Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Программное обеспечение ЭВМ и информационные технологии**

Лабораторная работа №4.

«Построение и программная реализация алгоритма наилучшего среднеквадратичного приближения»

Студент: **Ивахненко Д. А**

Группа: **ИУ7-46Б**

Оценка (баллы): ___________

Преподаватель: **Градов В.М.**

Москва

2021

Цель работы

Получение навыков построения алгоритма метода наименьших квадратов с использованием полинома заданной степени при аппроксимации табличных функций с весами

Исходные данные

1. Таблица функции **с весами** ρ_i с количеством узлов N.

x	y	ρ_i

2. Степень аппроксимирующего полинома – n.

Описание алгоритма

Для применения метода наименьших квадратов в случае аппроксимации полиномом следует действовать следующим образом.

1. Выбирается степень полинома $n \ll N$. Обычно степень полинома не превышает $5-6$.
2. Составляется система линейных алгебраических уравнений типа:

$$\sum_{m=0}^{n}(x^k, x^m)\, a_m = (y, x^k)\ ,\ 0 \le k \le n\ ,$$

где $(x^k, x^m) = \sum_{i=1}^{N} \rho_i x_i^{k+m}, \quad (y, x^k) = \sum_{i=1}^{N} \rho_i y_i x_i^k\ .$

3. В результате решения СЛАУ находятся коэффициенты полинома a_k.

Исходный код программы

main.py

```python
def main():
    file = input('Выберите таблицу: [1/2/3/4]\n')
    x, y, ro = read_table(PATH_TO_TABLE + file + '.txt')

    all_n = list(map(int, input('Введите степени полинома: ').
split()))

    print(colored('===========================', 'yellow'))
    print(colored('\tSOURCE TABLE', 'yellow'))
    print(colored('===========================', 'yellow'))


    all_n = [n for n in all_n]

    print_table(x, y, ro)


    print(colored('===========================', 'yellow'))
    print(colored('\tMATRICIES', 'yellow'))
    print(colored('===========================', 'yellow'))


    all_a = [root_mean_square(x, y, ro, n+1) for n in all_n]

    result = list(zip(all_a, all_n))

    print(colored('===========================', 'green'))
    print(colored('\tRESULT', 'green'))
    print(colored('===========================', 'green'))

    for a, n in result:
        pr_a = np.array(a)
        print(f'n = {n}\na = {pr_a}')
        print(colored('===========================', 'green'))

    show(result, x, y, ro)
```

implementation.py

```
def matprint(mat, fmt=".2f"):
    col_maxes = [max([len(("{:"+fmt+"}").format(x)) for x in col
]) for col in mat.T]
    for x in mat:
        for i, y in enumerate(x):
            print(("{:"+str(col_maxes[i])+fmt+"}").format(y),
end="  ")
        print("")

def f(x_arr, coeff):
    res = np.zeros(len(x_arr))
    for i in range(len(coeff)):
        res += coeff[i]*(x_arr**i)
    return res

def read_table(filename):
    f = open(filename, "r")
    x, y, ro = [], [], []
    for line in f:
        line = line.split(" ")
        x.append(float(line[0]))
        y.append(float(line[1]))
        ro.append(float(line[2]))
    return x, y, ro

def print_table(x, y, ro):
    length = len(x)
    print("x       y       ro")
    for i in range(length):
        print("%.4f %.4f %.4f" % (x[i], y[i], ro[i]))
    print()
```

```
def show(result, all_x, all_y, all_ro):
    t = np.arange(min(all_x), max(all_x)+0.2, 0.02)
    plt.title('Среднеквадратичное приближение.')
    plt.ylabel("y")
    plt.xlabel("x")

    colors = ['k', 'g', 'b', 'y', 'c']

    for a, n in result:
        color = colors.pop()
        plt.plot(t, f(t, a), color=color, label=f'n = {n}')

    for x, y, ro in zip(all_x, all_y, all_ro):
        plt.plot(x, y, 'ro', markersize=ro+5)

    plt.rc('grid', linestyle="-", color='black')
    plt.grid(True)
    plt.legend()
    plt.show()
```

```python
def root_mean_square(x, y, ro, n):
    length = len(x)
    sum_x_n = [sum([x[i]**j*ro[i] for i in range(length)]) for j
 in range(n*2 -1)]
    sum_y_x_n = [sum([x[i]**j*ro[i]*y[i] for i in range(length
)]) for j in range(n)]
    matr = [sum_x_n[i:i+n] for i in range(n)]
    for i in range(n):
        matr[i].append(sum_y_x_n[i])
    print(colored("\nSOURCE:", 'red'))
    matprint(np.array(matr))
    return calc_gauss(matr)

def calc_gauss(matr):
    n = len(matr)
    # приводим к треугольному виду
    for k in range(n):
        for i in range(k+1,n):
            coeff = -(matr[i][k]/matr[k][k])
            for j in range(k,n+1):
                matr[i][j] += coeff*matr[k][j]
    print(colored("\nTRIANGLED:", 'red'))
    matprint(np.array(matr))
    print(colored('===========================', 'yellow'))

    a = [0 for i in range(n)]
    for i in range(n-1, -1, -1):
        for j in range(n-1, i, -1):
            matr[i][n] -= a[j]*matr[i][j]
        a[i] = matr[i][n]/matr[i][i]
    return a
```

Результаты работы

В качестве результатов привожу для каждой степени полинома:

а) Исходную таблицу

б) Результат расчетов коэффициентов a_i

в) Две матрицы (начальную и треугольную)

г) График, на котором построены точки из исходной таблицы, а также кривые для выбранных степеней.

