Szymon Konopek A=2, B=1, C=3, D

................................................................... ---------------------------------

(Imię i nazwisko) (A, B, C, D)

Parametry:

M = 6

N = 26

**Raport z Pracowni nr 1**

Zadanie 1

1. Cel zadania

Celem zadania było obliczenie złożoności obliczeniowej metody Banachiewicza.

1. Metody

W doświadczeniu wykorzystano kilka klas stworzonych w języku Python. Odpowiedni projekt stworzono i kompilowano w środowisku VSCode na komputerze stacjonarnym z procesorem Intel Core i5 9600k.

1. Przebieg doświadczenia i wyniki

Doświadczenie rozpoczęto od ustalenia maksymalnego rozmiar macierzy n. Z kilku prób ustalono, że dla

* n = 300, czas rozwiązywania układu wynosi 0.47 sekundy.

Opracowano metodę mierz\_czas, która pozwala obliczyć czas sortowania macierzy o

zadanej długości jedną z dwóch badanych metod. Poniżej zamieszczono kod tej

metody:

def mierz\_czas(self, metoda, k):

czas = 0.0

ukl = uklad.Uklad(k)

for \_ in range(self.M):

ukl.losuj\_uklad\_symetryczny\_dodatnio\_okreslony()

if (metoda == 1):

m1 = banachiewicz.Banachiewicz(ukl)

stoper = time.time()

m1.rozklad()

m1.rozwiaz\_trojkatny\_dolny()

m1.rozwiaz\_trojkatny\_gorny()

czas = time.time() - stoper

elif (metoda == 2):

m2 = gauss.Gauss(ukl)

stoper = time.time()

m2.eliminacja()

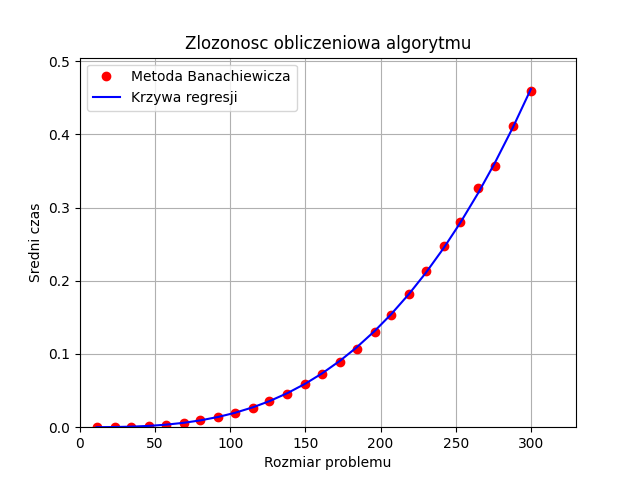
m2.rozwiaz\_trojkatny()

czas = time.time() - stoper

return czas/self.M

Następnie wywołano metodę *badaj\_zlozonosc* podając jako parametry klasy *Banachiewicz* obiekt *ukl* będącym układem o maksymalnym rozmiarze macierzy *n = 300.*

Wykres 1. Zależność czasu obliczania wyniku macierzy metodą Banachiewicza od wielkości macierzy.



Empiryczna złożoność obliczeniowa wyniosła: n3.0144 co bliskie jest teoretycznej wartości n3 .

Wnioski

W wyniku przeprowadzonego eksperymentu udało się oszacować złożoność obliczeniową algorytmu z wykorzystaniem metody Banachiewicza. Otrzymana eksperymentalna złożoność jest bliska teoretycznej wartości 𝑂(𝑛3).

Zadanie 2

1. Cel zadania

Celem zadania było porównanie dwóch metod obliczania macierzy przy użyciu – metody Banachiewicza oraz metody Gaussa.

1. Metody

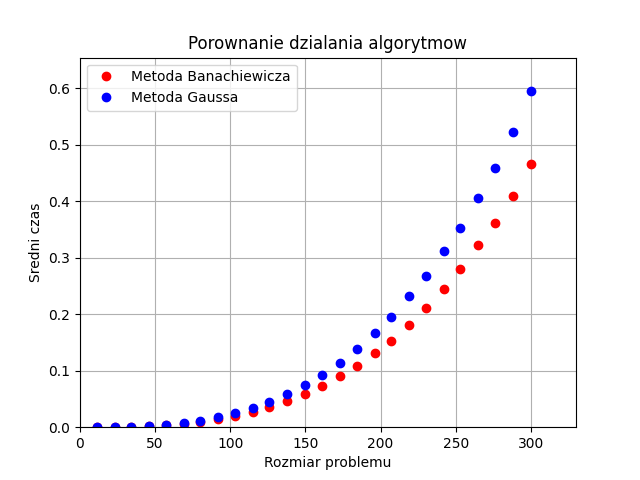
W doświadczeniu wykorzystano kilka klas stworzonych w języku Python. Odpowiedni projekt stworzono i kompilowano w środowisku VSCode na komputerze stacjonarnym z procesorem Intel Core i5 9600k.

1. Przebieg doświadczenia i wyniki

Wykorzystano maksymalną wielkość macierzy ustaloną w zadaniu 1. Ze względu na to, że w zadaniu 1 opracowano metodę *mierz\_czas* w ten sposób, aby umożliwiała zastosowanie obu metod, można ją wykorzystać w tym zadaniu. Po wywołaniu metody *porownaj\_metody* dla obiektów klasy *Banachiewicz* oraz *Gauss* z parametrem *ukl* będącym obiektem klasy *Uklad,* będącym macierzą o wielkości *n = 300,* udało się uzyskać wykres zamieszczony na następnej stronie.

1. Wnioski

W wyniku przeprowadzonego doświadczenia okazało się, że metoda Banachiewicza jest szybsza od metody Gaussa. różnice czasów dla krótkich list są niewielkie, ale rosną wraz ze wzrostem długości listy.



Wykres 2. Porównanie metod obliczania macierzy

(czerwone punkty – sortowanie przez wstawianie, niebieskie – sortowanie przez wybieranie)