

Построение и анализ алгоритмов

Алгоритмические задачи

Построение алгоритмов

- Формулировка / постановка задачи
 - Точка А: Входные данные
 - Точка Б: Конечная цель
- Наивная реализация: «напрямую через поле»
 - Буквальная реализация формулировки задачи
 - Вычислительная сложность
- Улучшение / оптимизация алгоритма: «объезд по шоссе»
 - «Усложнение» алгоритма → упрощение работы
 - Количество операций $T(n)$
 - Размер памяти $M(n)$

Оценка вычислительной сложности

- Время вычислений T в зависимости от размера задачи n
 - $T(n) = O(?)$
- Теоретические оценки
 - Анализ псевдокода / исходного текста
 - Все операции выполняются за одинаковое время
 - Отбрасываем младшие степени и постоянные коэффициенты
- Хронометраж
 - Многократный прогон
 - Сводные таблицы

Точность таймера

```
1  using System;
2  using System.Diagnostics;
3  class Program
4  {
5      static void Main()
6      {
7          long frequency = Stopwatch.Frequency;
8          Console.WriteLine($"Частота таймера: {frequency} Гц");
9      }
10 }
```

```
C:\00\bin\Release\net9.0>
Частота таймера: 10000000 Гц
```

Простой таймер Python

```
n = 300_000
nums = [1] * n

import time
t = time.time()
m = max(nums)
dt = time.time() - t

print(f"{n} ; {dt:.10}")
```

```
C:\TMP\alg>python timer2.py
300000;0.01562452316
```

```
C:\TMP\alg>python timer2.py
300000;0.0
```

```
C:\TMP\alg>python timer2.py
300000;0.00651884079
```

```
C:\TMP\alg>python timer2.py
300000;0.01562333107
```

Точный таймер: High-resolution timer

```
n = 10
nums = [1] * n

import time
t = time.perf_counter()
m = max(nums)
dt = time.perf_counter() - t

print(f"{n} ; {dt:.10f}")
```

```
C:\TMP\alg>python timer3.py
10;0.0000022000
```

```
C:\TMP\alg>python timer3.py
10;0.0000027000
```

```
C:\TMP\alg>python timer3.py
10;0.0000023000
```

```
C:\TMP\alg>python timer3.py
10;0.0000022000
```

Формат вывода: cs / py

- Число знаков дробной части
- **.ToString("F10")** или **:F10** **:.10f**
- Версия Excel EN RU
 - Разделитель полей: запятая или точка с запятой
 - Десятичный разделитель: точка или запятая
- Региональные настройки EN RU
- Поиск и замена
- **.Replace('.', ',')** **.replace('.', ',')**

```
Console.WriteLine($"{n};{stopwatch.Elapsed.TotalSeconds.ToString("F10").Replace('.', ',')}");  
print(f"{n};{dt:.10f}".replace(".", ","))
```

Поиск максимума

Find max

Поиск максимального значения

- Однопроходный алгоритм
 - Начальное значение
 - Проверка условия
 - Обновление максимума
-
- Наилучший и наихудший случаи
 - Оценка сверху и оценка снизу