1-3 – примеры работы при одинаковых весах точек

4 – пример работы с разными весами точек

1.

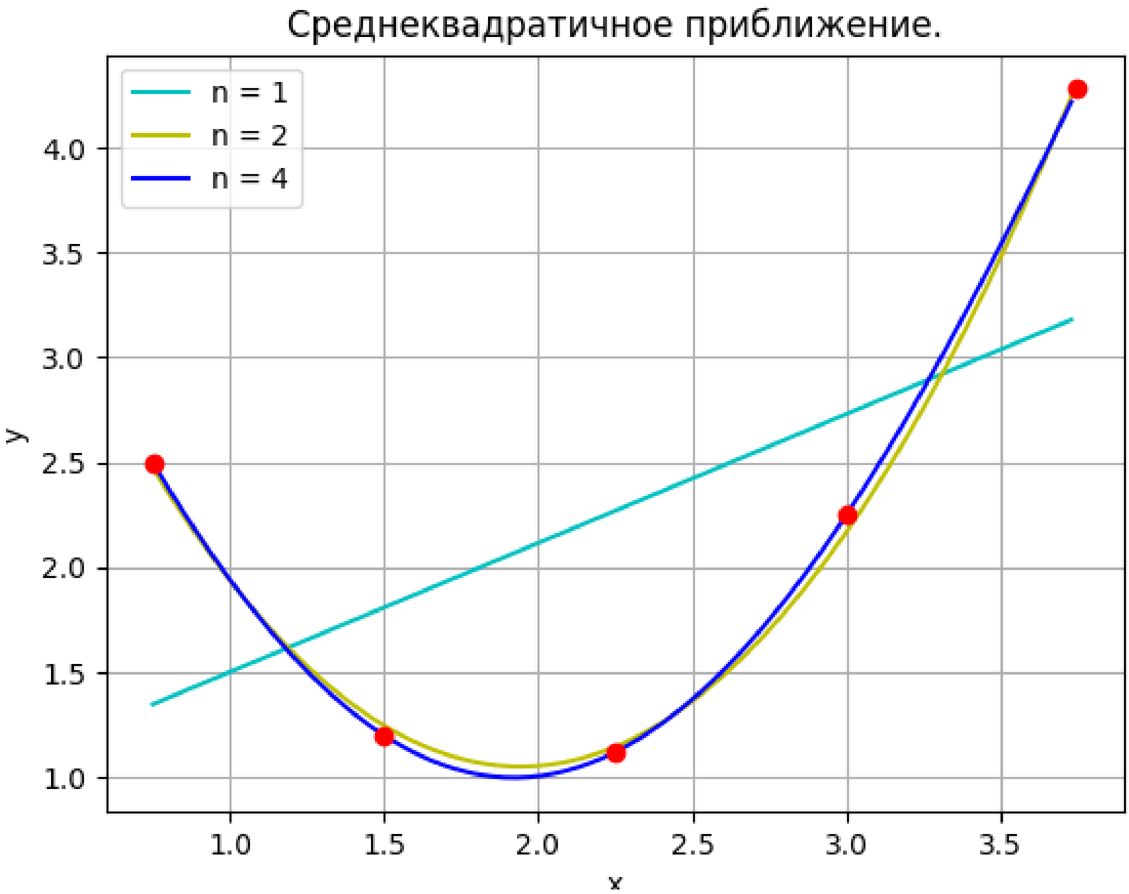
```
Введите степени полинома: 1 2 4
==============================
          SOURCE TABLE
==============================
x       y       ro
0.7500 2.5000 1.0000
1.5000 1.2000 1.0000
2.2500 1.1200 1.0000
3.0000 2.2500 1.0000
3.7500 4.2800 1.0000
```

```
============================
        RESULT
============================
n = 1
a = [0.89 0.61]
============================
n = 2
a = [4.82 -3.88 1.00]
============================
n = 4
a = [4.73 -3.36 0.32 0.29 -0.04]
============================
```

```
==============================
          MATRICIES
==============================


SOURCE:
 5.00   11.25   11.35
11.25   30.94   29.00

TRIANGLED:
5.00   11.25   11.35
0.00    5.62    3.46
==============================
```

```
==============================


SOURCE:
 5.00   11.25    30.94   11.35
11.25   30.94    94.92   29.00
30.94   94.92   309.76   90.21

TRIANGLED:
5.00   11.25   30.94   11.35
0.00    5.62   25.31    3.46
0.00    0.00    4.43    4.43
==============================
```

```
==============================


SOURCE:
  5.00     11.25     30.94      94.92     309.76     11.35
 11.25     30.94     94.92     309.76    1050.07     29.00
 30.94     94.92    309.76    1050.07    3651.23     90.21
 94.92    309.76   1050.07    3651.23   12924.58    304.32
309.76   1050.07   3651.23   12924.58   46350.18   1064.21

TRIANGLED:
5.00   11.25   30.94   94.92   309.76   11.35
0.00    5.62   25.31   96.19   353.11    3.46
0.00    0.00    4.43   29.90   145.59    4.43
0.00   -0.00    0.00    2.56    23.07   -0.16
0.00    0.00    0.00    0.00     0.82   -0.03
==============================
```



