

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 №4**  
**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: Алгоритмы поиска наименьшего и наибольшего элемента в массиве

Цель: Научиться применять метод наименьших квадратов и находить коэффициент парной корреляции, исследовать алгоритм поиска наименьшего или наибольшего элемента в массиве.

Порядок выполнения работы:

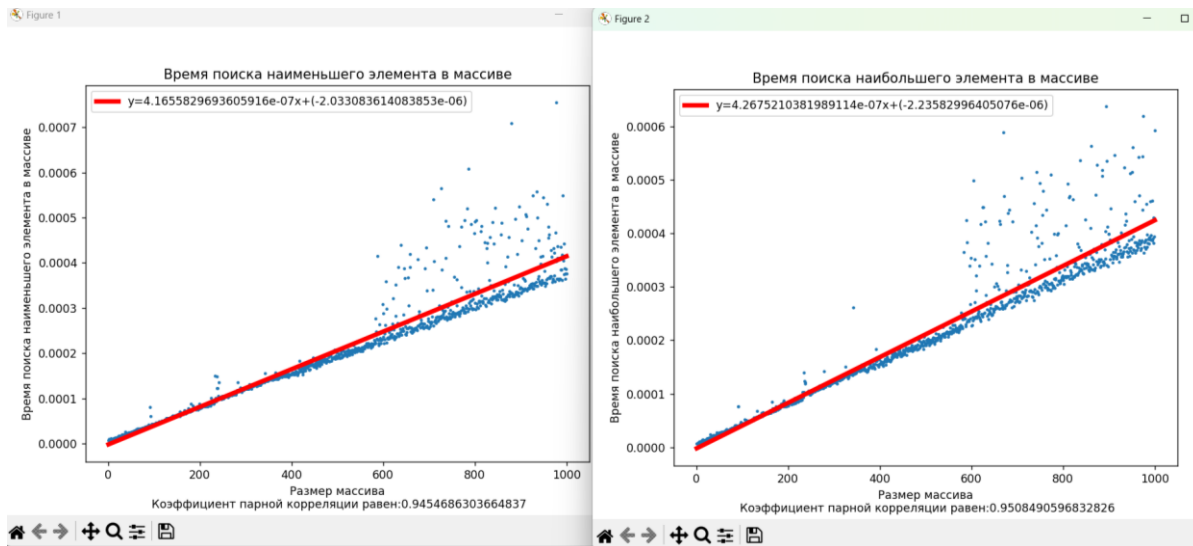


Рисунок 1 – Результат работы программы

Код программы:

```
import random
import numpy as np
import timeit
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 minElement(arr):
    temp=arr[0]
    for i in range(0, len(arr)):
        if arr[i]<temp:
            temp=arr[i]
    return temp
def maxElement(arr):
```

```

temp=arr[0]
for i in range(0,len(arr)):
    if arr[i]>temp:
        temp=arr[i]
return temp
arrTimeMin=[]
x=[]
arrTimeMax=[]
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)
    timePoisk=(timeit.timeit(lambda: minElement(arr2), number=50))/50
    print()
    print("Время поиска наименьшего элемента в массиве из ",i," элементов:
",timePoisk)
    arrTimeMin.append(timePoisk)
    timePoisk=(timeit.timeit(lambda: maxElement(arr2), number=50))/50
    print()
    print("Время поиска наибольшего элемента в массиве из ",i," элементов:
",timePoisk)
    arrTimeMax.append(timePoisk)
    print()
sumArrTime=sum(arrTimeMin)
sumX=sum(x)
kvSumX=0
sumUmnXY=0
bn=len(x)
for i in x:
    kvSumX+=i*i
for i in range(0,len(arrTimeMin)):
    sumUmnXY+=x[i]*arrTimeMin[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):
    func.append(koef1*(i)+koef2)
sumArrTimeMax=sum(arrTimeMax)
sumUmnXYMax=0
for i in range(0,len(arrTimeMax)):
    sumUmnXYMax+=x[i]*arrTimeMax[i]
matrixMax = np.array([[kvSumX, sumX],
                      [sumX, bn]])
detMax = np.linalg.det(matrix)
matrix1KramerMax = np.array([[sumUmnXYMax, sumX],
                              [sumArrTimeMax, bn]])
det1Max=np.linalg.det(matrix1KramerMax)
matrix2KramerMax = np.array([[kvSumX, sumUmnXYMax],
                              [sumX, sumArrTimeMax]])
det2Max=np.linalg.det(matrix2KramerMax)
koef1Max=det1Max/detMax
koef2Max=det2Max/detMax
funcMax=[]
for i in range(1,1001):
    funcMax.append(koef1Max*(i)+koef2Max)
plt.figure(figsize=(8,6))
plt.figure(1)

```

```

plt.title("Время поиска наименьшего элемента в массиве")
plt.plot(x, func, color='red', linewidth=4)
plt.scatter(x, arrTimeMin, s=3)
plt.xlabel('Размер массива\n Коэффициент парной корреляции
равен: '+str(corr(x, arrTimeMin)))
plt.legend(['y='+str(koef1)+"x+("+str(koef2)+")"])
plt.ylabel("Время поиска наименьшего элемента в массиве")
plt.figure(figsize=(8, 6))
plt.figure(2)
plt.title("Время поиска наибольшего элемента в массиве")
plt.plot(x, funcMax, color='red', linewidth=4)
plt.scatter(x, arrTimeMax, s=3)
plt.xlabel('Размер массива\n Коэффициент парной корреляции
равен: '+str(corr(x, arrTimeMax)))
plt.legend(['y='+str(koef1Max)+"x+("+str(koef2Max)+")"])
plt.ylabel("Время поиска наибольшего элемента в массиве")
plt.show()

```

## Вывод

Время поиска наименьшего и наибольшего элемента в массиве увеличивается линейно при увеличении размера массива. Коэффициент парной корреляции позволяет определить связь между величинами, а метод наименьших квадратов позволяет построить график зависимости на основе экспериментальных данных.