The Analysis of Algorithms

Find $\max(X_1, \dots, X_n)$

$m \leftarrow X_n$
 $k \leftarrow n-1$

$k=0?$

Yes

Output m

No

$X_k > m?$

Yes

No

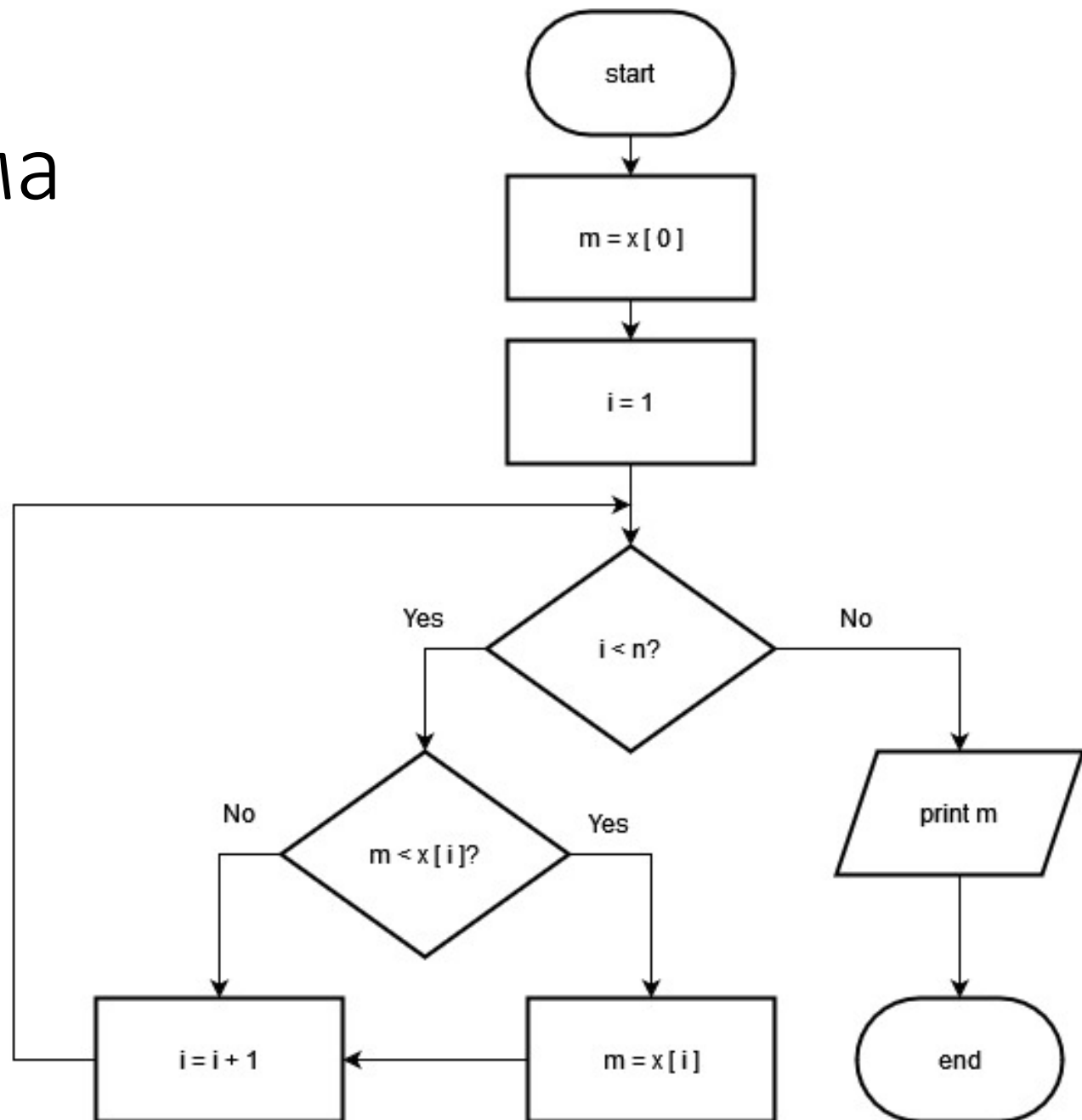
$m \leftarrow X_k$

$k \leftarrow k-1$

The Analysis of Algorithms

$n=4$ $X_1=2.3$ $X_2=7.1$ $X_3=4.0$ $X_4=5.9$

Схема алгоритма



```

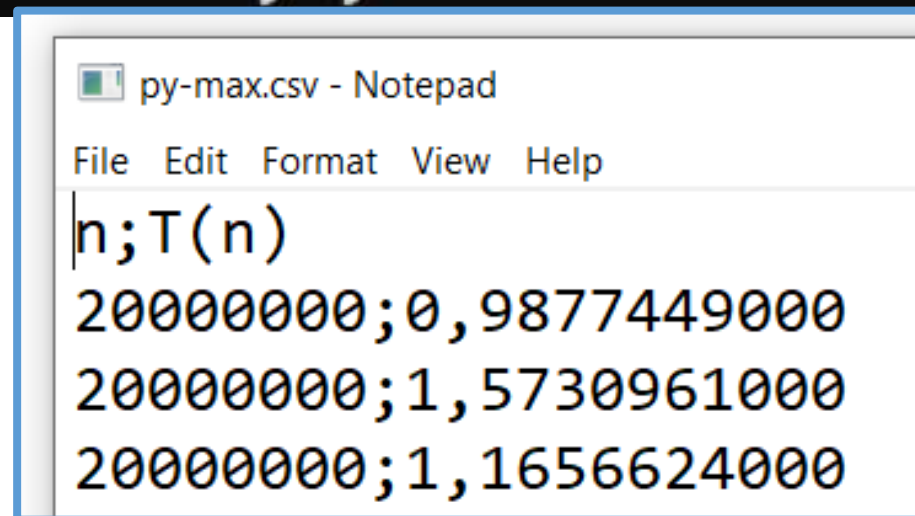
def find_max(nums):
    max_num = nums[0]
    for i in range(1, len(nums)):
        if max_num < nums[i]:
            max_num = nums[i]
    return max_num

import time
print("n;T(n)")
for n in range(20_000_000, 100_000_001, 20_000_000):
    nums = n * [1]
    # nums = [2] + (n - 1) * [1]
    for i in range(10):
        t = time.perf_counter()
        x_max = find_max(nums)
        dt = time.perf_counter() - t
        print(f"{n};{dt:.10f}".replace(".", ","), flush=True)

