

# Sprawozdanie PAMSI

## Tablica dynamiczna, algorytmy powiększania.

- **Wprowadzenie**

Celem ćwiczenia jest pomiar czasu zmiany rozmiarów tablicy dynamicznej, przy użyciu różnych algorytmów powiększania.

Zastosowane algorytmy powiększania:

**Powiększ +1** – algorytm alokuje nową tablicę o rozmiarze  $n+1$  ( $n$  stary rozmiar tablicy), przepisuje starą tablicę do nowej, usuwa starą tablicę i zwraca nową.

**Powiększ  $\times 2$**  – algorytm alokuje nową tablicę o rozmiarze  $n \times 2$ , przepisuje starą tablicę do nowej, usuwa starą tablicę i zwraca nową.

**Powiększ  $n^2$**  – algorytm alokuje nową tablicę o rozmiarze  $n^2$ , przepisuje starą tablicę do nowej, usuwa starą tablicę i zwraca nową.

- Tabele pomiarowe

Nowy Rozmiar	Zwiększ + 1 [ms]	Zwiększ × 2 [ms]	Zwiększ ^2 [ms]
100	1,92	0,17	0,06
	2,58	0,23	0,08
	2,59	0,22	0,09
	2,60	0,21	0,09
	3,20	0,22	0,08
	2,63	0,21	0,08
	2,58	0,23	0,08
	2,65	0,22	0,07
	2,67	0,19	0,09
	2,53	0,20	0,07
1000	10,35	3,0	0,45
	12,91	1,4	0,27
	12,02	1,4	0,28
	9,50	1,7	0,25
	9,21	1,8	0,23
	12,26	1,4	0,25
	11,95	1,8	0,28
	9,45	1,9	0,32
	10,22	1,5	0,27
	12,03	1,7	0,25
10000	74,09	18	7
	73,90	18	7
	73,90	18	8
	73,70	18	9
	73,85	19	7
	73,92	18	7
	74,00	21	9
	79,24	17	8
	78,91	19	7
	78,12	19	8
100000	843,47	90	33
	838,77	83	29
	841,52	91	30
	842,90	87	35
	839,27	92	29
	838,95	74	30
	839,71	78	31
	840,15	80	29
	838,89	91	27
	841,24	85	30
1000000	9406,61	498	62
	9354,22	487	60
	9384,94	425	48
	9400,25	492	65
	9359,78	446	58

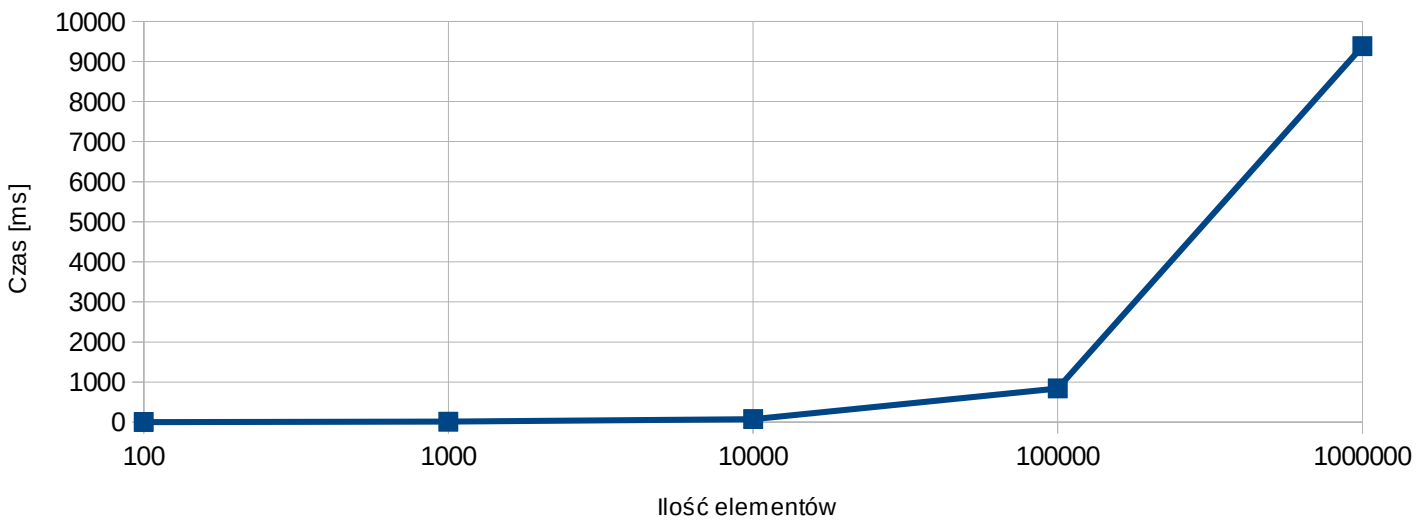
Tab 1

Uśrednione wyniki pomiarów			
Nowy Rozmiar	Zwiększ + 1 [ms]	Zwiększ × 2 [ms]	Zwiększ ^2 [ms]
100	2,595	0,210	0,079
1000	10,989	1,760	0,285
10000	75,364	18,500	7,700
100000	840,485	85,100	30,300
1000000	9381,161	469,600	58,600