Среднеквадратичное приближение.
n = 1
n = 2
n = 4
4.0
3.5
3.0
2.5
2.0
1.5
1.0
y
1.0
1.5
2.0
2.5
3.0
3.5
x

2.

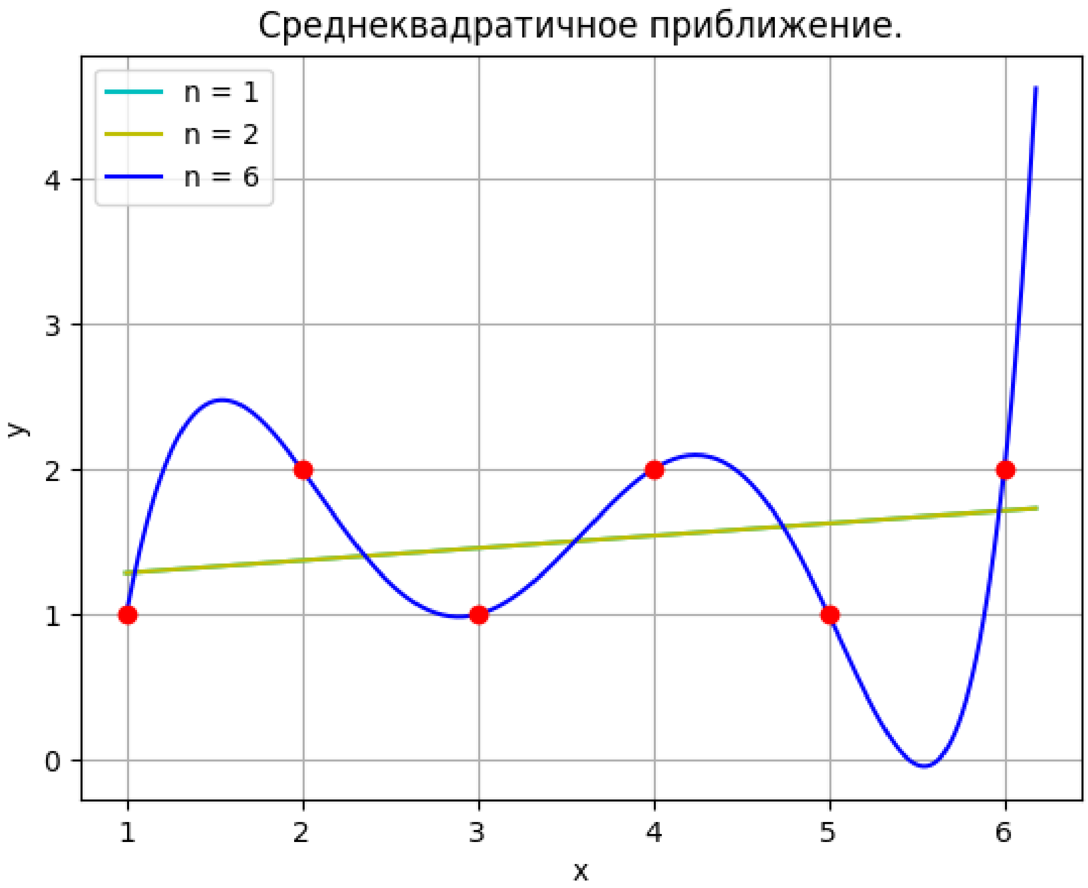
```
Введите степени полинома: 1 2 6
===========================
        SOURCE TABLE
===========================
x      y      ro
1.0000 1.0000 1.0000
2.0000 2.0000 1.0000
3.0000 1.0000 1.0000
4.0000 2.0000 1.0000
5.0000 1.0000 1.0000
6.0000 2.0000 1.0000
```

```
===========================
        RESULT
===========================
n = 1
a = [1.20 0.09]
===========================
n = 2
a = [1.20 0.09 0.00]
===========================
n = 6
a = [-16.29 30.94 -15.74 1.34 1.00 -0.27 0.02]
===========================
```

```
===========================
         MATRICIES
===========================

SOURCE:
  6.00  21.00   9.00
 21.00  91.00  33.00

TRIANGLED:
6.00  21.00  9.00
0.00  17.50  1.50
===========================
```

```
===========================

SOURCE:
  6.00   21.00    91.00    9.00
 21.00   91.00   441.00   33.00
 91.00  441.00  2275.00  147.00

TRIANGLED:
6.00  21.00   91.00  9.00
0.00  17.50  122.50  1.50
0.00   0.00   37.33  0.00
===========================
```

```
===========================

SOURCE:
    6.00      21.00       91.00       441.00      2275.00      12201.00       67171.00      9.00
   21.00      91.00      441.00      2275.00     12201.00      67171.00      376761.00     33.00
   91.00     441.00     2275.00     12201.00     67171.00     376761.00     2142595.00    147.00
  441.00    2275.00    12201.00     67171.00    376761.00    2142595.00    12313161.00    729.00
 2275.00   12201.00    67171.00    376761.00   2142595.00   12313161.00    71340451.00   3843.00
12201.00   67171.00   376761.00   2142595.00  12313161.00   71340451.00   415998681.00  21033.00
67171.00  376761.00  2142595.00  12313161.00  71340451.00  415998681.00  2438235715.00 117987.00

TRIANGLED:
6.00  21.00   91.00  441.00  2275.00  12201.00   67171.00   9.00
0.00  17.50  122.50  731.50  4238.50  24467.50  141662.50   1.50
0.00   0.00   37.33  392.00  2997.33  20440.00  132197.33   0.00
0.00   0.00    0.00   64.80   907.20   8460.00   66528.00   4.80
0.00   0.00    0.00    0.00    82.29   1440.00   15840.00   0.00
0.00   0.00    0.00    0.00     0.00     57.14    1200.00   7.62
0.00   0.00    0.00    0.00     0.00      0.00      -0.00  -0.00
===========================
```


