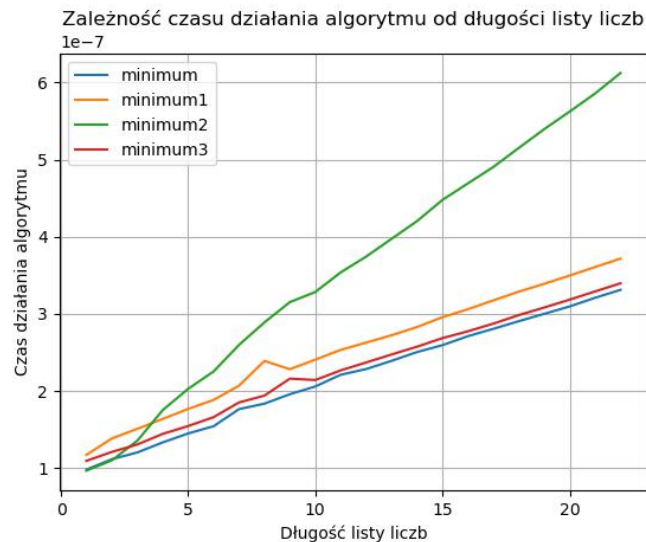


## Lista 1, zadanie 1

minimum- to algorytm, który porównuje elementy znajdujące się na liście  
minimum1- to algorytm, który tworzy kopię listy  
minimum2- to algorytm, który wykorzystuje pętlę while  
minimum3- to algorytm, który wykorzystuje iteratory

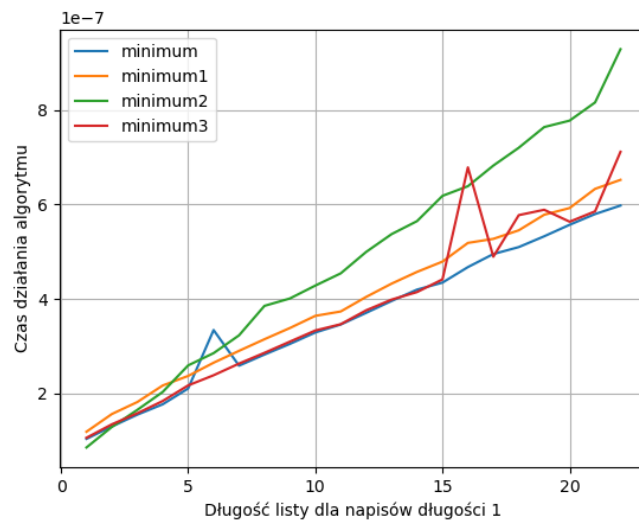
### 1. Czas działania algorytmu zależny od długości listy dla list liczb



Rysunek 1. Wykres: Czas działania algorytmu zależny od długości listy dla list liczb

W początkowej fazie obliczeń dla list długości (czyt. elementów) 1,2,3 algorytm minimum2 okazał się najszybszy, jednak później od listy długości większej, niż 4 stał się najwolniejszym. Najszybszym okazał się algorytm pod nazwą minimum. Drugi najszybszy był algorytm minimum3, a trzeci minimum1.

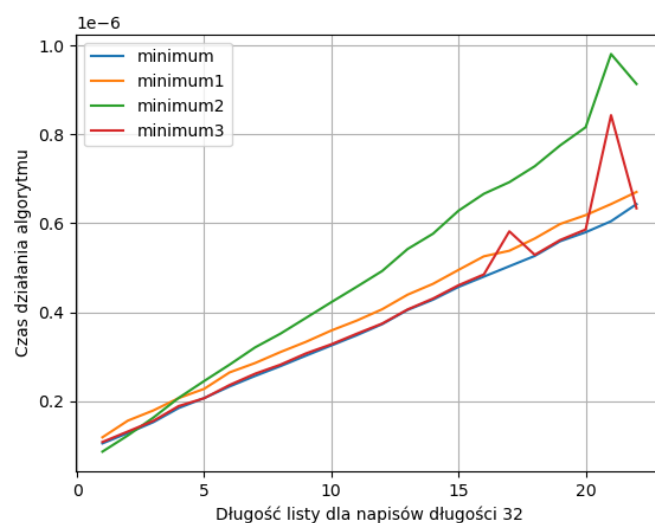
## 2. Czas działania algorytmu zależny od długości listy dla list napisów długości 1