```

Python – Tee – CSV - Excel

```
C:\TMP\alg>python find_max_perf.py | tee py-max.csv  
n;T(n)  
20000000;0,9877449000  
20000000;1,5730961000  
20000000;1,1656624000
```

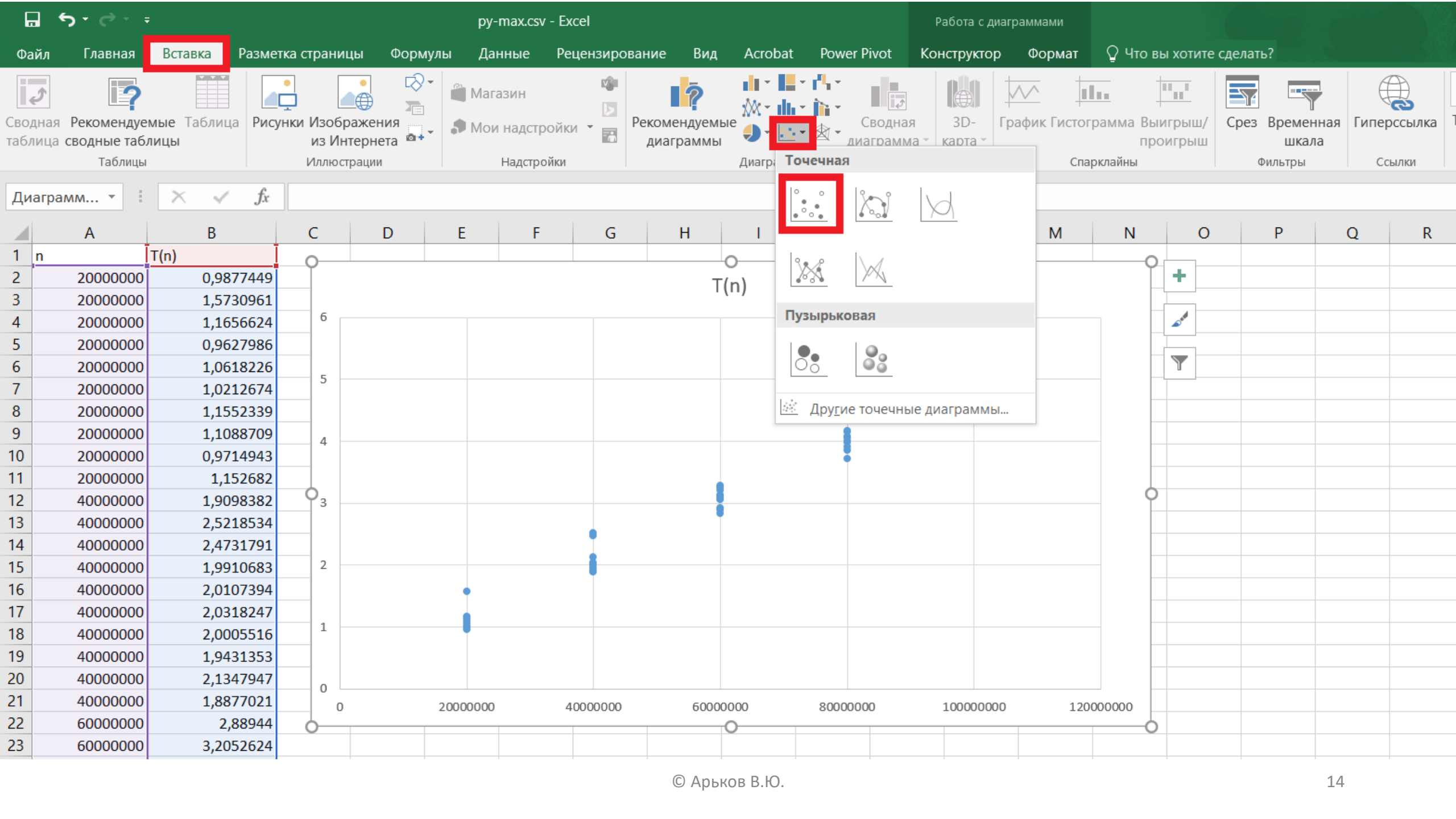


```
py-max.csv - Notepad  
File Edit Format View Help  
n;T(n)  
20000000;0,9877449000  
20000000;1,5730961000  
20000000;1,1656624000
```

