

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №3
дисциплины «Алгоритмизация»
Вариант 29

Выполнил:
Саенко Андрей Максимович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», заочная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., канд. технических
наук, доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____
Ставрополь, 2023 г.

Тема: Метод наименьших квадратов

Цель: Научиться применять метод наименьших квадратов и находить коэффициент парной корреляции.

Порядок выполнения работы:

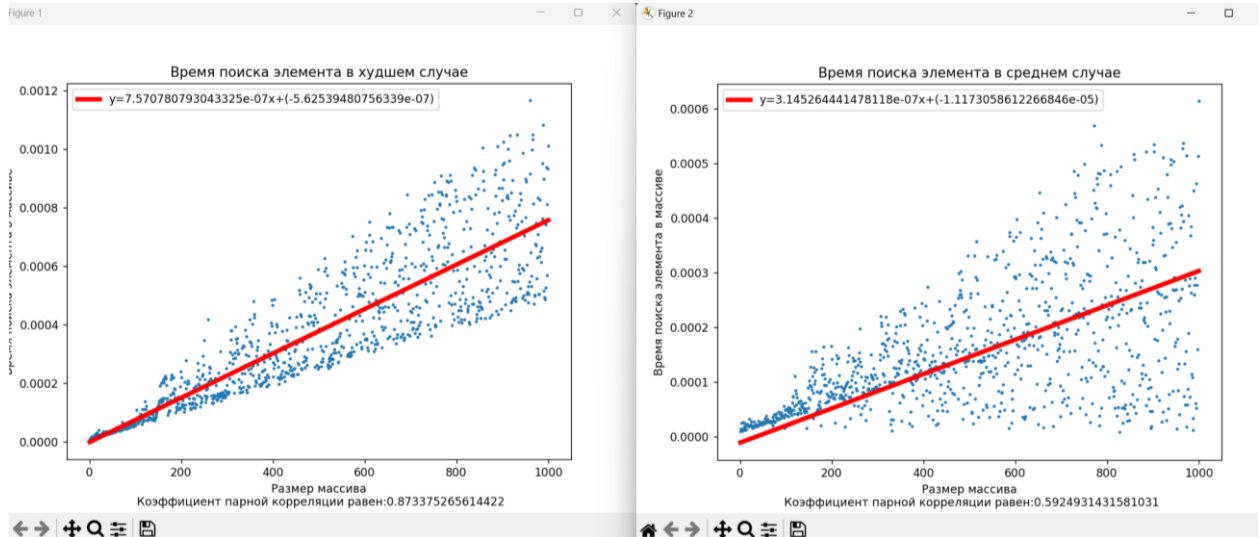


Рисунок 1 – Результат работы программы

Код программы:

```
import random
import numpy as np
import timeit
from statistics import correlation
import matplotlib.pyplot as plt
from math import sqrt
def corr(arrX, arrY):
    sigma1=0
    sigma2=0
    sigma3=0
    xSred=0
    ySred=0
    sumX=0
    sumY=0
    for i in range(len(arrX)):
        sumX+=arrX[i]
        sumY+=arrY[i]
    xSred=sumX/len(arrX)
    ySred=sumY/len(arrY)
    for i in range(len(arrX)):
        sigma1+=(arrX[i]-xSred)*(arrY[i]-ySred)
        sigma2+=(arrX[i]-xSred)**2
        sigma3+=(arrY[i]-ySred)**2
    return sigma1/(sqrt(sigma2)*sqrt(sigma3))
def poisk(arr, element):
    zn=0
    for i in range(0, len(arr)):
        if arr[i]==element:
            zn=1
            return 1
            break
    if zn==0:
        return 0
zn=0
```

```

arrTime=[]
x=[]
arrTimeBad=[]
for i in range(1,1001):
    arr2=[0 for i in range(0,i)]
    x.append(i)
    for j in range(0,len(arr2)):
        arr2[j]= random.randint(500,1000)
    poisk(arr2,1)
    timePoisk=(timeit.timeit(lambda: poisk(arr2,1), number=50))/50
    print()
    print("Время поиска в массиве из ",i," элементов в худшем случае: ",timePoisk)
    arrTime.append(timePoisk)
    timePoisk=(timeit.timeit(lambda: poisk(arr2,arr2[round(len(arr2)/2)]),
number=50))/50
    print()
    print("Время поиска в массиве из ",i," элементов в среднем случае:
",timePoisk)
    arrTimeBad.append(timePoisk)
    print()
sumArrTime=sum(arrTime)
sumX=sum(x)
kvSumX=0
sumUmnXY=0
bn=5
for i in x:
    kvSumX+=i*i
for i in range(0,len(arrTime)):
    sumUmnXY+=x[i]*arrTime[i]
matrix = np.array([[kvSumX, sumX],
                    [sumX, bn]])
det = np.linalg.det(matrix)
matrix1Kramer = np.array([[sumUmnXY, sumX],
                           [sumArrTime, bn]])
det1=np.linalg.det(matrix1Kramer)
matrix2Kramer = np.array([[kvSumX, sumUmnXY],
                           [sumX, sumArrTime]])
det2=np.linalg.det(matrix2Kramer)
koef1=det1/det
koef2=det2/det
func=[]
for i in range(1,1001):
    sumArrTimeBad=sum(arrTimeBad)
    sumUmnXYBad=0
    for i in range(0,len(arrTime)):
        sumUmnXYBad+=x[i]*arrTimeBad[i]
    matrixBad = np.array([[kvSumX, sumX],
                           [sumX, bn]])
    detBad = np.linalg.det(matrix)
    matrix1KramerBad = np.array([[sumUmnXYBad, sumX],
                                  [sumArrTimeBad, bn]])
    det1Bad=np.linalg.det(matrix1KramerBad)
    matrix2KramerBad = np.array([[kvSumX, sumUmnXYBad],
                                  [sumX, sumArrTimeBad]])
    det2Bad=np.linalg.det(matrix2KramerBad)
    koef1Bad=det1Bad/detBad
    koef2Bad=det2Bad/detBad
    funcBad=[]
    for i in range(1,1001):
        funcBad.append(koef1Bad*(i)+koef2Bad)
plt.figure(figsize=(8,6))
plt.figure(1)
plt.title("Время поиска элемента в худшем случае")
plt.plot(x,func,color='red',linewidth=4)
plt.scatter(x, arrTime,s=3)
plt.xlabel('Размер массива\n Коэффициент парной корреляции
равен: '+str(corr(x,arrTime)))

```

```

plt.legend(['y='+str(koef1)+"x+("+str(koef2)+") "])
plt.ylabel("Время поиска элемента в массиве")
plt.figure(figsize=(8,6))
plt.figure(2)
plt.title("Время поиска элемента в среднем случае")
plt.plot(x,funcBad,color='red',linewidth=4)
plt.scatter(x,arrTimeBad,s=3)
plt.xlabel('Размер массива\n Коэффициент парной корреляции  
равен: '+str(corr(x,arrTimeBad)))
plt.legend(['y='+str(koef1Bad)+"x+("+str(koef2Bad)+") "])
plt.ylabel("Время поиска элемента в массиве")
plt.show()

```

Вывод

Время поиска элемента в массиве увеличивается линейно в худшем и среднем случаях при увеличении размера массива, если используется алгоритм линейного поиска. Коэффициент парной корреляции позволяет определить связь между величинами, а метод наименьших квадратов позволяет построить график зависимости на основе экспериментальных данных.