

# Технология измерений

---

«Хронометраж»

```
# find maximum value in a list
def find_max(nums):
    return max(nums)

import time

# choose n to get t = 0.1 ... 1.0 sec
n = 4_000_000
# nums = n * [1] # best case
nums = range(n) # worst case
t = time.perf_counter()
x_max = find_max(nums)
dt = time.perf_counter() - t
print(n, dt)
```

# Подбираем размер задачи (от 0.1 до 1 сек)

```
C:\11>python maxx.py  
4000000 1.1377638000003572
```

Худший случай

```
C:\11>python maxx.py  
4000000 0.10536360000014611
```

```
C:\11>python maxx.py  
4000000 0.053779100000610924
```

Лучший случай

```
C:\11>python maxx.py  
4000000 0.5737364999995407
```

# Подбираем n

```
C:\00\bin\x64\Release\net9.0>00  
1000000000 0,5927914
```

```
C:\00\bin\x64\Release\net9.0>00  
2000000000 0,1241308
```

```
C:\00\bin\x64\Release\net9.0>00  
2000000000 0,1226144
```

```
C:\00\bin\x64\Release\net9.0>00  
1000000000 0,5975981
```

```
using System;  
using System.Diagnostics;  
using System.Linq;  
  
class Program  
{  
    // find maximum in array (built-in)  
    static int FindMax(int[] nums)  
    {  
        return nums.Max();  
    }  
  
    static void Main(string[] args)  
    {  
        // Choose array size (0.1 ... 1 sec)  
        int n = 200_000_000;  
        int[] nums = new int[n];  
        for (int i = 0; i < n; i++)  
            // nums[i] = 1;    // Best case  
            nums[i] = i;      // Worst case  
  
        Stopwatch stopwatch = Stopwatch.StartNew();  
        int x_max = FindMax(nums);  
        stopwatch.Stop();  
        double t = stopwatch.Elapsed.TotalSeconds;  
  
        Console.WriteLine($"{n} {t}");  
    }  
}
```

# Организуем цикл

- Внешний цикл - размер задачи  $n$ 
  - 5 значений = 5 точек на графике
  - от 8\_000\_000 до 40\_000\_000 шаг 8\_000\_000
- Внутренний цикл – повторение опытов
  - 10 прогонов
  - «Усреднение» – удалить выбросы и случайный разброс



# Py: EN/RU

- Проверяем работоспособность
- [Ctrl + C]

```
C:\11>python maxx_csv.py
n;T(n)
8000000;0,2113006000
8000000;0,2074640000
8000000;0,2112839000
8000000;0,2089632000
8000000;0,2089441000
8000000;0,2096931000
8000000;0,2148051000
8000000;0,2087375000
8000000;0,2146122000
8000000;0,2180566000
```

# Перенаправляем вывод в файл

- Перенаправляем стандартный вывод в файл – ждем завершения

```
C:\11>python maxx_csv.py > maxx_py_worst.csv  
C:\11>_
```

- Стандартный вывод на экран и в файл – наблюдаем работу
  - «Т-образное соединение»

```
C:\11>python maxx_csv.py | tee maxx_py_worst.csv  
n;T(n)  
8000000;0,2171052000  
8000000;0,2149020000  
8000000;0,2277043000  
8000000;0,2214901000
```

# C#: EN/RU

```
using System;
using System.Diagnostics;
using System.Linq;

class Program
{
    // find maximum in array (built-in)
    static int FindMax(int[] nums)
    {
        return nums.Max();
    }
    static void Main()
    {
        Console.WriteLine("n;T(n)");
        for (int n = 200_000_000; n <= 1_000_000_000; n += 200_000_000)
        {
            int[] nums = new int[n];
            for (int i = 0; i < n; i++)
                nums[i] = i; // Worst: i    Best: 1
            for (int i = 0; i < 10; i++)
            {
                Stopwatch stopwatch = Stopwatch.StartNew();
                int x_max = FindMax(nums);
                stopwatch.Stop();
                double t = stopwatch.Elapsed.TotalSeconds;
                Console.WriteLine($"{n};{t:F10}".Replace(',', ' '));
            }
        }
    }
}
```