Текст влево

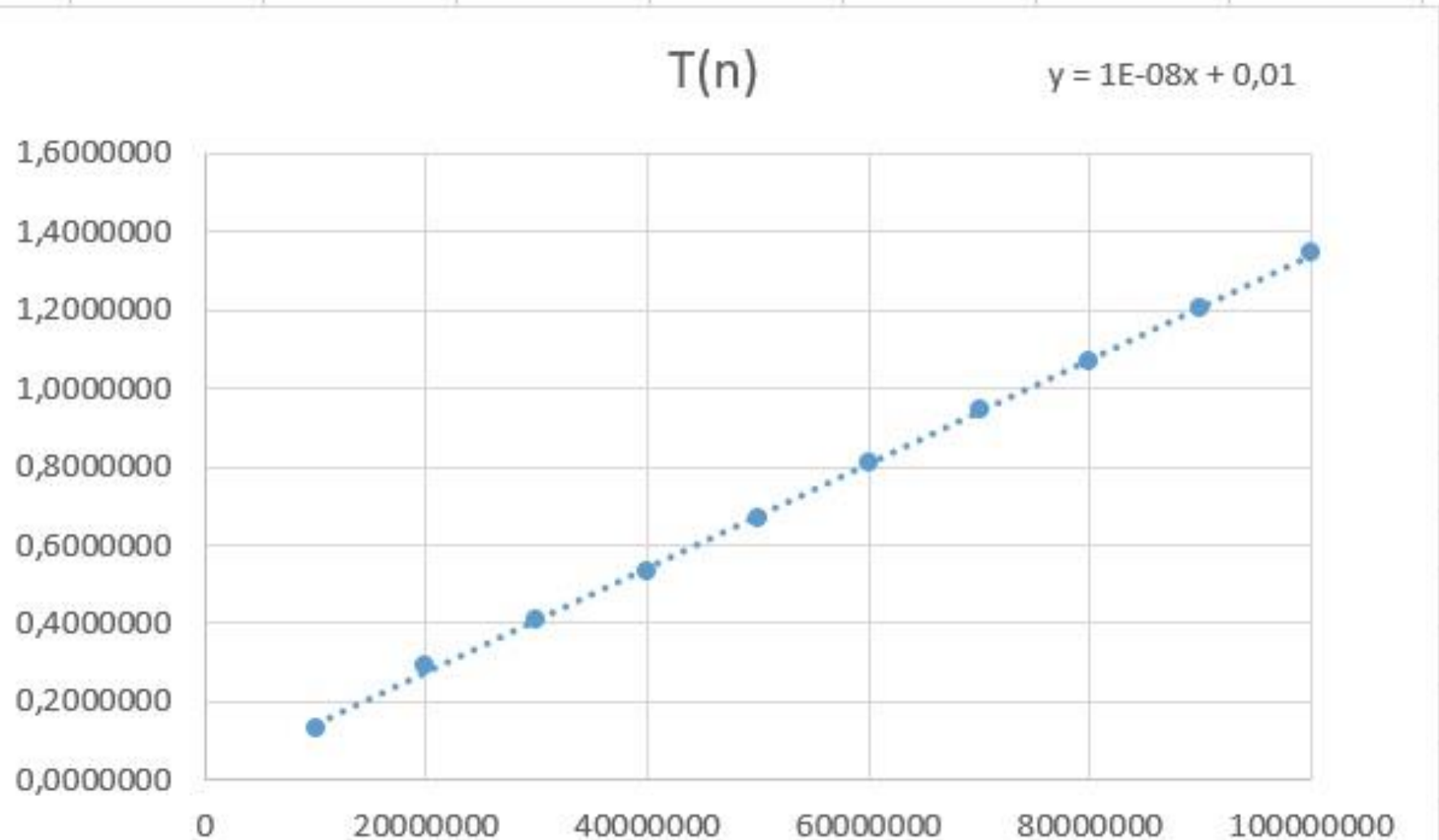
	A	B
1	n	T(n)
2	20000000	0,9877449
3	20000000	1,5730961
4	20000000	1,1656624

Числа вправо



Время выполнения и линия тренда

n	T(n)
10000000	0,1342037
20000000	0,2948463
30000000	0,4084883
40000000	0,5336411
50000000	0,6671250
60000000	0,8090007
70000000	0,9452651
80000000	1,0698245
90000000	1,2030134
100000000	1,3440893



6.csv - E

Файл Главная **Вставка** Разметка страницы Формулы Данные Рецензирование Вид Power Pivot Что вы

Сводная таблица Рекомендуемые таблицы Таблицы Рисунки Изображения из Интернета Иллюстрации Магазин Мои надстройки Надстройки Рекомендуемые диаграммы Диаграммы Сводные диаграммы

D1 X ✓ fx n

	A	B	C	D	E	F	G	H	I	J	K
1	n	T(n)									
2	10000000	0,520994186									
3	20000000	1,029001951									
4	30000000	1,594005346									
5	40000000	1,974999666									
6	50000000	2,409005642									
7	60000000	3,013005972									
8	70000000	3,976006508									
9	80000000	4,364999294									
10	90000000	5,029995441									
11	100000000	5,471997976									
12	10000000	0,481000423									
13	20000000	0,959000111									
14	30000000	1,436994076									
15	40000000	1,922006369									
16	50000000	2,40399313									
17	60000000	2,888005734									
18	70000000	3,921995163									
19	80000000	4,939997673									
20	90000000	4,652003527									
21	100000000	5,807002783									
22	10000000	0,64300108									
23	20000000	1,05700016									
24	30000000	1,617003679									
25	40000000	2,214999676									

