

# Sprawozdanie

Daniel Majchrzycki, 218476

- Program przypisuje wartość 1 do kolejnych komórek pamięci w tablicy dynamicznej, powiększając ją w przypadku braku miejsca. Program stosuje 4 strategie powiększania tablicy.
  - Powiększanie tablicy o 1 w przypadku braku miejsca.
  - Powiększanie tablicy o 1000 w przypadku braku miejsca.
  - Powiększanie tablicy o 50 000 w przypadku braku miejsca.
  - Powiększanie tablicy 2x w przypadku braku miejsca.

Testy zostały przeprowadzone dla różnych rozmiarów danych wejściowych:

- $10^1$
- $10^2$
- $10^3$
- $10^6$
- $10^7$
- $10^8$
- $10^9$  – nie udało mi się sprawdzić z powodu zbyt małej ilości pamięci RAM na testowym komputerze.

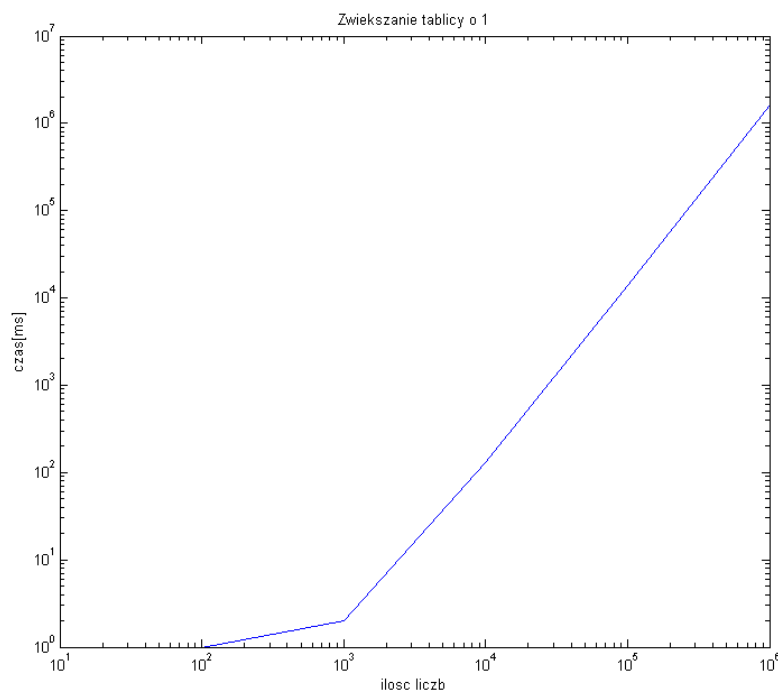
- Wyniki:

Strategia\liczba danych	$10^1$	$10^2$	$10^3$	$10^4$	$10^5$	$10^6$	$10^7$	$10^8$
a)	1	1	2	129	13869	$1.61 \cdot 10^6$	—	—
b)	1	1	1	3	32	1844	15560	—
c)	1	1	1	1	7	53	3261	327667
d)	1	1	1	3	14	68	698	6962

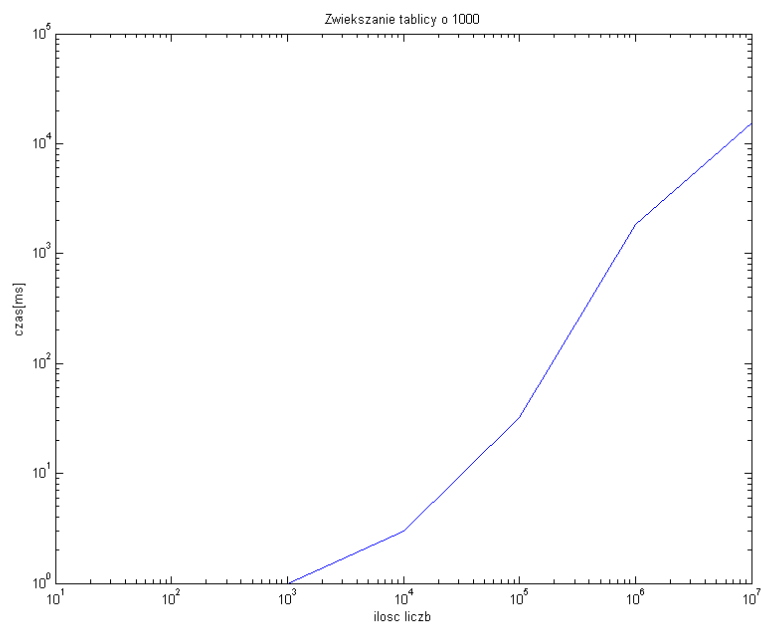
Tabela 1) Wyniki pomiarów różnych strategii powiększania tablic dynamicznych. Wyniki podane w ms.

