# Adam Mytnik, Przemysław Rola, Juliusz Wasieleski Informatyka, III rok, grupa 6 październik 2023

Algorytmy macierzowe – rekurencyjne mnożenie macierzy - sprawozdanie

# Opis ćwiczenia

Naszym zadaniem było , po wybraniu naszego ulubionego języka, wygenerowanie losowych macierzy których elementy są z przedziału i zaimplementowanie algorytmów:

* Rekurencyjnego mnożenia macierzy metodą Binet’a
* Rekurencyjnego mnożenia macierzy metodą Strassena
* Rekurencyjnego mnożenia macierzy metodą zaproponowaną przez sztuczną inteligencję.

Następnie, mieliśmy sprawdzić działanie naszych implementacji na losowo wygenerowanych macierzach rozmiarów gdzie .

# Środowisko, biblioteki, założenia oraz użyte narzędzia

Ćwiczenie wykonałem w języku Python przy użyciu Jupyer Notebooka. Do obliczeń, przechowywania danych użyłem bibliotek *numpy, pandas.*

Do rysowania wykresów użyłem biblioteki *matplotlib.*

Wszystkie obliczenia prowadziłem 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.

# Implementacja algorytmów

## Rekurencyjne mnożenie macierzy metodą Binet’a

### Pseudokod

### Istotne fragmenty implementacji

## Rekurencyjne mnożenie macierzy metodą Strassena

### Pseudokod

SMU*(A,B): # Strassen Matrix Multiplication*

**Jeżeli** *A* oraz *B* mają rozmiar 1**:**

**Zwróć** *A*\**B*

**W przeciwnym wypadku:**

Podziel *A* i *B* na 4 równych rozmiarów mniejsze macierze

**Zapisz** do pomocniczych zmiennych *M*:

M1 = SMU(A11+A22, B11+B22)

M2 = SMU(A21+ A22, B11)

M3 = SMU(A11, B12- B22)

M4 = SMU(A22, B21- B11)

M5 = SMU(A11+ A12, B22)

M6 = SMU(A21- A11, B11+ B12)

M7 = SMU(A12- A22 B21+B22)

**Zapisz** macierz C jako:

C1 = M1 + M4 - M5 + M7

C2 = M3 + M5

C3 = M2 + M4

C4 = M1- M2 + M3 + M6

**Zwróć**C

### Istotne fragmenty implementacji

Do operacji macierzowych dodawania posłużyła biblioteka numpy.

## Rekurencyjne mnożenie macierzy metodą zaproponowaną przez sztuczną inteligencję

### Pseudokod

### Istotne fragmenty implementacji

# Analiza wykonanych pomiarów

## Pomiary mnożenia macierzy metodą Binet’a

### Pomiary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| k | operacje addytywne | operacje multiplikatywne | Wszystkie operacje zmiennoprzecinkowe | czas wykonania |
| 2 | 48 | 64 |  | 0.000194 |
| 3 | 448 | 512 |  | 0.001333 |
| 4 | 3840 | 4096 |  | 0.011614 |
| 5 | 31744 | 32768 |  | 0.084631 |
| 6 | 258048 | 262144 |  | 0.731710 |
| 7 | 2080768 | 2097152 |  | 6.501183 |
| 8 | 16711680 | 16777216 |  | 51.505552 |

Tab. Xx Pomiary cos

### Analiza wyników

#### Zależność operacji addytywnych od multiplikatywnych

Coś

#### Szacunek złożoności obliczeniowej

Coś

## Pomiary mnożenia macierzy metodą Strassena

### Pomiary

### Analiza wyników

#### Zależność operacji addytywnych od multiplikatywnych

Coś

#### Szacunek złożoności obliczeniowej

Coś

## Pomiary mnożenia macierzy metodą zaproponowaną przez sztuczną inteligencję

### Pomiary

### Analiza wyników

#### Zależność operacji addytywnych od multiplikatywnych

Coś

#### Szacunek złożoności obliczeniowej

Coś

## Porównanie wyników trzech powyższych algorytmów

## Porównanie z Octave / MATLAB

# Wnioski

* Wiosek 1
* Wniosk 2