Среднеквадратичное приближение.
n = 1
n = 2
n = 6
y
x
0
1
2
3
4
1
2
3
4
5
6

3.

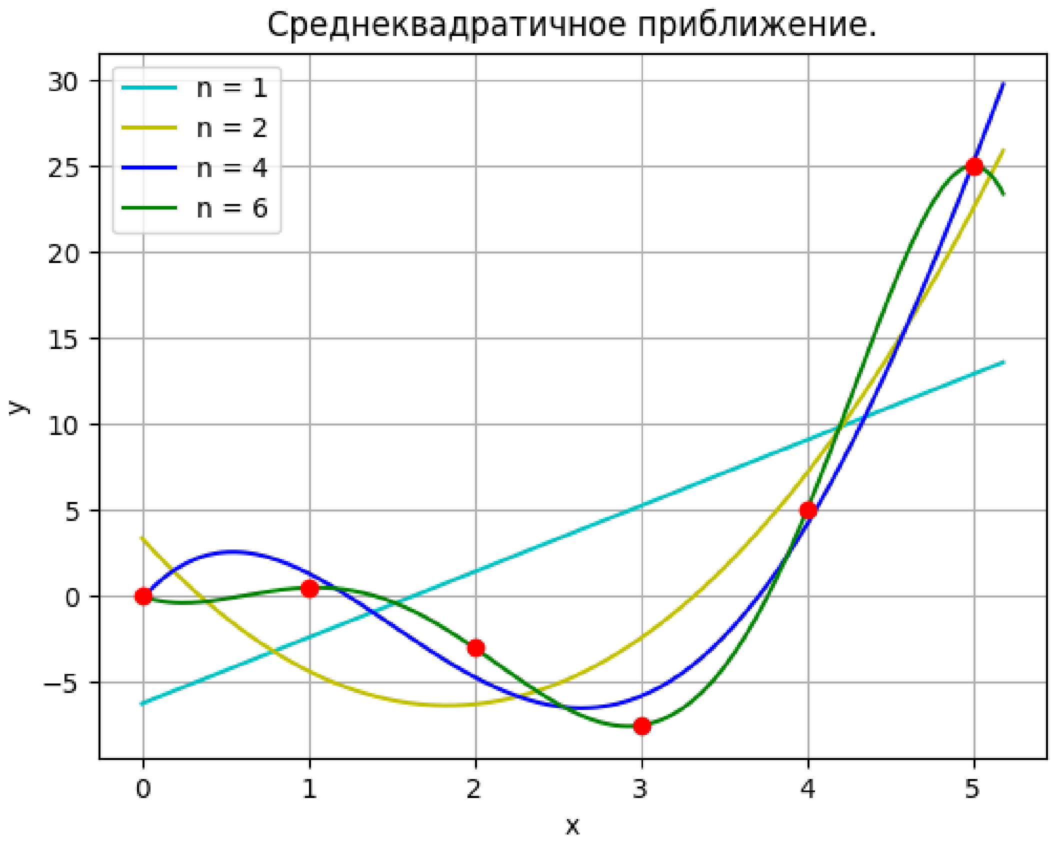
```
Введите степени полинома: 1 2 4 6
===========================
        SOURCE TABLE
===========================
x       y       ro
0.0000 0.0000 1.0000
1.0000 0.5000 1.0000
2.0000 -3.0000 1.0000
3.0000 -7.5000 1.0000
4.0000 5.0000 1.0000
5.0000 25.0000 1.0000
```

```
===========================
        RESULT
===========================
n = 1
a = [-6.24 3.83]
===========================
n = 2
a = [3.37 -10.59 2.88]
===========================
n = 4
a = [-0.17 10.99 -12.86 3.64 -0.26]
===========================
n = 6
a = [0.00 -3.44 8.97 -5.70 0.41 0.31 -0.04]
===========================
```

```
==============================
        MATRICIES
==============================

SOURCE:
 6.00   15.00    20.00
15.00   55.00   117.00

TRIANGLED:
6.00   15.00   20.00
0.00   17.50   67.00
==============================
```

```
==============================

SOURCE:
 6.00    15.00    55.00    20.00
15.00    55.00   225.00   117.00
55.00   225.00   979.00   626.00

TRIANGLED:
6.00   15.00   55.00    20.00
0.00   17.50   87.50    67.00
0.00    0.00   37.33   107.67
==============================
```

```
==============================

SOURCE:
  6.00     15.00     55.00    225.00     979.00     20.00
 15.00     55.00    225.00    979.00    4425.00    117.00
 55.00    225.00    979.00   4425.00   20515.00    626.00
225.00    979.00   4425.00  20515.00   96825.00   3219.00
979.00   4425.00  20515.00  96825.00  462979.00  16250.00

TRIANGLED:
6.00   15.00   55.00   225.00    979.00    20.00
0.00   17.50   87.50   416.50   1977.50    67.00
0.00    0.00   37.33   280.00   1653.33   107.67
0.00    0.00    0.00    64.80    648.00    66.90
0.00    0.00    0.00     0.00     82.29   -21.43
==============================
```