- Wykresy:

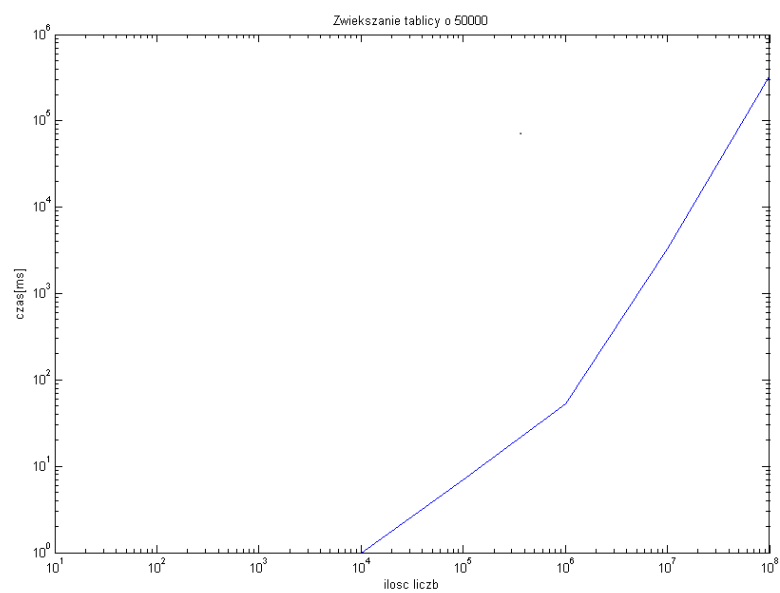
a)



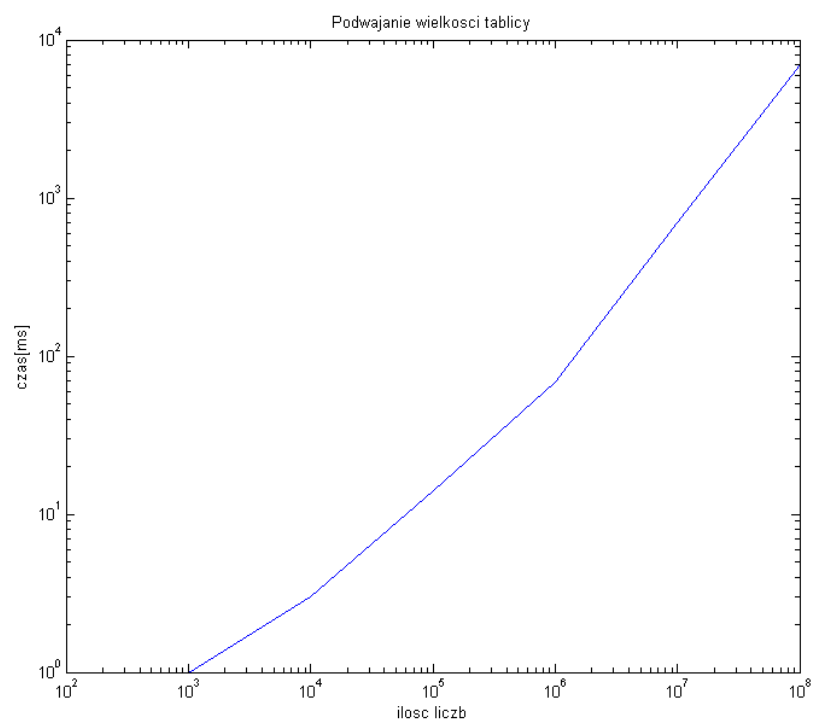
b)



c)



d)



#### 4. Wnioski

Algorytm zwiększania tablicy o 1 jest bezużyteczny w przypadku dużych tablic, z powodu bardzo dużej ilości operacji kopiowania całej tablicy.

Algorytmy zwiększania tablicy o 1000 i 50000 osiągają znacznie lepsze rezultaty, nie powodując przy tym sytuacji w której duża część zaalokowanej pamięci jest nie używana

Algorytm podwajający rozmiar tablicy przy każdym braku pamięci osiąga najlepsze wyniki w przypadku dużych tablic, poprzez znaczną redukcję ilości operacji kopiowania tablic. Może jednak wystąpić sytuacja w której prawie połowa tablicy będzie zaalokowana a nie wykorzystana.