**Raport 1**

Rekurencyjne mnożenie macierzy

*Michał Kobiera, Maciej Pięta*

1. **Opis pseudokodu**

Algorytm Binet`a:

* Jeśli rozmiar macierzy wynosi 2x2:
  + Pomnóż macierz standardowym mnożeniem macierzy
  + Zwróć wynik
* W przeciwnym przypadku:
  + Podziel macierze na 4 mniejsze części (A11, A12, A21, A22)
  + Rekurencyjnie oblicz cztery iloczyny macierzy  
    C11 = wynik mnożenia A11 i B11, a także wynik mnożenia A12 i B21, dodane do siebie.

C12 = wynik mnożenia A11 i B12, a także wynik mnożenia A12 i B22, dodane do siebie.

C21 = wynik mnożenia A21 i B11, a także wynik mnożenia A22 i B21, dodane do siebie.

C22 = wynik mnożenia A21 i B12, a także wynik mnożenia A22 i B22, dodane do siebie.

* + Zwróć obliczoną macierz C jako wynik

Algorytm Strassena:

* Jeśli rozmiar macierzy wynosi 2x2:
  + Pomnóż macierz standardowym mnożeniem macierzy
  + Zwróć wynik
* W przeciwnym przypadku:
  + Podziel macierze na 4 mniejsze części (A11, A12, A21, A22)
  + Wykonaj siedem rekurencyjnych wywołań, obliczając siedem częściowych wyników:  
    P1 = StrassenMlt(A11 + A22, B11 + B22)

P2 = StrassenMlt(A21 + A22, B11)

P3 = StrassenMlt(A11, B12 - B22)

P4 = StrassenMlt(A22, B21 - B11)

P5 = StrassenMlt(A11 + A12, B22)

P6 = StrassenMlt(A21 - A11, B11 + B12)

P7 = StrassenMlt(A12 - A22, B21 + B22)

* + Oblicz cztery podmacierze wynikowe C11, C12, C21 i C22 za pomocą wyrażeń:

C11 = P1 + P4 - P5 + P7

C12 = P3 + P5

C21 = P2 + P4

C22 = P1 - P2 + P3 + P6

* + Zwróć obliczoną macierz C jako wynik

Algorym AI

1. **Kluczowe fragmenty kodu**

**Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie**

Rysunek 1

Do generowania macierzy została wykorzystana funkcja random.random() z modułu numpy

Funkcja random\_matrix() zwraca macierz wypełnioną wartościami z przedziału [min\_val, max\_val)

**Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie**

Rysunek 2

Implementacja algorytmu Binet`a

Macierze z modułu numpy dzielone są na mniejsze bloki

Następnie rekurencyjnie obliczana jest macierz C

Funkcja zwraca wynik „składając” macierz C wykorzystując np.hstack() orac np.vstack()

**Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie**

Rysunek 3

Implementacja algorytmu Strassena

Analogicznie jak w poprzednim punkcie, macierz dzielona jest na mniejsze bloki

Następnie rekurencyjnie liczone są bloki pomocnicze P według podanych wzorów

Z obliczonych mniejszych bloków składana i zwracana jest macierz wynikowa C.

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Rysunek 4

Obie implementacje w warunku końcowym rekurencji korzystają ze standardowego mnożenia macierzy 2x2

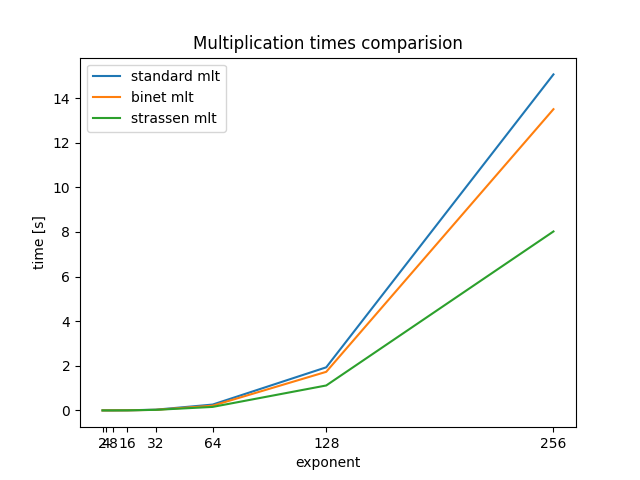
**Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie**