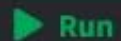


## C#: Вывод на экран и в файл CSV

```
C:\00\bin\x64\Release\net9.0>n;T(n)  
2000000000;0.1276321000  
2000000000;0.1185067000
```

```
C:\00\bin\x64\Release\net9.0> > maxx_cs.csv
```

```
C:\00\bin\x64\Release\net9.0> | tee maxx_cs_worst.csv  
n;T(n)  
2000000000;0.1233352000  
2000000000;0.1205683000
```



Run



Share

Command Line Arguments



40000000 0.182827389



Download



\*\* Process exited - Return Code: 0 \*\*

Maxx.java +

```
import java.util.Arrays;

public class Maxx {
    // Find maximum in array (built-in)
    static int findMax(int[] nums) {
        return Arrays.stream(nums).max().orElse(Integer.MIN_VALUE);
    }

    public static void main(String[] args) {
        int n = 40_000_000;
        int[] nums = new int[n];
        for (int i = 0; i < n; i++) {
            nums[i] = 1; // Best: 1    Worst: i
        }

        long t0 = System.nanoTime();
        int x_max = findMax(nums);
        long t1 = System.nanoTime();
        double t = (t1 - t0) / 1_000_000_000.0;
        System.out.println(n + " " + t);
    }
}
```

# Открываем CSV в Excel

- Worst.csv
- Best.csv
- Текст прижат влево
- Числа прижаты вправо

B2		✕ ✓ $f_x$		0,2171052	
	A	B	C		
1	n	T(n)			
2	8000000	0,2171052			
3	8000000	0,214902			
4	8000000	0,2277043			
5	8000000	0,2214901			
6	8000000	0,2222563			

# Проблемы загрузки

- Разделитель полей (столбцов)

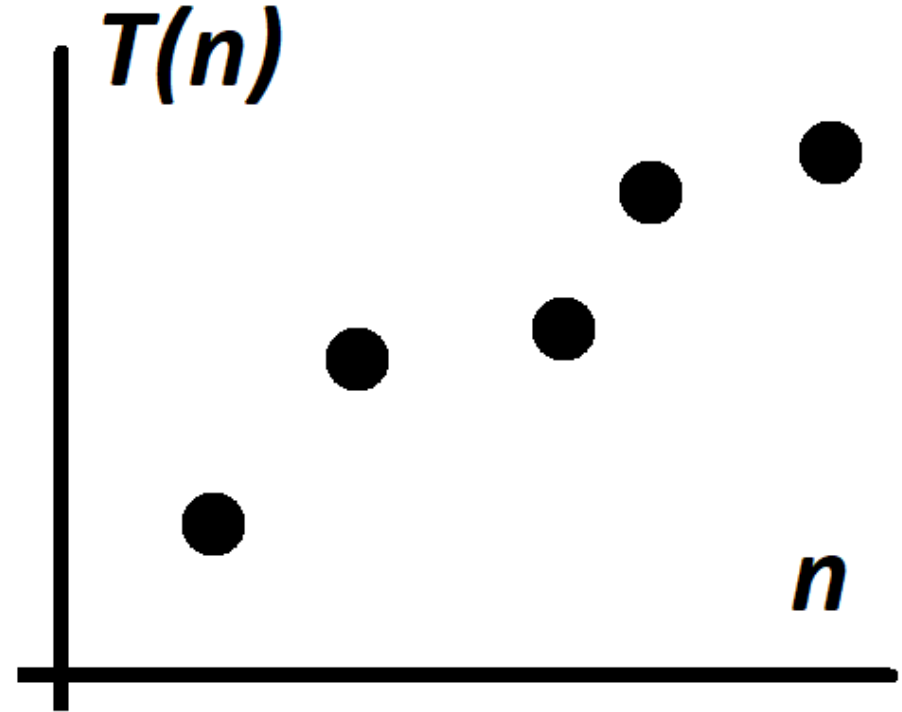
A2    ✕    ✓ $f_x$ 8000000,0.1075582000						
	A	B	C	D	E	F
1	n,T(n)					
2	8000000,0.1075582000					
3	8000000,0.1080755000					
4	8000000,0.1123634000					
5	8000000,0.1098603000					