Среднеквадратичное приближение.
n = 1
n = 2
n = 4
n = 6
30
25
20
15
10
5
0
−5
y
0
1
2
3
4
5
x

4. Демонстрация работы с весами точек.

x	y	ρ_i (1)	ρ_i (2)	ρ_i (3)
1	1	1	1	1
2	2	1	1	4
3	4	1	4	1
7	9	1	1	1

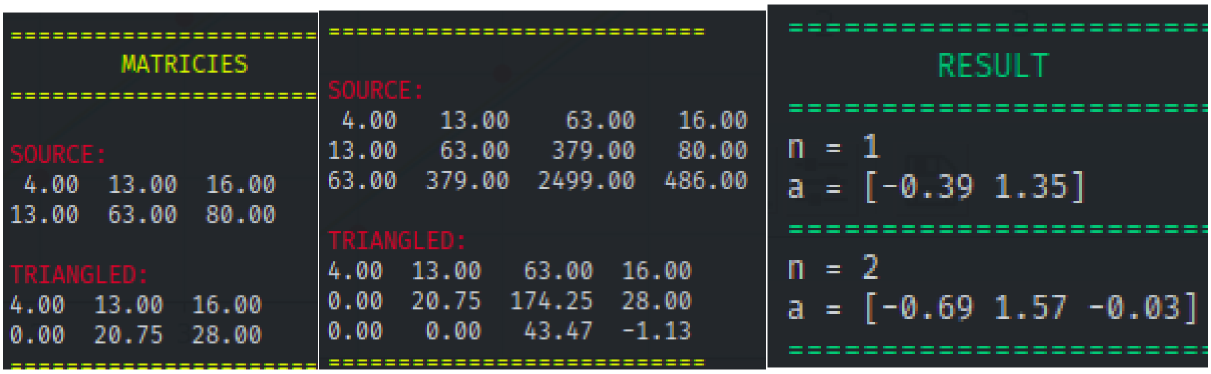
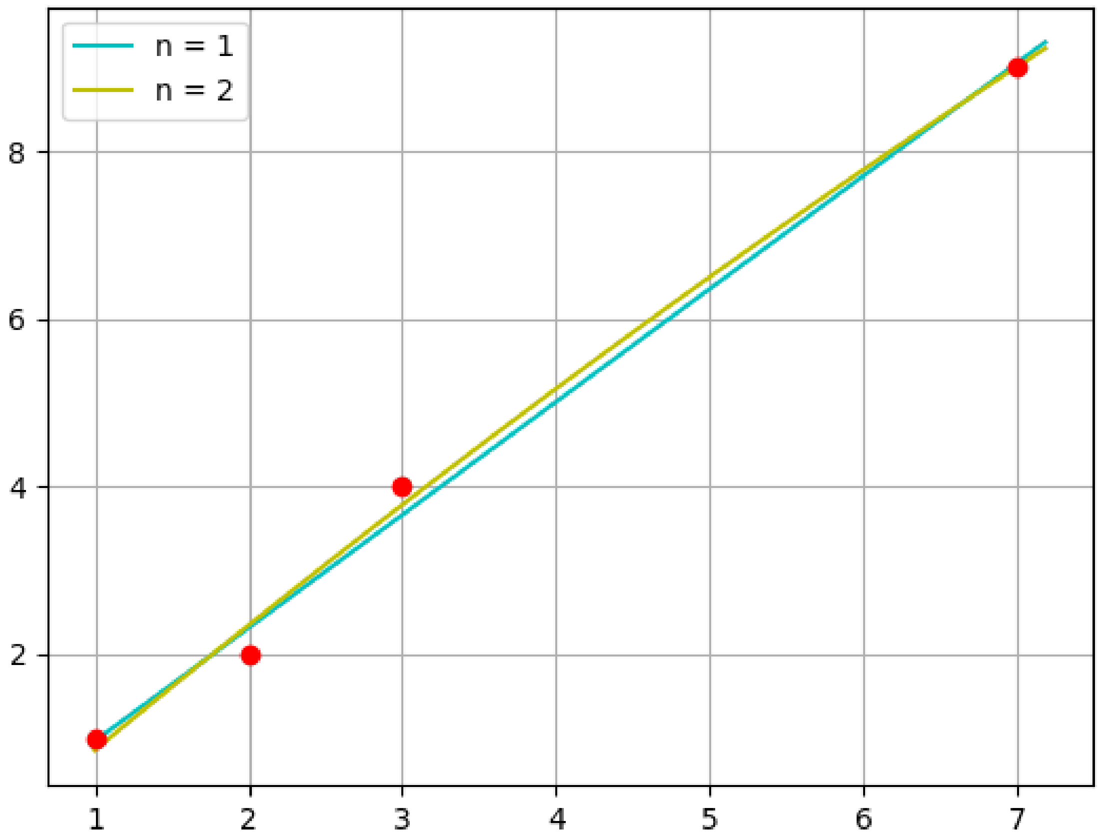
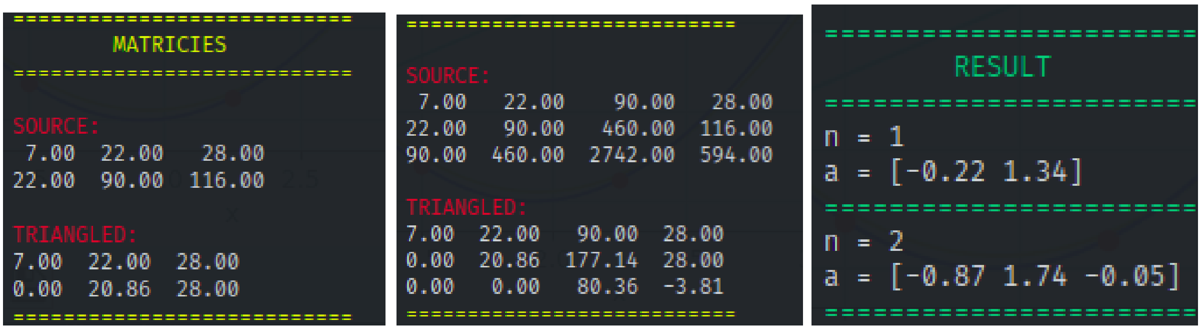