Rysunek 2. Wykres: Czas działania algorytmu zależny od długości listy dla list napisów długości 1

W początkowej fazie obliczeń dla list o małej długości (1,2,3 elementy) algorytm minimum2 okazał się najszybszy, jednak później dla coraz to dłuższych list stał się najwolniejszy. Najszybszym okazał się algorytm pod nazwą minimum, jednak dla listy o liczbie elementów 6 był najwolniejszy. Między minimum i minimum3 w początkowej fazie wyliczeń (długość listy 1,2,3,4,5) jak i środkowej (długość listy 7,8,9,10,11,12,13,14,15,18) oraz fragment dla list długości od 20,21,22 algorytmy te działały w podobnym czasie, różnica jest praktycznie niezauważalna. Minimum1 było najwolniejsze na początku (długość listy od 1,2,3,4), później uplasowało się, można rzec jako 3 najszybszy algorytm.

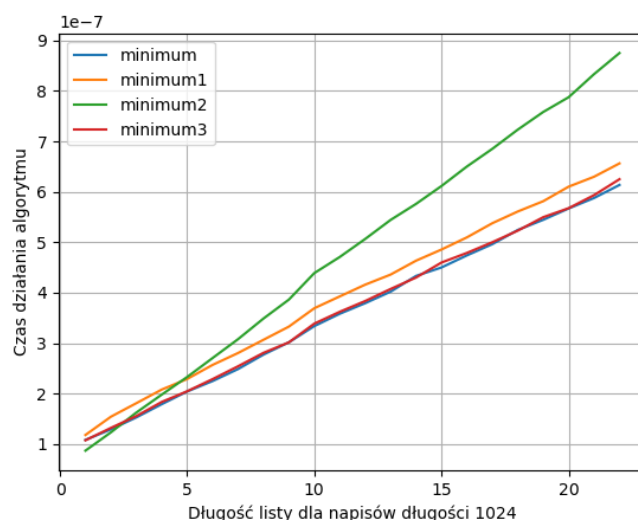
## 3. Czas działania algorytmu zależny od długości listy dla list napisów długości 32



Rysunek 3. Wykres: Czas działania algorytmu zależny od długości listy dla list napisów długości 32

W początkowej fazie obliczeń dla list składających się z 1,2, jak i 3 elementów algorytm minimum2 okazał się najszybszy, jednak później od listy długości większej niż 3 stał się najwolniejszym. Bardzo sprawny okazał się algorytm minimum. Choć między minimum3, a minimum nie było dużej konkurencji (różnice w czasie wyliczeń są niezauważalne praktycznie) to dla ostatniej długości listy, czyli 23 elementowej wygrywa minimum3. Zarówno dla listy o długości (czyt. liczba elementów w liście) 16,17,18, jak i 20,21,22 algorytm minimum szybciej poradził sobie z wyznaczeniem minimum. Minimum1 dla list długości 1,2,3,4 było najwolniejsze. Dla reszty list o różnych długościach zajmowało 3 miejsce pod względem szybkości.