- Десятичный разделитель

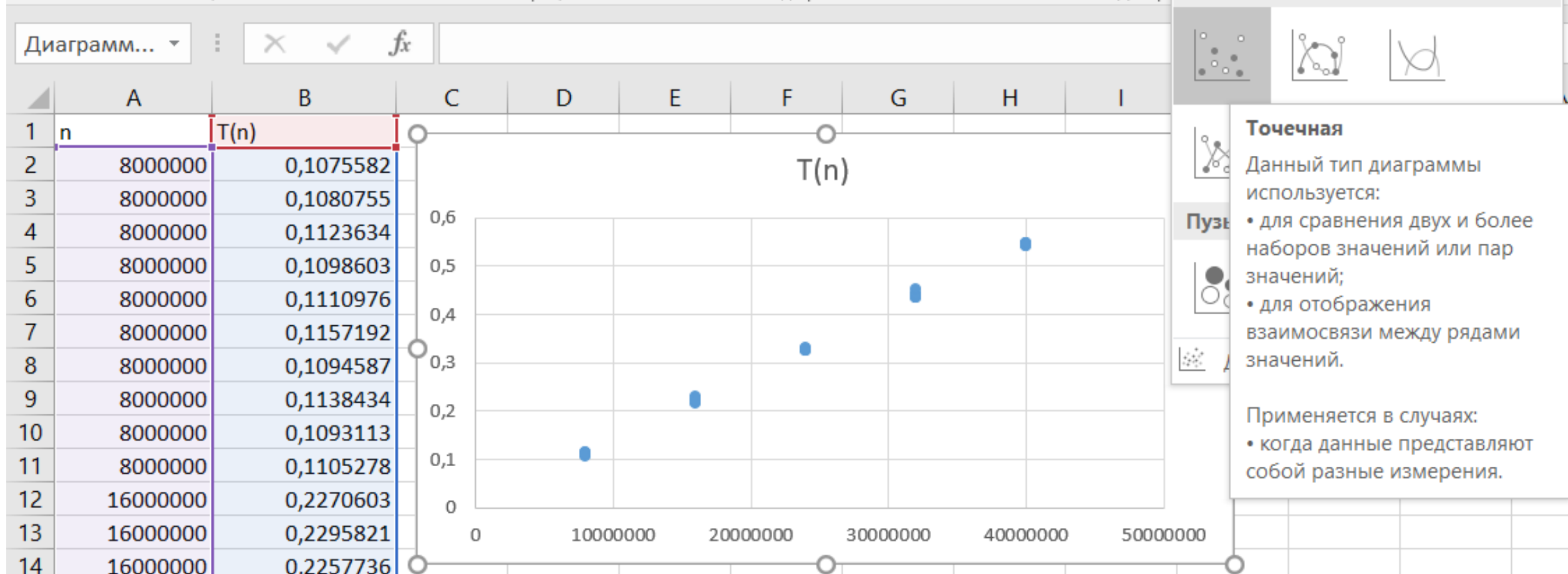
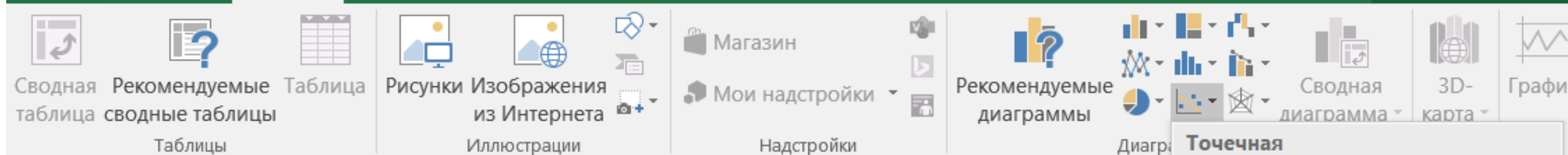
B2    ✕    ✓ $f_x$ 0.1075582000				
	A	B	C	D
1	n	T(n)		
2	8000000	0.1075582000		
3	8000000	0.1080755000		
4	8000000	0.1123634000		
5	8000000	0.1098603000		