Создание сводной таблицы

'6'!\$D\$1

Создание сводной таблицы

Выберите данные для анализа:

☒ Выбрать таблицу или диапазон

Таблица или диапазон: '6'!\$A\$1:\$B\$101

☐ 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: '6'!\$D\$1

Укажите, следует ли проанализировать несколько таблиц:

☐ Добавить эти данные в модель данных

OK Отмена

 Сводная таблица	Активное поле: <div> <div></div> <div>↓</div> <div>↑</div> <div>↔</div> </div> <div> <div>→</div> <div>Группировать</div> </div> <div> <div>Вставить срез</div> <div>Вставить временную шкалу</div> <div>Подключения к фильтрам</div> </div> <div> <div>Активное поле</div> <div>Фильтр</div> </div>	<div> <div>↺</div> <div>Обновить</div> </div> <div> <div>↻</div> <div>Источники данных</div> </div> <div> <div>⚙️</div> <div>Действия</div> </div> <div> <div>Данные</div> </div>	<div> <div>📄</div> <div>Поля, элементы и наборы</div> </div> <div> <div>🔗</div> <div>Средства OLAP</div> </div> <div> <div>📊</div> <div>Отношения</div> </div> <div> <div>Вычисления</div> </div>	<div> <div>📊</div> <div>Сводная диаграмма</div> </div> <div> <div>🔍</div> <div>Рекомендуемые сводные таблицы</div> </div> <div> <div>Сервис</div> </div>	<div> <div>📄</div> <div>Показать</div> </div>
---	--	---	---	--	---

Поля сводной таблицы

Выберите поля для добавления в отчет:

Поиск

☐ n

☐ T(n)

ДРУГИЕ ТАБЛИЦЫ

Перетащите поля в нужную область:

ФИЛЬТРЫ


СТОЛБЦЫ

СТРОКИ

Σ ЗНАЧЕНИЯ

Сводная таблица1

Чтобы построить отчет, выберите поля из списка полей сводной таблицы



n	T(n)
10000000	0,520994186
20000000	1,029001951
30000000	1,594005346
40000000	1,974999666
50000000	2,409005642
60000000	3,013005972
70000000	3,976006508
80000000	4,364999294
90000000	5,029995441
100000000	5,471997976
10000000	0,481000423
20000000	0,959000111
30000000	1,436994076
40000000	1,922006369
50000000	2,40399313
60000000	2,888005734
70000000	3,921995163
80000000	4,939997673
90000000	4,652003527
100000000	5,807002783
10000000	0,64300108
20000000	1,05700016
30000000	1,617003679
40000000	2,214999676
50000000	2,652000666

Поля сводной таблицы

Выберите поля для добавления в отчет:

Поиск

☐ n
☐ T(n)

ДРУГИЕ ТАБЛИЦЫ

Перетащите поля в нужную область:

ФИЛЬТРЫ

СТОЛБЦЫ

СТРОКИ

Σ ЗНАЧЕНИЯ

Поля сводной таблицы

Выберите поля для добавления в отчет:

Поиск

- ☒ n
- ☒ T(n)

ДРУГИЕ ТАБЛИЦЫ...

Перетащите

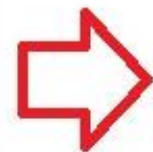
ФИЛЬТР

СТРОКИ

n

Сумма по полю T(n)

- Переместить вверх
- Переместить вниз
- Переместить в начало
- Переместить в конец
- Переместить в фильтр отчета
- Переместить в названия строк
- Переместить в названия столбцов
- Переместить в значения
- Удалить поле
- Параметры полей значений...



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

OK Отмена

```

1  using System;
2  using System.Diagnostics;
3  class Program
4  {
5      static int FindMax(int[] nums)
6      {
7          int max = nums[0];
8          foreach (int num in nums)
9              if (num > max)
10                 max = num;
11         return max;
12     }
13     static void Main()
14     {
15         int n = 100_000_000;
16         int[] nums = new int[n];
17         for (int i = 0; i < n; i++)
18             nums[i] = i;
19         Stopwatch stopwatch = Stopwatch.StartNew();
20         int x_max = FindMax(nums);
21         stopwatch.Stop();
22         Console.WriteLine($"{n} {stopwatch.Elapsed.TotalSeconds}");
23     }
24 }