Rysunek 5

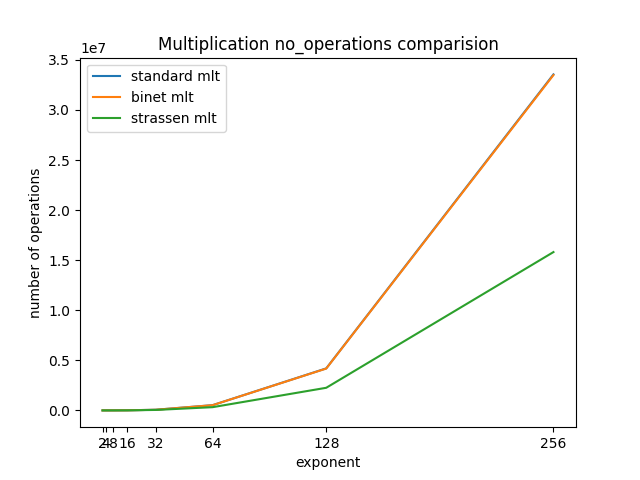
W celu przetestowania szybkości działania implementacji zostały obliczone szybkość wykonania  
oraz ilość operacji dla poszczególnych wykładników

1. **Wykresy**

****

Wykres 1

Wykres ilustrujący porównanie szybkości wykonania poszczególnych algorytmów.  
*(Linie na wykresie zostały dodane w celu poprawy czytelności obrazu)*



Wykres 2

Wykres ilustrujący ilość wykonanych operacji zmiennoprzecinkowych dla poszczególnych algorytmów.  
Wyniki dla standardowego mnożenia macierzy i mnożenia metodą Binte`a są bardzo podobne i nie widać różnicy liczbie wykonanych operacji dla takiej skali na wykresie.  
*(Linie na wykresie zostały dodane w celu poprawy czytelności obrazu)*

**Obraz zawierający tekst, zrzut ekranu, wyświetlacz, diagram

Opis wygenerowany automatycznie**

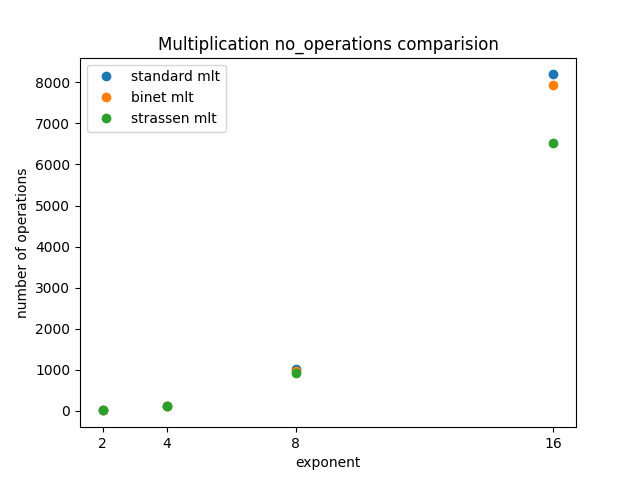
Wykres 3

**Obraz zawierający tekst, zrzut ekranu, linia, diagram

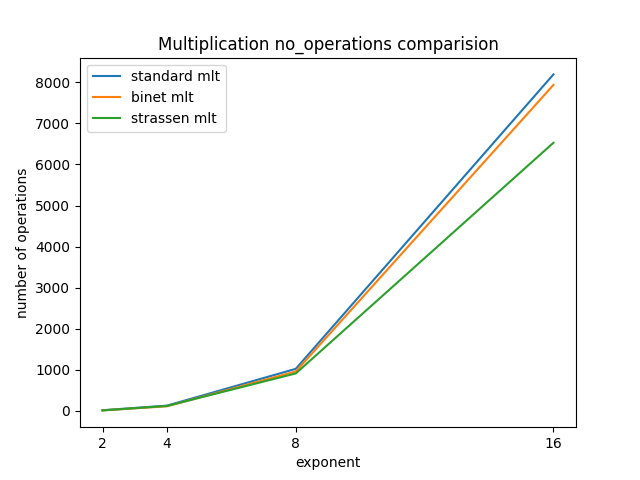
Opis wygenerowany automatycznie**

Wykres 4

Wykresy ilustrujące szybkość działania poszczególnych algorytmów przy niskich wartościach wykładnika  
*(Na drugim wykresie poglądowo zostały dodane linie łączące punkty)*

****

Wykres 5

****

Wykres 6

Wykresy ilustrujące szybkość działania poszczególnych algorytmów przy niskich wartościach wykładnika  
*(Na drugim wykresie poglądowo zostały dodane linie łączące punkty)*

1. **Ocena złożoności obliczeniowej**
2. **Porównanie z MATLAB**