# Диаграмма

- Строим график
  - Диаграмма разброса
  - Диаграмма рассеяния
  - Корреляционное поле
  - Поле корреляции
  - «Диаграмма X – Y»
  - Scatter Plot
- Выделяем столбцы
- Вставка – Диаграмма – Точечная – Точечная

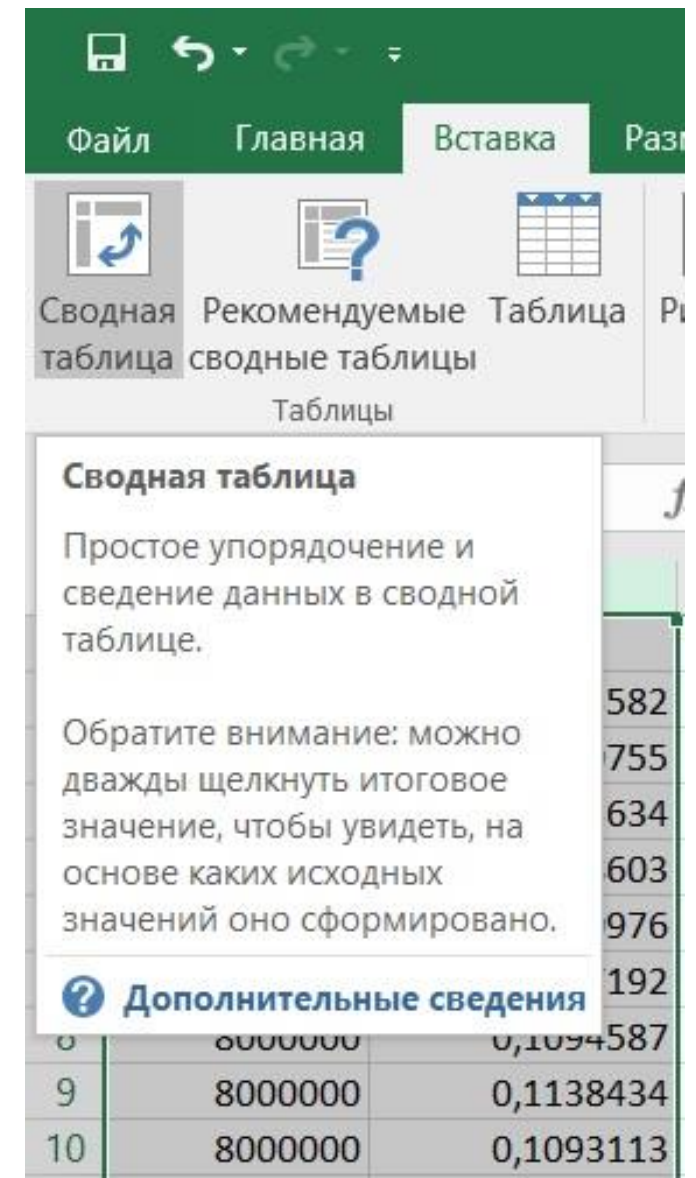






# Сводная таблица

- Выделяем диапазон
- Таблица с заголовками
  - [Ctrl + Home]
  - [Shift + Right]
  - [Ctrl + Shift + Down]
- Вставка – Сводная таблица
- Insert – Pivot Table



# Адрес ячейки для вывода

	A	B	C	D	E	F	G	H	I	J	K
1	n	T(n)									
2	8000000	0,1075582									
3	8000000	0,1080755									
4	8000000	0,1123634									
5	8000000	0,1098603									
6	8000000	0,1110976									
7	8000000	0,1157192									
8	8000000	0,1094587									
9	8000000	0,1138434									
10	8000000	0,1093113									
11	8000000	0,1105278									
12	16000000	0,2270603									
13	16000000	0,2295821									
14	16000000	0,2257736									
15	16000000	0,2170472									
16	16000000	0,2190196									
17	16000000	0,2182475									
18	16000000	0,2186502									
19	16000000	0,2185358									

Создание сводной таблицы

Выберите данные для анализа:

☒ Выбрать таблицу или диапазон

Таблица или диапазон: maxx\_py\_best!\$A\$1:\$B\$51

☐ Использовать внешний источник данных

Выбрать подключение...