```

```

C:\00\bin\Release\net9.0>00
100000000 0,0649237

```

```

C:\00\bin\Release\net9.0>00
100000000 0,0650805

```

```

C:\00\bin\Release\net9.0>00
300000000 0,1924222

```

```

C:\00\bin\Release\net9.0>00
300000000 0,1977138

```



```

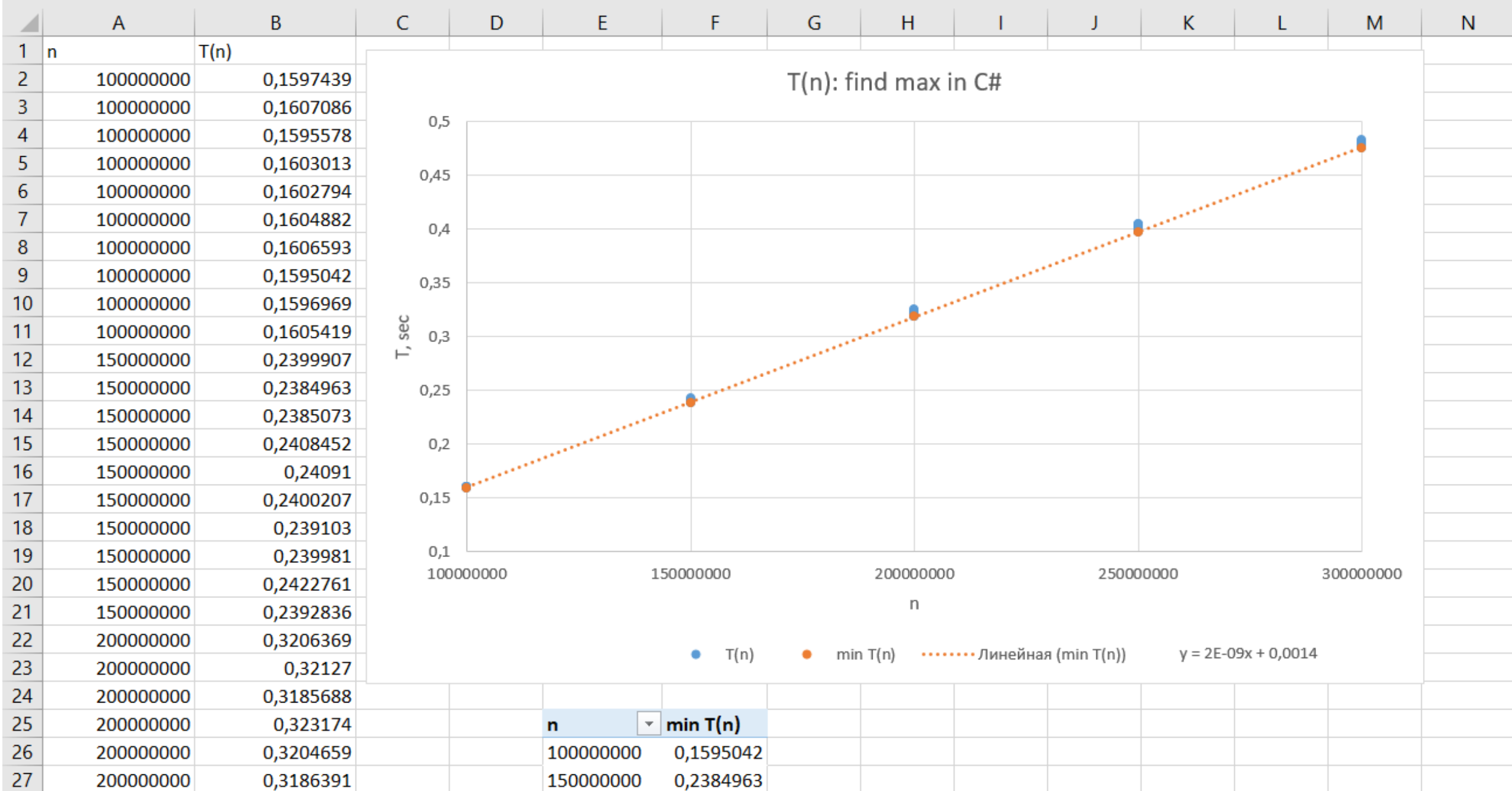
1  using System;
2  using System.Diagnostics;
3
4  class Program
5  {
6      static int FindMax(int[] nums)
7      {
8          int max = nums[0];
9          for (int i = 1; i < nums.Length; i++)
10             if (max < nums[i])
11                 max = nums[i];
12         return max;
13     }
14
15     static void Main()
16     {
17         Console.WriteLine("n;T(n)");
18         for (int n = 100_000_000; n <= 300_000_000; n += 50_000_000)
19         {
20             int[] nums = new int[n];
21             for (int i = 0; i < n; i++)
22                 nums[i] = 1;
23             for (int i = 0; i < 10; i++)
24             {
25                 Stopwatch stopwatch = Stopwatch.StartNew();
26                 int x_max = FindMax(nums);
27                 stopwatch.Stop();
28                 Console.WriteLine($"{n};{stopwatch.Elapsed.TotalSeconds:F10}");
29             }
30         }
31     }
32 }