Tab 2

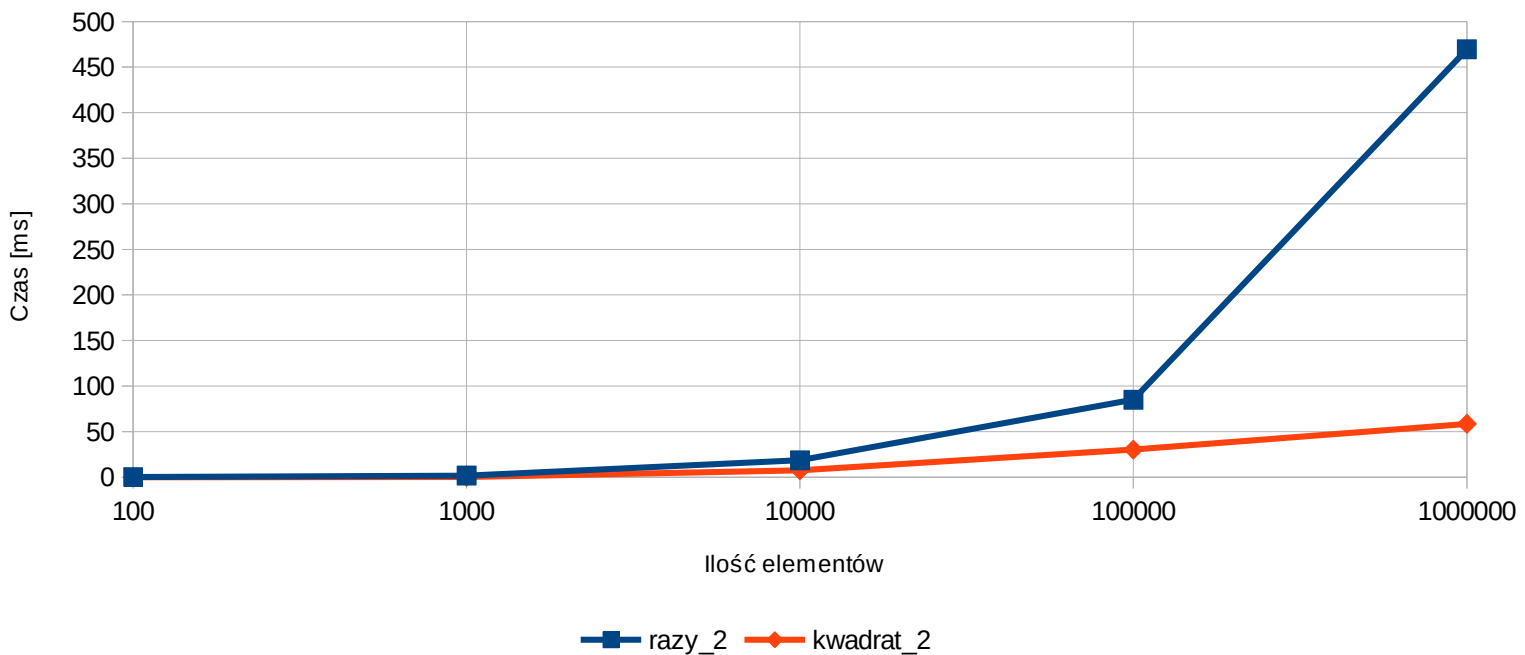
Wykres I

Algorytm zwiększania +1



Wykres II

Porównanie algorytmu zwiększania  $\times 2$  i  $\wedge 2$



### • Wnioski

Porównując wyniki z *Tab 2* i *Wykres I*, *Wykres II* zgodnie z przewidywaniami algorytm powiększania + 1, okazał się być najmniej wydajny. Jest to spowodowane wielokrotną potrzebą alokowania nowej tablicy i przepisywania do niej starej tablicy. Z *Wykres II* widzimy, że algorytm powiększania  $\wedge 2$ , jest szybszy od algorytmu  $\times 2$ . Jest to spowodowane większą ilością wykonywanych akcji alokowania nowej tablicy i przepisywania starej. Nie oznacza to jednak, że algorytm  $\wedge 2$  lub  $\times 2$  jest algorytmem uniwersalnie lepszym, ponieważ tworzy on zwykle tablice o wiele większe od zadanego rozmiaru, alokując przy tym dużo niepotrzebnej pamięci. Aby uzyskać maksymalną wydajność powiększania tablicy dynamicznej powinniśmy dopasowywać skalę powiększenia do odpowiedniego algorytmu.