Имя подключения:

☐ Использовать модель данных этой книги

Укажите, куда следует поместить отчет сводной таблицы:

☐ На новый лист

☒ На существующий лист

Диапазон: maxx\_py\_best!\$K\$2

Укажите, следует ли проанализировать несколько таблиц:

☐ Добавить эти данные в модель данных

OK Отмена

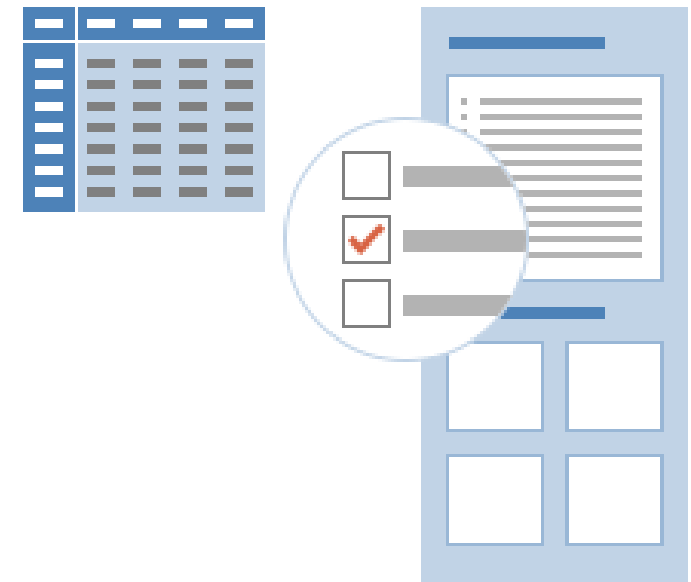
# Настроим сводную таблицу

- Строки: n
- Значения: T
- Минимальное  $T(n)$

Названия строк	Минимум по полю T(n)
80000000	0,1075582
160000000	0,2170472
240000000	0,3240958
320000000	0,4340132
400000000	0,5415516

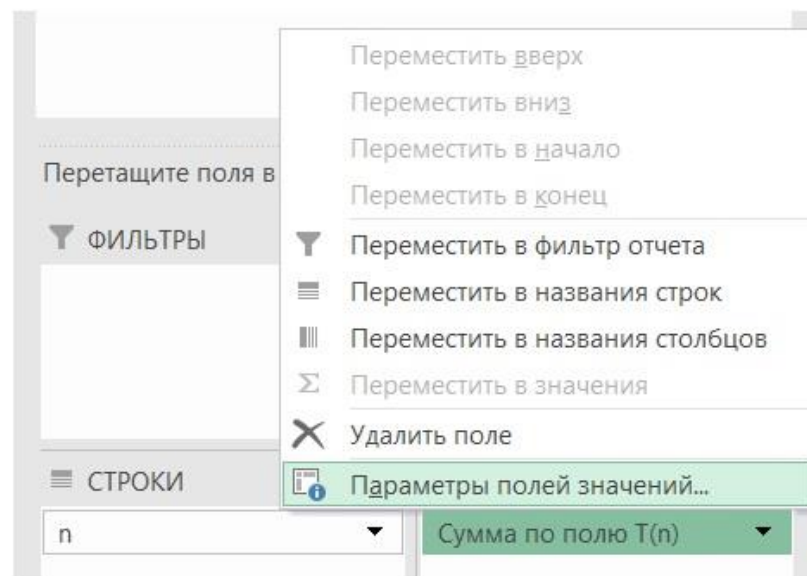
## Сводная таблица1

Чтобы построить отчет, выберите поля из списка полей сводной таблицы



# Агрегируем: Минимальное значение

Названия строк	Сумма по полю T(n)
8000000	1,1078154
16000000	2,2119225
24000000	3,2720949
32000000	4,3841593
40000000	5,4528033
<b>Общий итог</b>	<b>16,4287954</b>



Параметры поля значений

Имя источника: T(n)

Пользовательское имя: Минимум по полю T(n)

Операция **Дополнительные вычисления**

**Операция**

Выберите операцию, которую следует использовать для сведения данных в выбранном поле