```

```

C:\00\bin\x64\Release\net9.0> tee find_max_cs.csv
n;T(n)
100000000;0,1597439000
100000000;0,1607086000
100000000;0,1595578000

```



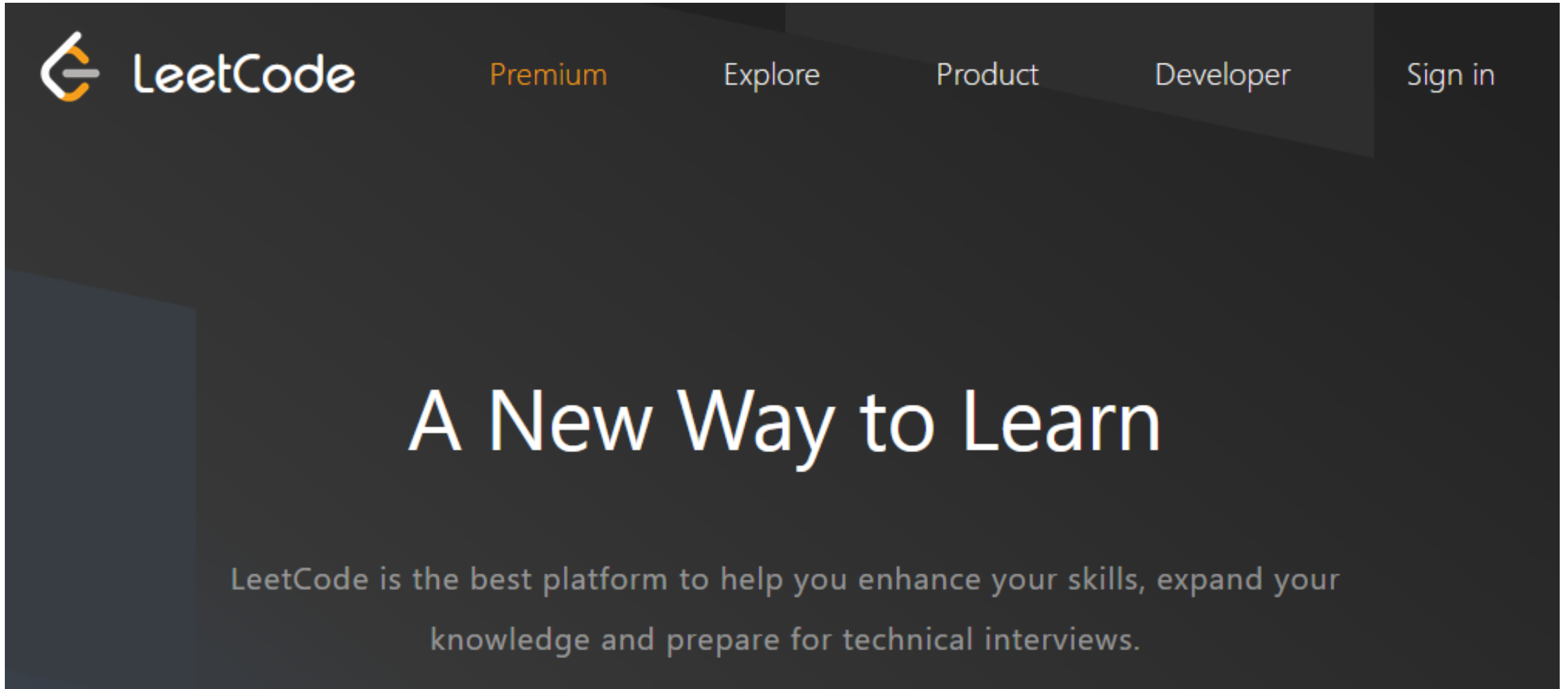
Алгоритмические задачи

Algorithmic problems

Алгоритмические задачи





- Изучение технологии программирования
- Алгоритмическая секция собеседования
- Олимпиады и соревнования
- Сборники заданий
- Онлайн тренажеры и платформы

LeetCode <https://leetcode.com/>



LeetCode

- online platform for coding interview preparation
- <https://leetcode.com/>
- <https://leetcode.com/studyplan/top-interview-150/>

Sliding Window		
<input type="radio"/> Minimum Size Subarray Sum	 Solution	Medium
<input type="radio"/> Longest Substring Without Repeating Characters	 Solution	Medium
<input type="radio"/> Substring with Concatenation of All Words	 Solution	Hard
<input type="radio"/> Minimum Window Substring	 Solution	Hard

Environments for the programming languages

- <https://support.leetcode.com/hc/en-us/articles/360011833974-What-are-the-environments-for-the-programming-languages->