Рисунок 1. Все единицы



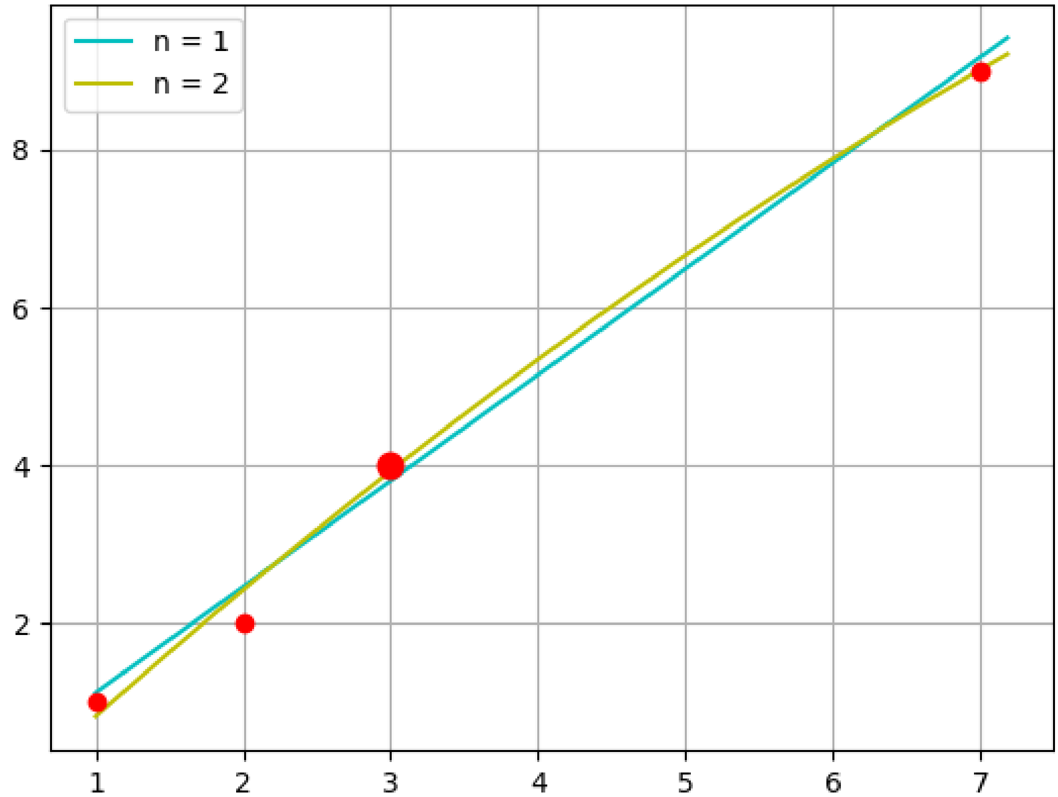


```
===========================
        MATRICIES
===========================

SOURCE:
 7.00  22.00   28.00
22.00  90.00  116.00

TRIANGLED:
7.00  22.00  28.00
0.00  20.86  28.00
===========================
```

```
==========================

SOURCE:
 7.00   22.00    90.00   28.00
22.00   90.00   460.00  116.00
