# Przemysław Rola, Juliusz Wasieleski Informatyka, III rok, grupa 6 styczeń 2024

Algorytmy macierzowe – algebra macierzy hierarchicznych – sprawozdanie

# Opis ćwiczenia

Naszym zadaniem było, po wybraniu naszego ulubionego języka, wygenerowanie losowych macierzy.

Następnie mieliśmy porównać czasy wykonania i błędy dla mnożenia macierzy skompresowanych razy wektor oraz razy samą siebie.

# Środowisko, biblioteki, założenia oraz użyte narzędzia

Ćwiczenie wykonaliśmy w języku Python przy użyciu Jupyer Notebooka. Do obliczeń, przechowywania danych użyliśmy bibliotek *numpy, pandas, scipy.*

Do rysowania wykresów użyliśmy biblioteki *matplotlib.*

Wszystkie obliczenia prowadziliśmy na komputerze Lenovo Y50-70 z systemem Windows 10 Pro w wersji 10.0.19045, procesor Intel Core i7-4720HQ 2.60GHz, 2601 MHz, rdzenie: 4, procesory logiczne: 8.

# Implementacje

## Mnożenie macierzy skompresowanej przez wektor

### Pseudokod



### Istotne fragmenty implementacji



## 3.2 Minimal degree permutation

### 3.2.1Pseudokod



### Istotne fragmenty implementacji

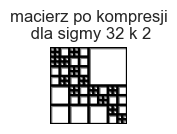


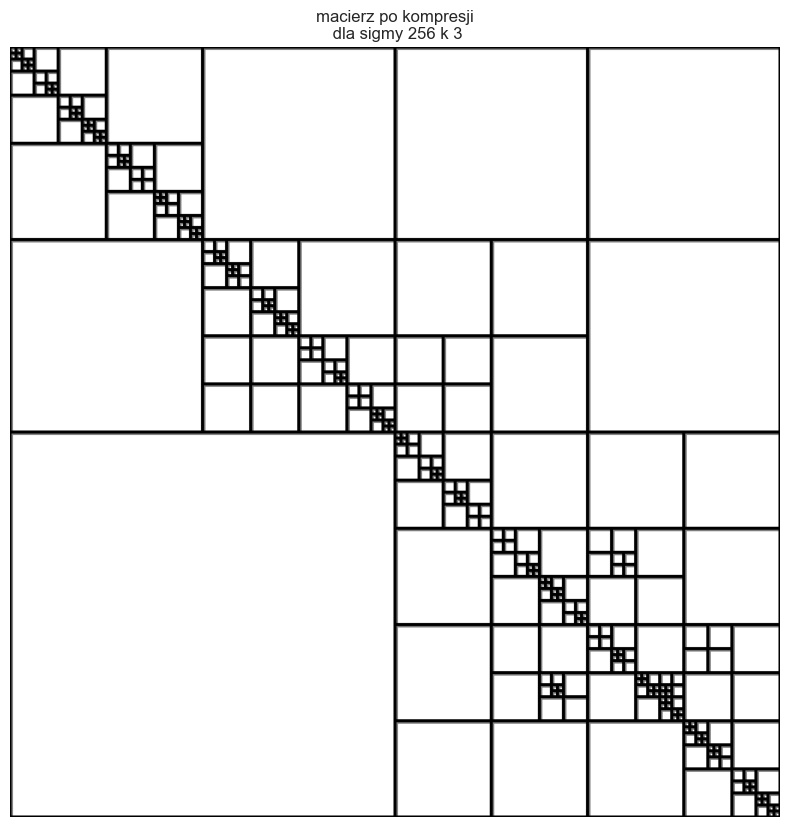
# Analiza wykonanych pomiarów

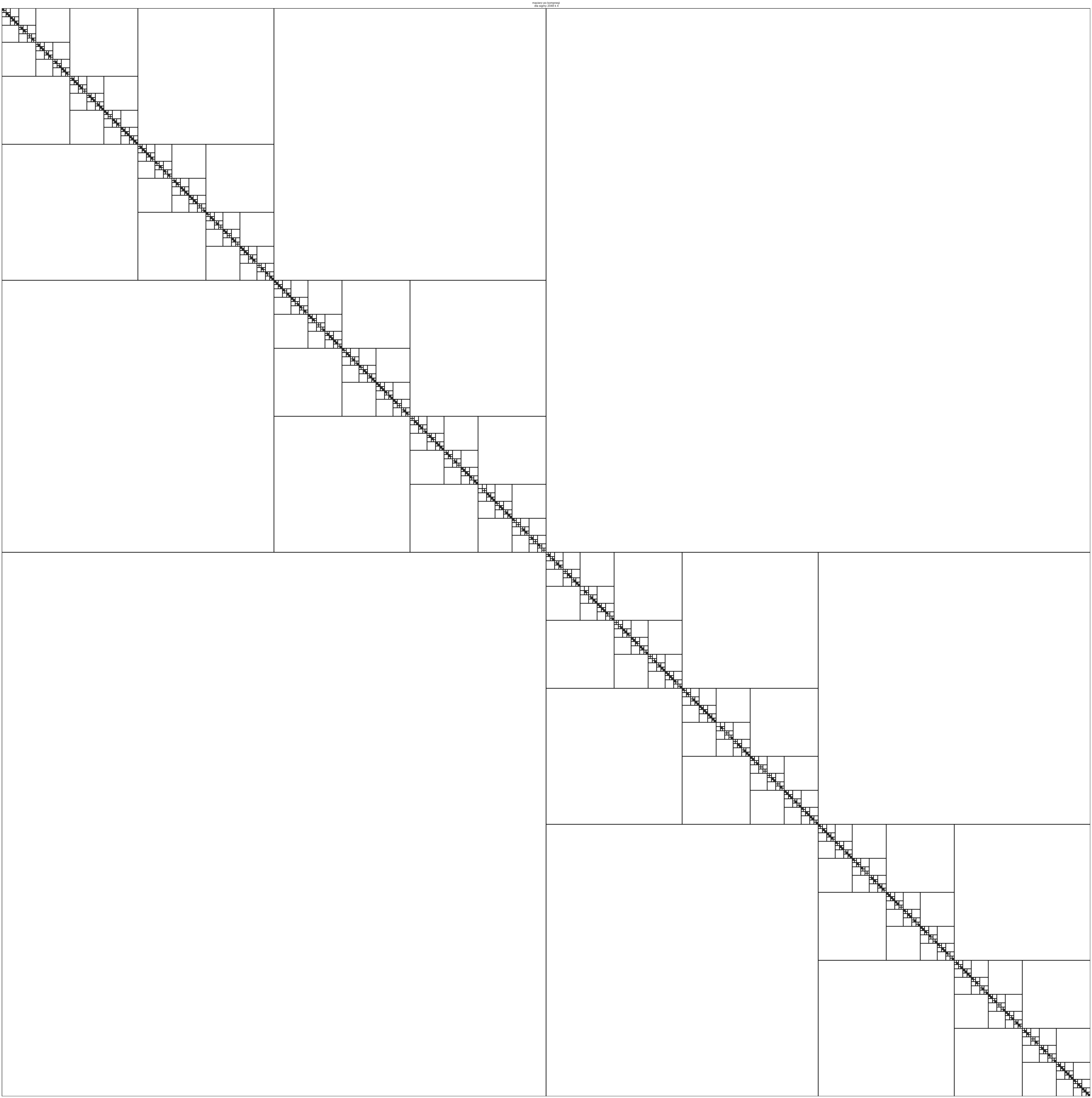
## Pomiary mnożenia razy wektor

| **k** | **s** | **time** | **error** |
| --- | --- | --- | --- |
| 2 | 32 | 0.002167 | 9.893376 |
| 3 | 256 | 0.018169 | 287.397327 |
| 4 | 2048 | 0.101239 | 3557.652625 |

Tab.1 pomiary mnożenia razy wektor







Rys.