Language	Version	Notes
Java	java 17	<p><code>OpenJDK 17</code> . Java 8 features such as lambda expressions and stream API can be used.</p> <p>Most standard library headers are already included automatically for your convenience.</p> <p>Includes <code>Pair</code> class from https://docs.oracle.com/javase/8/javafx/api/javafx/util/Pair.html.</p>
Python3	Python 3.10	<p>Most libraries are already imported automatically for your convenience, such as <code>array</code>, <code>bisect</code>, <code>collections</code>. If you need more libraries, you can import it yourself.</p> <p>For Map/TreeMap data structure, you may use <code>sortedcontainers</code> library.</p>
C#	C# 10 with .NET 6 runtime	

Скользящее окно

Sliding window

Метод скользящего окна

- 643. Maximum Average Subarray I
 - <https://leetcode.com/problems/maximum-average-subarray-i/>
 - <https://wcademy.ru/sliding-window-method/>
- Дан массив из n целых чисел
- Найдите непрерывный подмассив длины k , который имеет максимальное среднее значение.
- Вывести максимальное среднее значение
- Скользящее среднее

Простой / наивный алгоритм

- Высокие затраты ресурсов
- Низкая эффективность
- Понимание задачи и подходов к решению
- Основа для сравнения – *Benchmark*
- Единственное решение (в некоторых случаях)
- Брутфорс – *Brute Force*
 - Полный перебор всех возможных вариантов решений

Наивный алгоритм

i Python3  |  Auto

```
1 class Solution:
2     def findMaxAverage(self, nums: List[int], k: int) -> float:
3         n = len(nums)
4         if k > n:
5             return -1
6         subarray_sum = sum(nums[:k])
7         max_avg = subarray_sum / k
8         for i in range(1, n - k + 1):
9             subarray_sum = sum(nums[i:i + k])
10            subarray_avg = subarray_sum / k
11            max_avg = max(max_avg, subarray_avg)
12        return max_avg
```

Testcase Result

Time Limit Exceeded

Улучшенный алгоритм

Submissions

✓ Accepted

Runtime

1004 ms

Beats 69.26% of users with Python3

Memory

27.87 MB

Beats 88.62% of users with Python3

i Python3 ▾ | 🔒 Auto

```
1 class Solution:
2     def findMaxAverage(self, nums: List[int], k: int) -> float:
3         n = len(nums)
4         if k > n:
5             return -1
6         subarray_sum = sum(nums[:k])
7         max_avg = subarray_sum / k
8         for i in range(k, n):
9             subarray_sum += nums[i] - nums[i-k]
10            subarray_avg = subarray_sum / k
11            max_avg = max(max_avg, subarray_avg)
12        return max_avg
```

Два указателя

Two pointers

Метод двух указателей

- Дан массив целых чисел
- Вернуть индексы двух чисел, сумма которых равна заданному числу.
- Есть ровно одно решение.

- <https://wcademy.ru/two-pointers-method/>
- <https://leetcode.com/problems/two-sum/>

Метод двух указателей

- Two-pointer technique
 - <https://leetcode.com/articles/two-pointer-technique/>
- 167. Two Sum II - Input Array Is Sorted
 - <https://leetcode.com/problems/two-sum-ii-input-array-is-sorted/>
- 26. Remove Duplicates from Sorted Array
 - <https://leetcode.com/problems/remove-duplicates-from-sorted-array/>
- 11. Container With Most Water
 - <https://leetcode.com/problems/container-with-most-water/>
- 27. Remove Element
 - <https://leetcode.com/problems/remove-element/>
- 125. Valid Palindrome
 - <https://leetcode.com/problems/valid-palindrome/>

Какой метод используется для нахождения максимального значения в массиве?

- a) Попарное сравнение
- b) Метод двух указателей
- c) Скользящее окно
- d) Ничего из вышеперечисленного

Какова временная сложность алгоритма поиска максимального значения в массиве?

- a) $O(n)$
- b) $O(\log n)$
- c) $O(n \log n)$
- d) $O(n^2)$

Тест

Test

Что представляет собой метод скользящего окна?

- a) Нахождение максимального значения в массиве
- b) Нахождение подмассива по заданному критерию
- c) Поиск двух элементов, сумма которых равна заданному числу
- d) Ничего из вышеперечисленного

Какие особенности имеет метод двух указателей?

- a) Используется для поиска двух элементов с заданной суммой
- b) Находит максимальное значение в массиве за $O(n)$
- c) Находит левую и правую границы подмассива за один проход
- d) Работает только на упорядоченных списках и массивах

Каково основное преимущество метода двух указателей?

- a) Меньшая вычислительная сложность
- b) Возможность работать с отсортированными массивами
- c) Более простая реализация
- d) Лучшие результаты на малых наборах данных