90.00  460.00  2742.00  594.00

TRIANGLED:
7.00  22.00   90.00  28.00
0.00  20.86  177.14  28.00
0.00   0.00   80.36  -3.81
==========================
```

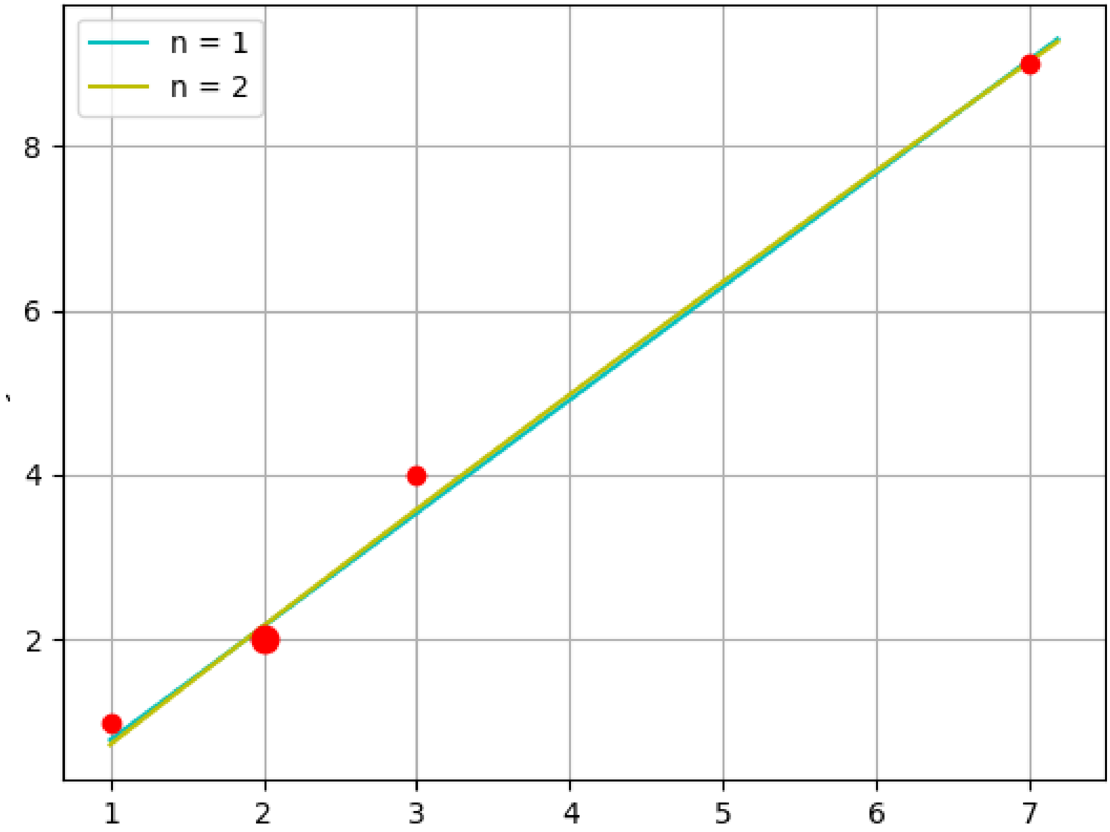
```
=========================
         RESULT
=========================
n = 1
a = [-0.22 1.34]
=========================
n = 2
a = [-0.87 1.74 -0.05]
=========================
```