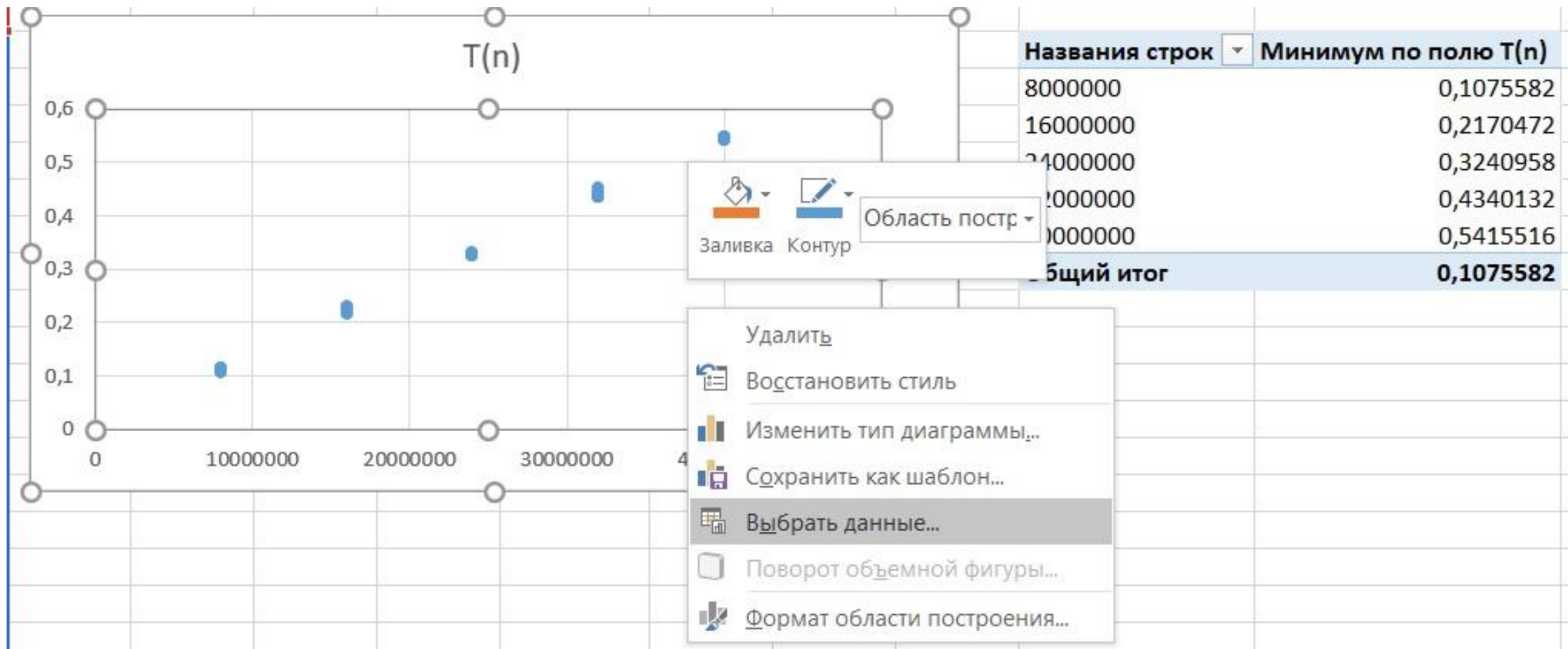
Сумма  
Количество  
Среднее  
Максимум  
**Минимум**  
Произведение

Числовой формат

ОК Отмена



# Данные для диаграммы



# Добавляем «ряд данных» на график

Выбор источника данных

Диапазон данных для диаграммы:

Диапазон данных слишком сложен для отображения. При выборе нового диапазона старый диапазон будет заменен.

