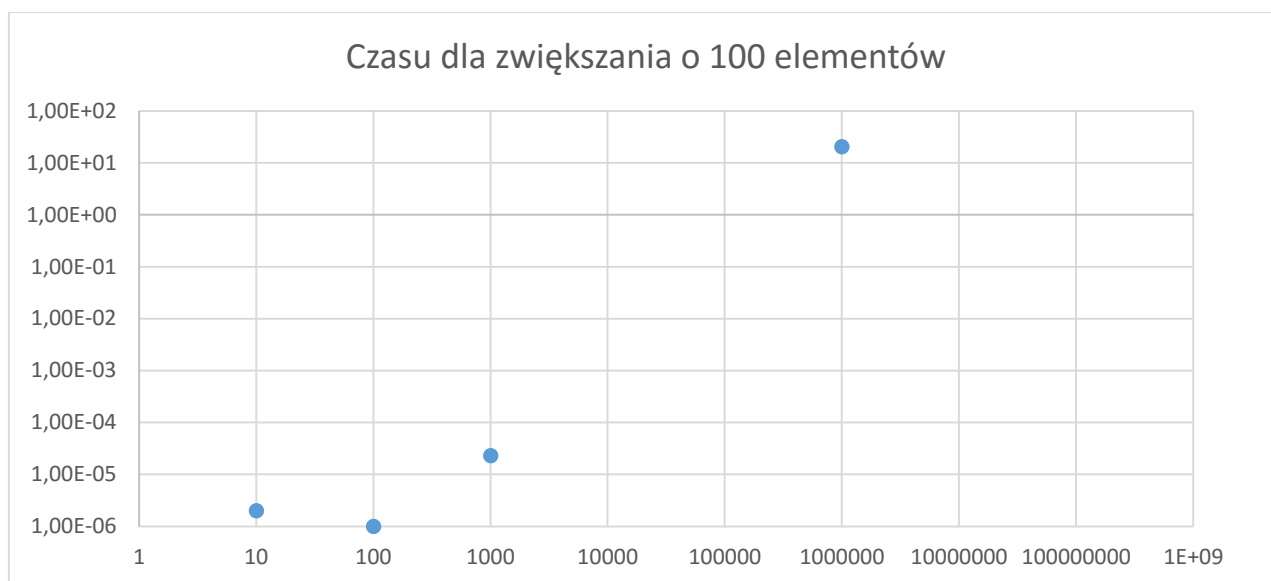
Liczba elementów	Krok o 1	Krok o 100	Krok o 100%
10	$5 * 10^{-6} s$	$2 * 10^{-6} s$	$5 * 10^{-6} s$
$10^2$	$4,3 * 10^{-5} s$	$10^{-6} s$	$3 * 10^{-6} s$
$10^3$	$2,71 * 10^{-3} s$	$2,3 * 10^{-5} s$	$1,8 * 10^{-5} s$
$10^6$	$2,71 * 10^3 s$	23,2 s	0,1512 s
$10^9$	-	-	20,5059 s

Wyniki pomiarów zostały przedstawione także na wykresach:

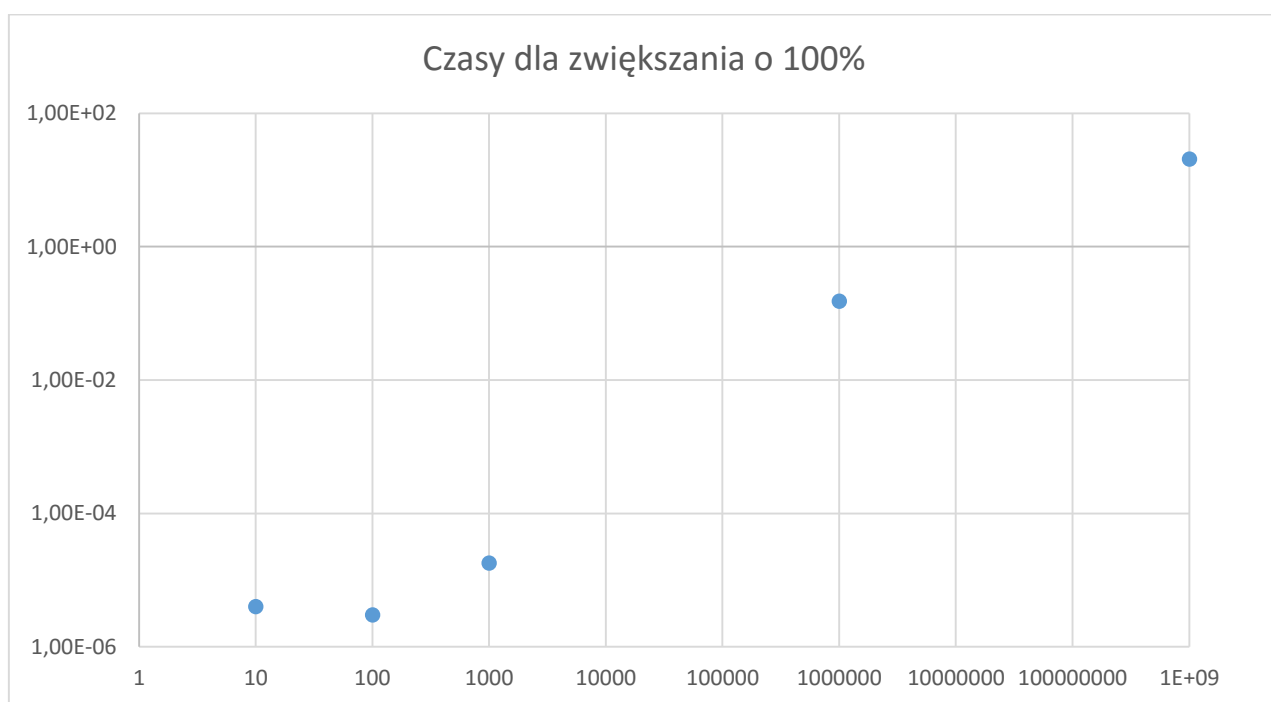
a) krok o 1 element:



b) krok o 100 elementów



c) krok o 100%:



Przy zwiększaniu rozmiaru tablicy o 1 i 100 elementów nie zostały wykonane pomiary dla wpisania  $10^9$  elementów, ponieważ przewyższały one stanowczo moc obliczeniową komputera.

Dla zwiększania rozmiaru tablicy z krokiem o 1 i o 100 elementów można zauważyć zależność kwadratową pomiędzy czasem wykonania operacji, a liczbą elementów do wpisania. Z kolei przy zwiększaniu rozmiaru tablicy o 100% zależność ta jest mniejsza od kwadratowej, lecz nadal nie liniowa.

### 3. Wnioski

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.