Рисунок 2. Точка (3, 4) (укрупнена) имеет больший вес



```
=====================        ==========================        =========================
      MATRICIES               SOURCE:                                   RESULT
=====================         7.00   19.00    75.00   22.00    =========================
                             19.00   75.00   403.00   92.00    n = 1
SOURCE:                      75.00  403.00  2547.00  510.00    a = [-0.60 1.38]
 7.00  19.00  22.00                                            =========================
19.00  75.00  92.00          TRIANGLED:                        n = 2
                             7.00  19.00   75.00  22.00        a = [-0.74 1.48 -0.01]
TRIANGLED:                   0.00  23.43  199.43  32.29        =========================
7.00  19.00  22.00           0.00   0.00   45.85  -0.54
0.00  23.43  32.29           ==========================
=====================
```

Рисунок 3. Точка (2, 2) (укрупнена) имеет больший вес



Ответы на контрольные вопросы

1. Что произойдет при задании степени полинома n=N-1 (числу узлов таблицы минус 1)

Поскольку для однозначного определения полинома N − 1 степени достаточно N точек, полином будет построен таким образом, что его график будет проходить через все табличные точки.

При такой конфигурации в выражении

$$\sum_{i=1}^{N} \rho_i [y(x_i) - \varphi(x_i)]^2 = min$$

часть выражения, находящаяся в скобках, обращается в 0, а значит зависимости от весов нет. (при любых заданных весах, значение полинома будет минимальным в случае прохода через табличные точки).

2. Будет ли работать Ваша программа при $n \geq N$? Что именно в алгоритме требует отдельного анализа данного случая и может привести к аварийной остановке?

Программа работать будет, но некорректно, т. к. начиная $n = N$ определитель СЛАУ, которую необходимо решить, будет тождественно равен 0. Из-за применения метода Гаусса-Жордана, аварийная остановка программы (деление на ноль) может произойти при приведении диагональной матрицы к единичной. Анализ можно проводить при решении СЛАУ или же на начальном этапе (ввод степени полинома)

3. Получить формулу для коэффициента полинома a_0 при степени полинома $n = 0$. Какой смысл имеет величина, которую представляет данный коэффициент?

Полученная формула:

$$a_0 = \frac{\sum_{i=1}^{N} \rho_i y_i}{\sum_{i=1}^{N} \rho_i}$$

Данную формулу можно преобразовать делением числителя и знаменателя на сумму весов, из чего получим математическое ожидание:

$$M[X] = \sum_{i=1}^{\infty} x_i \rho_i$$

4. Записать и вычислить определитель матрицы СЛАУ для нахождения коэффициентов полинома для случая, когда $n = N = 2$. Принять все веса = 1.

x_i	y_i	ρ_i
x_0	y_0	1
x_1	y_1	1

СЛАУ:

$$\begin{cases} a_0 + (x_0 + x_1)a_1 + (x_0^2 + x_1^2)a_2 = y_0 + y_1 \\ (x_0 + x_1)a_0 + (x_0^2 + x_1^2)a_1 + (x_0^3 + x_1^3)a_2 = y_0 x_0 + y_1 x_1 \\ (x_0^2 + x_1^2)a_0 + (x_0^3 + x_1^3)a_1 + (x_0^4 + x_1^4)a_2 = y_0 x_0^2 + y_1 x_0^2 \end{cases}$$

$$\Delta = (x_0^2 + x_1^2)(x_0^4 + x_1^4) + (x_0 + x_1)(x_0^3 + x_1^3)(x_0^2 + x_1^2) +$$
$$(x_0^2 + x_1^2)(x_0 + x_1)(x_0^3 + x_1^3) - (x_0^2 + x_1^2)(x_0^2 + x_1^2)(x_0^2 + x_1^2) -$$
$$(x_0^3 + x_1^3)(x_0^3 + x_1^3) - (x_0 + x_1)(x_0 + x_1)(x_0^4 + x_1^4) = 0$$

$\Delta = 0 \Rightarrow$ система решений не имеет (см. вопрос 2)

5. Построить СЛАУ при выборочном задании степеней аргумента полинома $\varphi(x) = a_0 + a_1 x^m + a_2 x^n$ причем степени n и m в этой формуле известны

$$\begin{cases} (x^0,\, x^0)a_0 + (x^0, x^m)a_1 + (x^0, x^n)a_2 = (y, x^0) \\ (x^m,\, x^0)a_0 + (x^m, x^m)a_1 + (x^m, x^n)a_2 = (y, x^m) \\ (x^n,\, x^0)a_0 + (x^n, x^m)a_1 + (x^n, x^n)a_2 = (y, x^n) \end{cases}$$

6. Предложить схему алгоритма решения задачи из вопроса 5, если степени n и m подлежат определению наравне с коэффициентами a_k , т.е. количество неизвестных равно 5

1. Перебираем все возможные пары n и m.

2. Для каждой пары ищем все коэффициенты a_i, а также ошибку.

3. Среди всех таких наборов выбираем тот, у которого ошибка будет наименьшей