Строка/столбец

Элементы легенды (ряды)

☒ T(n)  
☒ min T

Подписи горизонтальной оси (категории)

8000000  
16000000  
24000000  
32000000  
40000000

Скрытые и пустые ячейки

OK Отмена

Названия строк	Минимум по полю T(n)
8000000	0,1075582
16000000	0,2170472
24000000	0,3240958
32000000	0,4340132
40000000	0,5415516
Общий итог	0,1075582

Изменение ряда

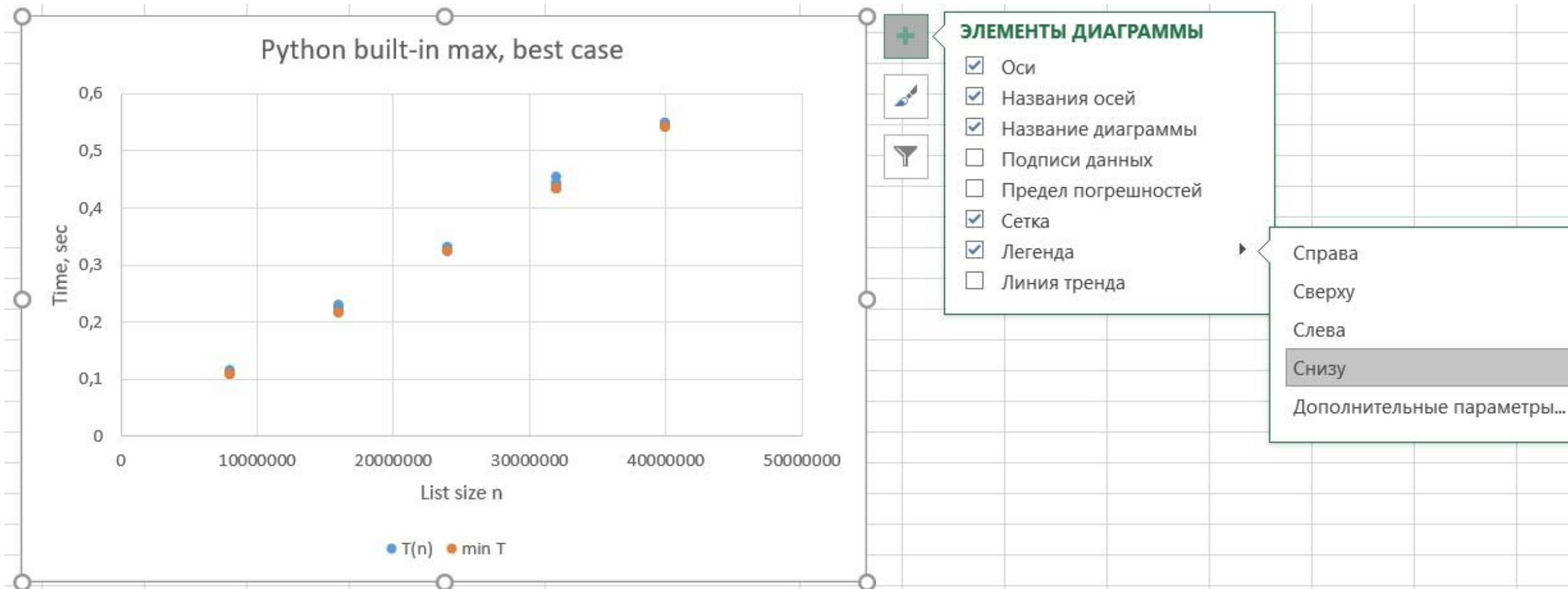
Имя ряда: min T = min T

Значения X: =maxx\_py\_best!\$K\$3:\$K\$7 = 8000000; 16000...

Значения Y: =maxx\_py\_best!\$L\$3:\$L\$7 = 0,1075582; 0,2...

OK Отмена

# Оформление диаграммы



# Линия «тренда»

## ЭЛЕМЕНТЫ ДИАГРАММЫ

- ☒ Оси
- ☒ Названия осей
- ☒ Название диаграммы
- ☐ Подписи данных
- ☐ Предел погрешностей
- ☒ Сетка
- ☒ Легенда
- ☐ Линия тренда

Линейный  
Экспоненциальный  
Линейный прогноз  
Скользящее среднее за два периода  
Дополнительные параметры...