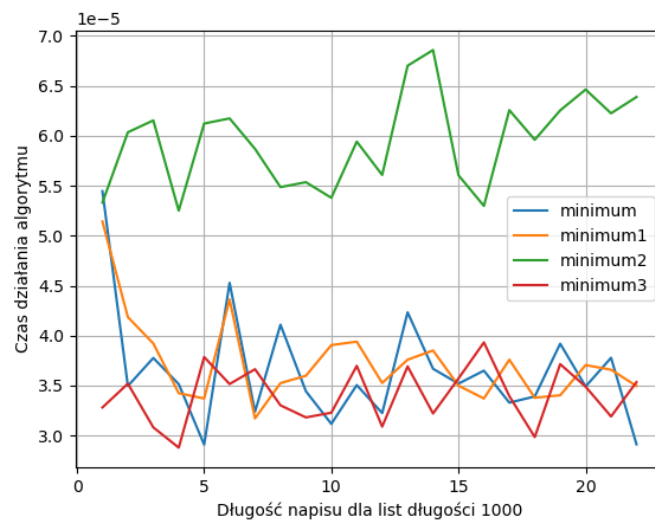
#### 4. Czas działania algorytmu zależny od długości listy dla list napisów długości 1024



Rysunek 4. Wykres: Czas działania algorytmu zależny od długości listy dla list napisów długości 1024

Najwolniejszy okazał się algorytm minimum2, chociaż z listami o długości 1,2 liczył najszybciej, a dla długości list 3,4,5 był trzecim najszybszy. Minimum okazało się najsprawniejsze w wyliczeniach, choć konkurowało z minimum3 (różnice praktycznie nie widoczne w ich czasie wyliczeń minimum z listy) to dla list długości 15,16,18 i 22,23 wygrywa minimum. Minimum1 liczyło najwolniej dla list długości 1,2,3,4. Później algorytm ten zajął trzecie miejsce.

## 5. Czas działania algorytmu zależny od napisu dla list długości 1000

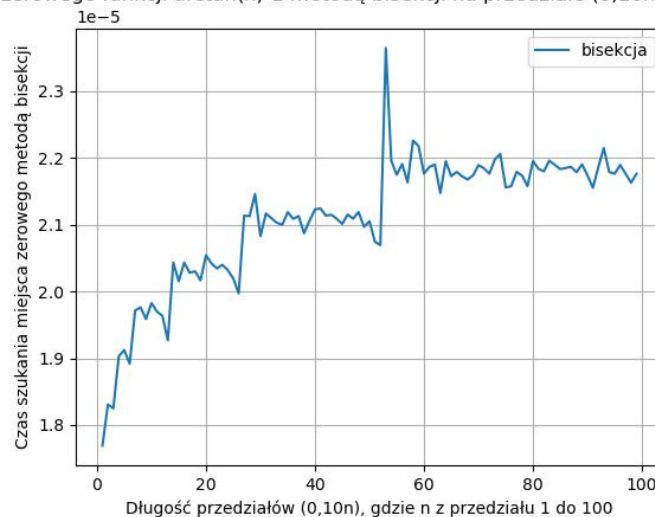


Rysunek 5. Wykres: Czas działania algorytmu zależny od napisu dla list długości 1000

Minimum2 dla każdej długości napisu listy 1000-elementowej poza 1 okazało się najwolniejsze. Dla napisów długości 2,3,4,9,10,11,12,14,17,20 najwolniejszym algorytmem okazał się minimum1, najszybciej policzył dla napisu długości 16. Najszybciej z różnymi długościami radził sobie algorytm pod nazwą minimum3, najwolniej liczył dla długości napisów: 5,16. Minimum najszybciej policzyło dla długości napisów 5,10,11,22,23, a najwolniej dla 1,6,8,13,18,19,21.

## zadanie 2

a zerowego funkcji  $\arctan(x)-1$  metodą bisekcji na przedziale  $(0,10n)$ , gdzie  $n$



Rysunek 6. Wykres: Czas szukania miejsca zerowego funkcji  $\arctan(x)-1$  metodą bisekcji na przedziale  $(0,10n)$ , gdzie  $n$  z przedziału 1 do 100

Otrzymany wykres jest zbliżony do wykresu logarytmu o podstawie 2