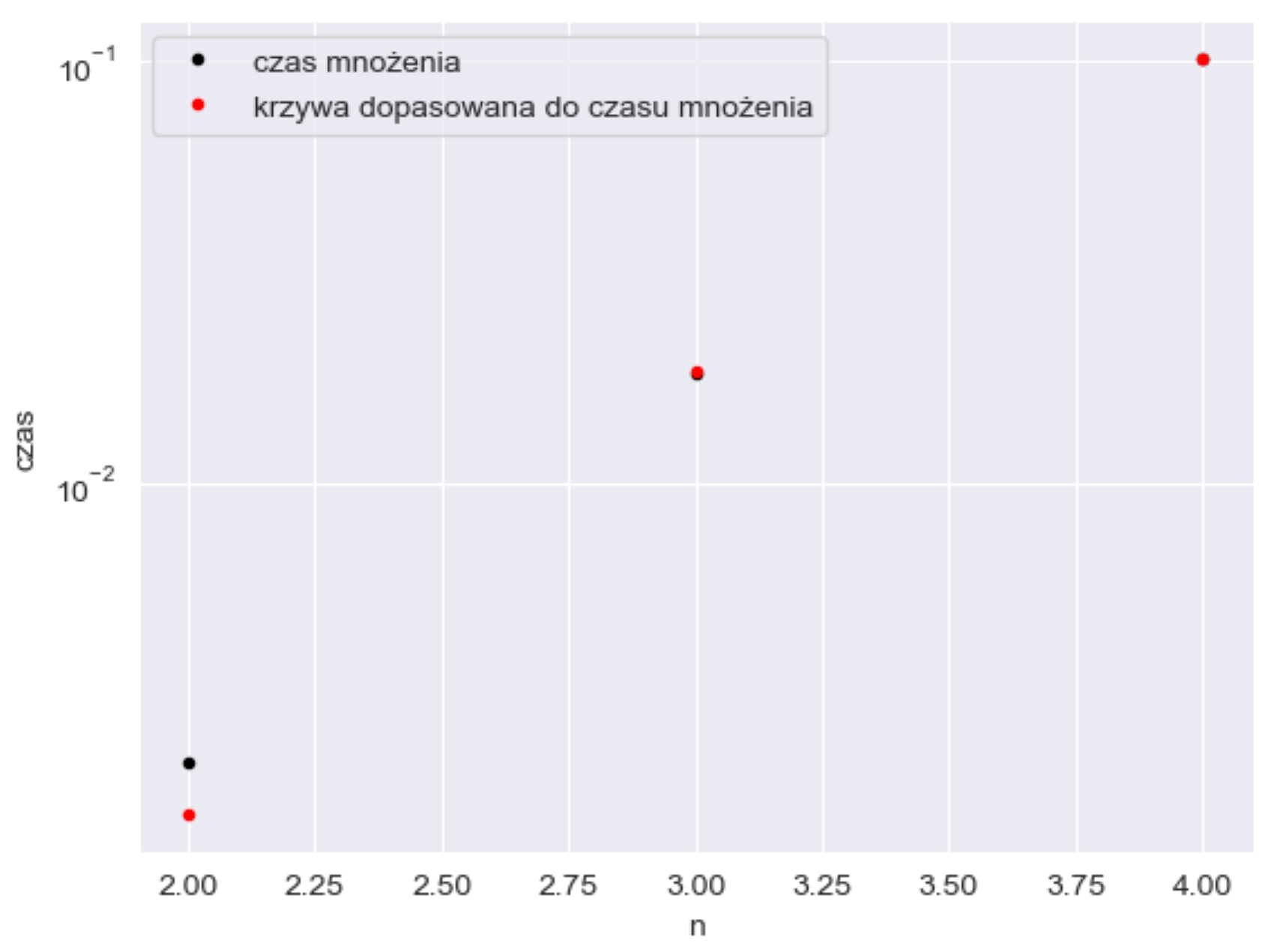
Złożoność obliczeniową szacowaliśmy empirycznie przy użyciu funkcji curve\_fit z modułu scipy.optimize, która aproksymuje funkcję przy użyciu metody najmniejszych kwadratów. My próbowaliśmy aproksymować dane do funkcji postaci:

**(1)**

Gdzie próbowaliśmy oszacować a oraz k.

Na podstawie czasu mnożenia macierzy metodą Bineta otrzymaliśmy:

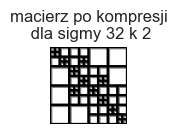
**(2)**

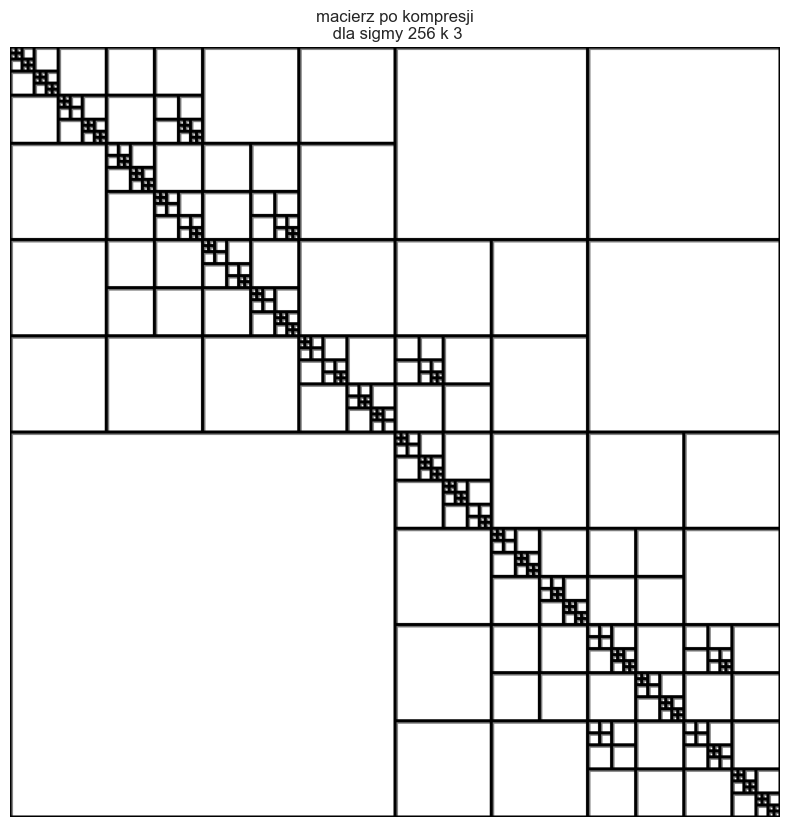


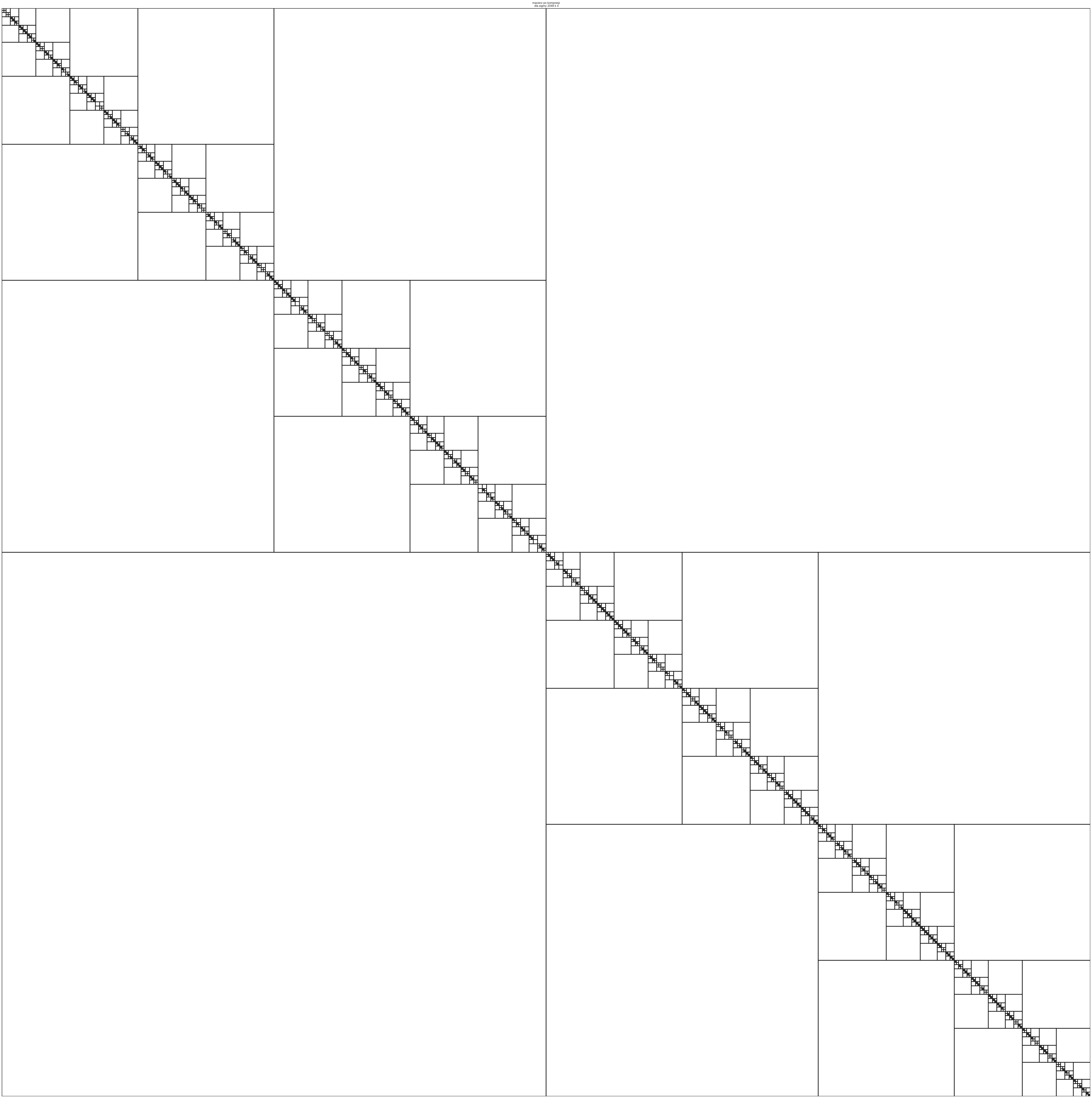
## 4.2 Pomiary mnożenia macierzy razy samą siebie

| **k** | **s** | **time** | **error** |
| --- | --- | --- | --- |
| 2 | 32 | 1.436627 | 168.266712 |
| 3 | 256 | 94.762856 | 2058.513895 |
| 4 | 2048 | 7836.314434 | 21588.409336 |