## Добавление линии тр...

Добавление линии тренда для ряда:

T(n)

min T

OK

## Формат линии тренда

### Параметры линии тренда ▼



### ▲ Параметры линии тренда



☐ Экспоненциальная

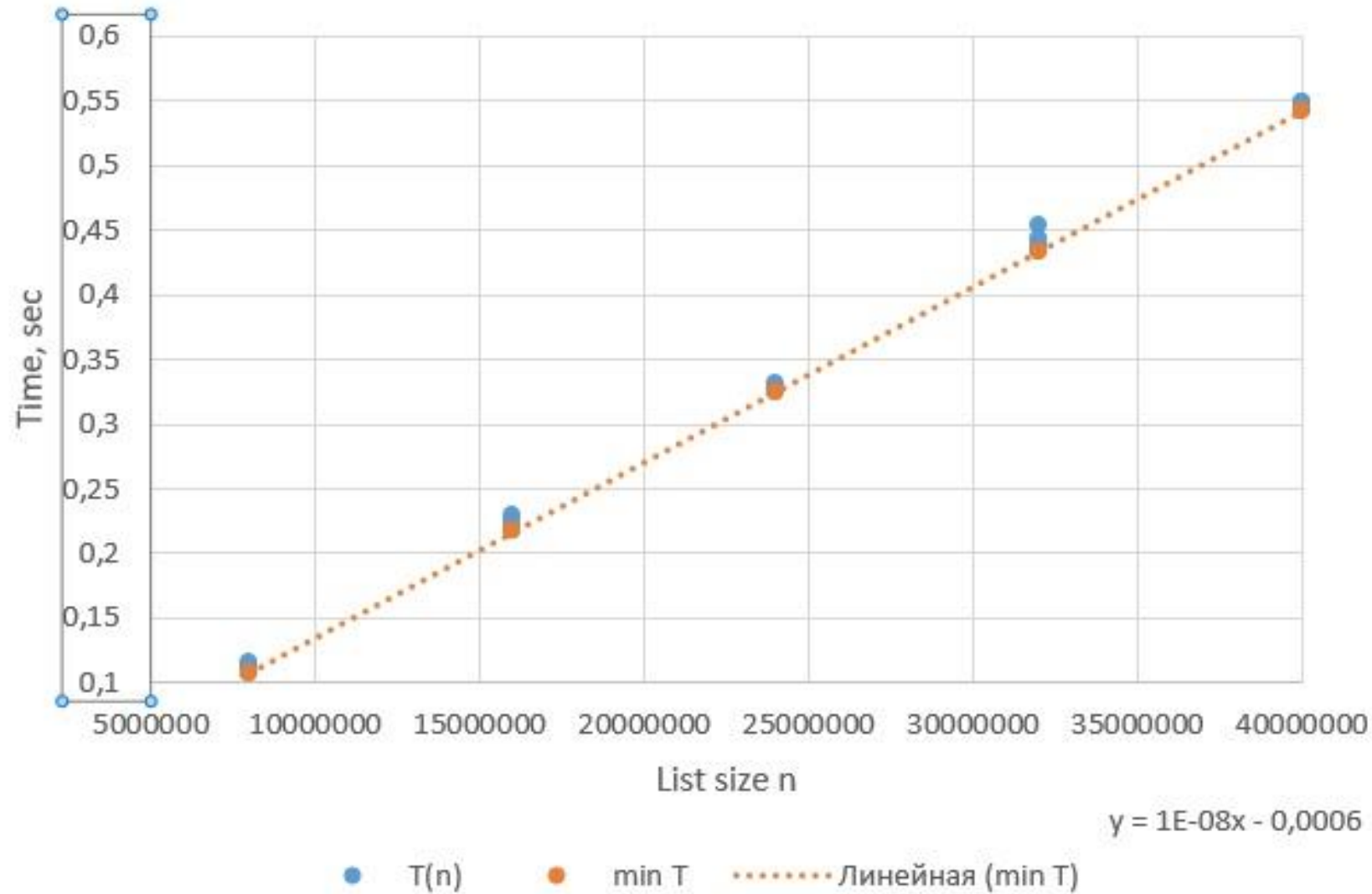


☒ Линейная

☐ Настроить пересечение

☒ показывать уравнение на диаграмме

## Python built-in max, best case



## Формат оси

### Параметры оси

Параметры текс



### Параметры оси

Границы

Минимум

Максимум

Единицы измерения

Главные

Дополнительные

Горизонтальная ось пересекает

☒ Автовыбор

☐ Значение оси

☐ Максимальное значение по о



## Python built-in max