Tab.2 pomiary mnożenia macierzy skompresowanych







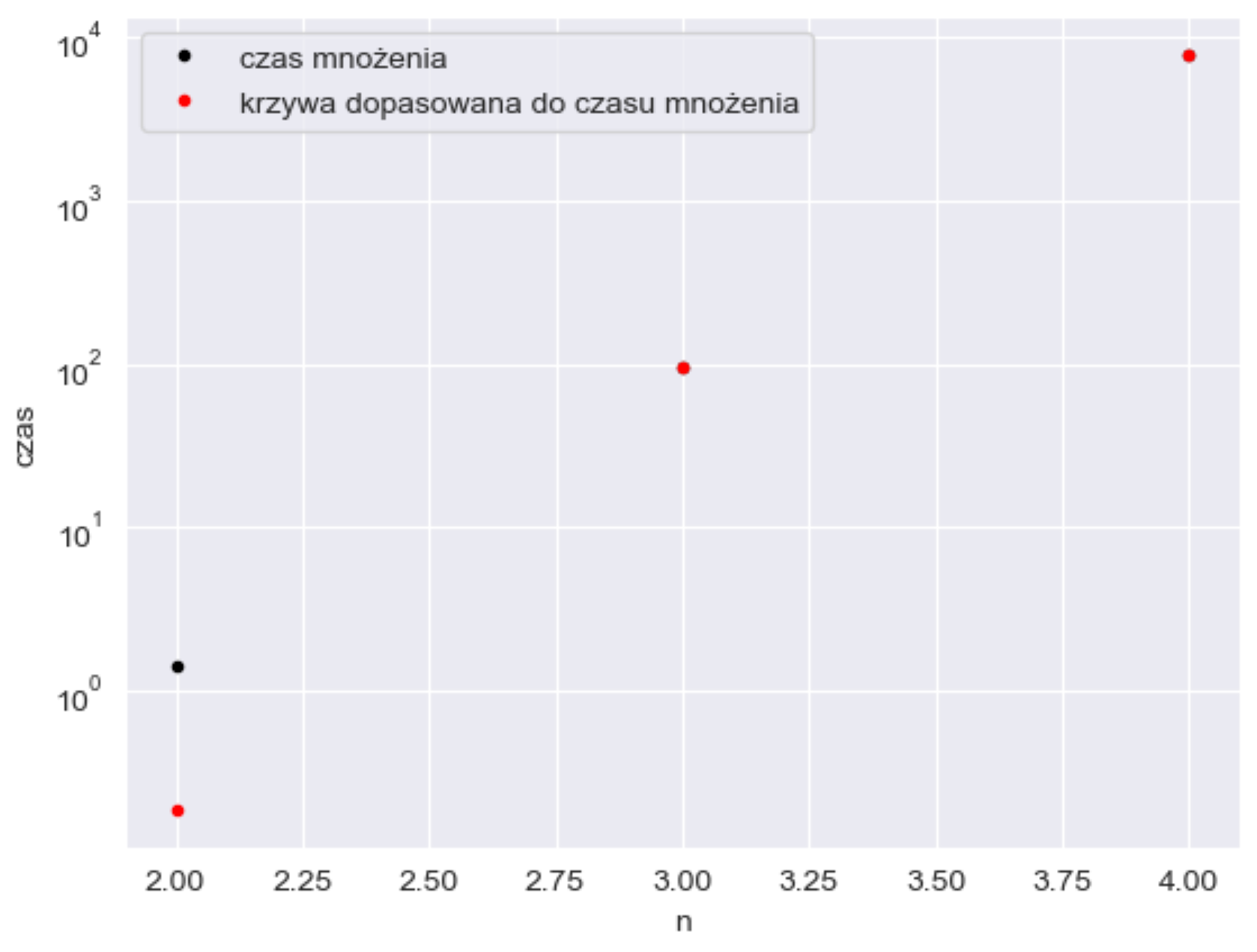
Złożoność obliczeniową szacowaliśmy empirycznie przy użyciu funkcji curve\_fit z modułu scipy.optimize, która aproksymuje funkcję przy użyciu metody najmniejszych kwadratów. My próbowaliśmy aproksymować dane do funkcji postaci:

**(1)**

Gdzie próbowaliśmy oszacować a oraz k.

Na podstawie czasu mnożenia macierzy metodą Bineta otrzymaliśmy:

**(2)**



Rys.2

# Wnioski