

Министерство цифрового развития, связи и
массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Сибирский государственный университет
телекоммуникаций и информатики» (СибГУТИ)

ЛАБОРАТОРНАЯ РАБОТА №4
по дисциплине «**Моделирование**»

Выполнил:
студент гр. ИС-142
«__» мая 2025 г.

/Григорьев Ю.В./

Проверил:
преподаватель
«__» мая 2025 г.

/Уженцева А.В./

Оценка « _____ »

Новосибирск 2025

ВВЕДЕНИЕ

В данной работе реализована программа для генерации выборки из массива чисел и построения матриц включения. Основная цель исследования — проанализировать, как различные критерии выборки (например, числа, кратные 5, простые числа и случайный отбор) влияют на структуру выборки и ее представление в виде бинарной матрицы.

Кроме того, рассматриваются два вида выборки:

- **С возвратом** (один и тот же элемент может быть выбран несколько раз).
- **Без возврата** (каждый элемент выбирается не более одного раза).

Для визуализации результатов строятся графики матриц включения, которые сохраняются в файлы.

ВЫПОЛНЕНИЕ РАБОТЫ

Программа создает массив целых чисел от 1 до некоторого значения `size`. В рамках эксперимента были выбраны три размера массива:

- 10 элементов (малый размер, для наглядности),
- 100 элементов (средний размер),
- 1000 элементов (крупный размер).

Для генерации массива используется функция:

```
def generate_numbers(size, seed=42):  
    """Генерация массива чисел от 1 до size."""  
    np.random.seed(seed)  
    numbers = np.arange(1, size + 1)  
    return numbers
```

Выборка из массива производится по следующим критериям:

1. Числа, кратные 5 – в выборку попадают числа, делящиеся на 5 без остатка.
2. Простые числа – выбираются только числа, имеющие ровно два делителя: 1 и само число.

3. Случайная выборка – каждый элемент массива включается в выборку с вероятностью 50%.

Реализация критериев выглядит следующим образом:

```
sample_criteria = [ ("Кратные пяти", lambda x: x % 5 == 0),  
                    ("Простые числа", lambda x: all(x % i != 0 for i in range(2,  
int(np.sqrt(x)) + 1)) and x > 1),  
                    ("Случайная выборка", lambda x: random.random() > 0.5) ]
```

После применения критерия создаются два типа выборок:

```
def sample_with_replacement(numbers, sample_size):  
    """Выборка с возвращением"""  
    return [int(random.choice(numbers)) for _ in range(sample_size)]  
  
def sample_without_replacement(numbers, sample_size):  
    """Выборка без возвращения"""  
    return list(map(int, random.sample(list(numbers), sample_size)))
```

После формирования выборки строится бинарная матрица, которая показывает, какие элементы были выбраны.

Формат матрицы:

- Столбцы – индексы всех возможных элементов исходного массива.
- Строки – элементы из выборки.
- Значение 1 – если элемент был выбран.
- Значение 0 – если элемент не выбран.

```
def build_matrix(numbers, sample):  
    """Формирование матрицы включения"""  
    n = len(sample)  
    m = len(numbers)  
    matrix = np.zeros((n, m), dtype=int)  
    for i, value in enumerate(sample):
```

```

        idx = np.where(numbers == value)[0][0] # Индекс в исходном
массиве

        matrix[i, idx] = 1

    return matrix

```

Для наглядного представления матрицы строятся графики, где:

- Ось X – индексы элементов исходного массива.
- Ось Y – номера элементов выборки.
- Цвет – черный (0 – элемент не был взят) и белый (1 – элемент был взят).

```

def save_matrix_plot(matrix, title, filename):

    """Сохранение графика матрицы в файл"""

    plt.figure(figsize=(10, 6))

    plt.imshow(matrix, cmap="gray", aspect="auto")

    plt.xlabel("Индексы элементов")

    plt.ylabel("Выборки")

    plt.title(title)

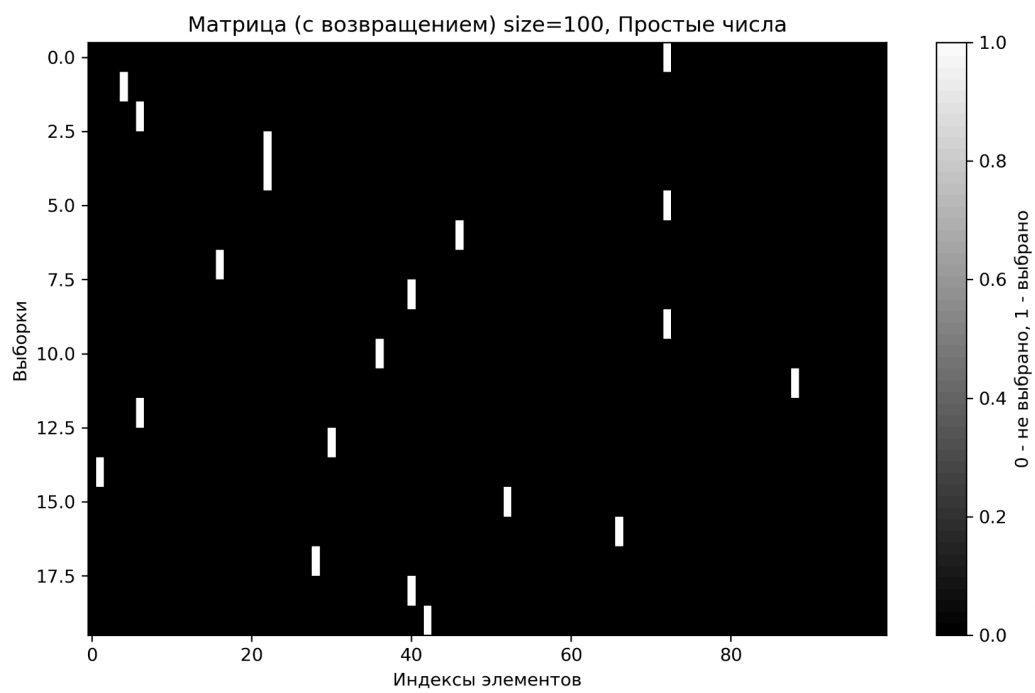
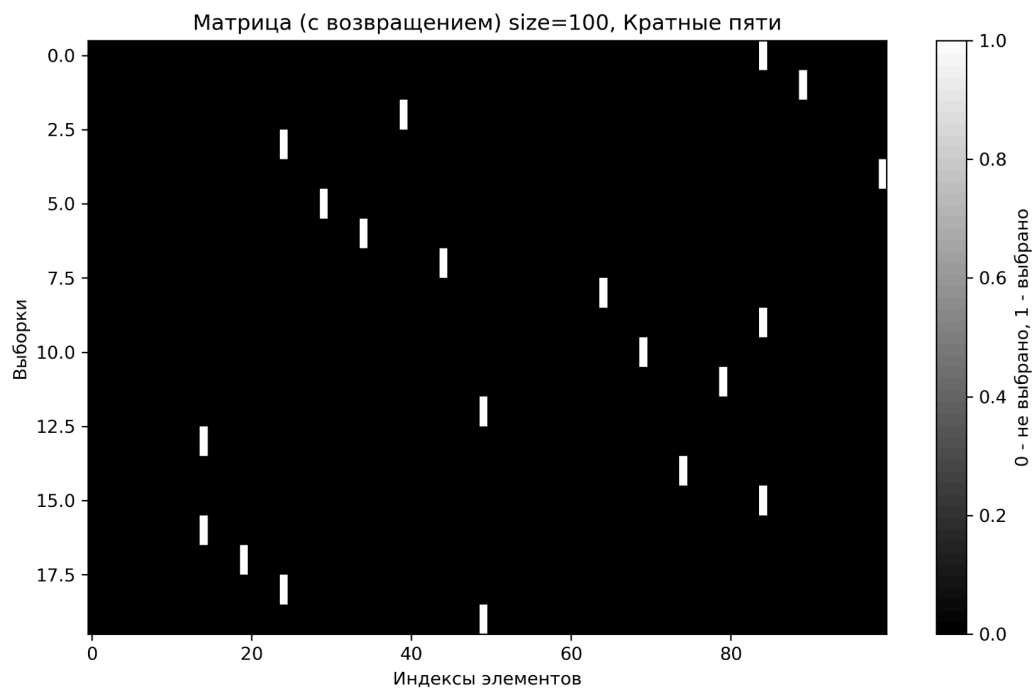
    plt.colorbar(label="0 - не выбрано, 1 - выбрано")

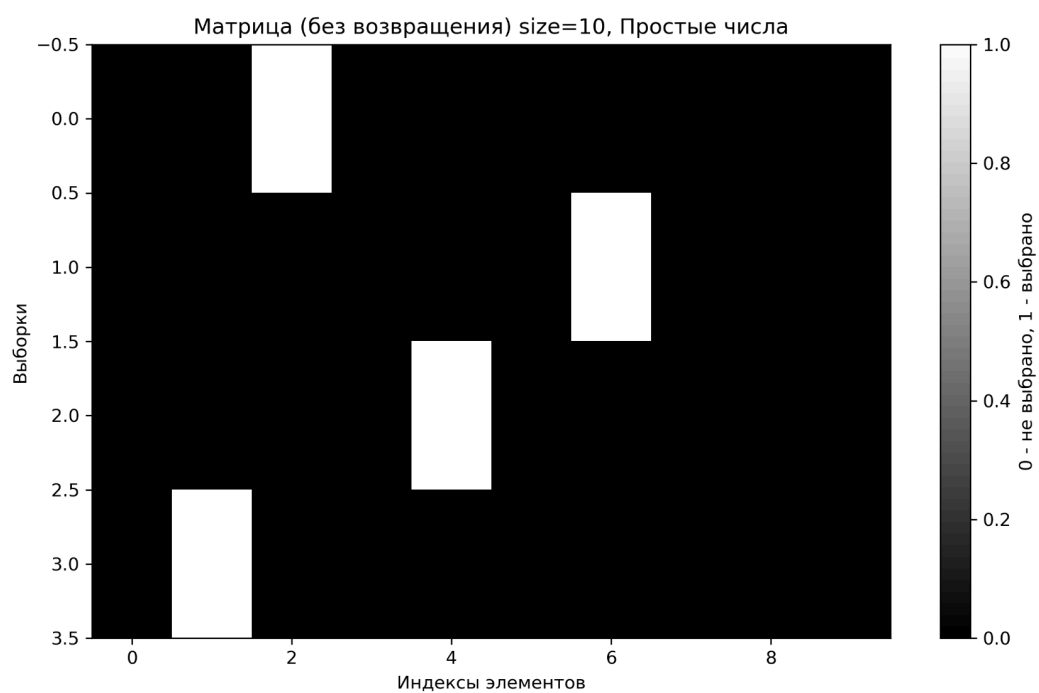
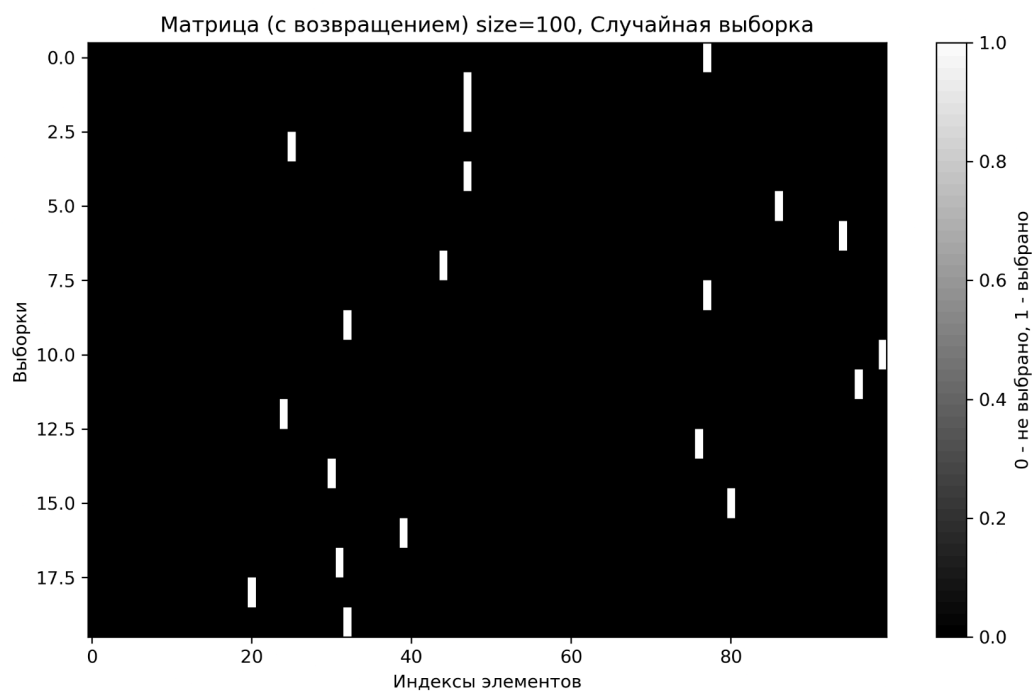
    plt.savefig(filename, dpi=300)

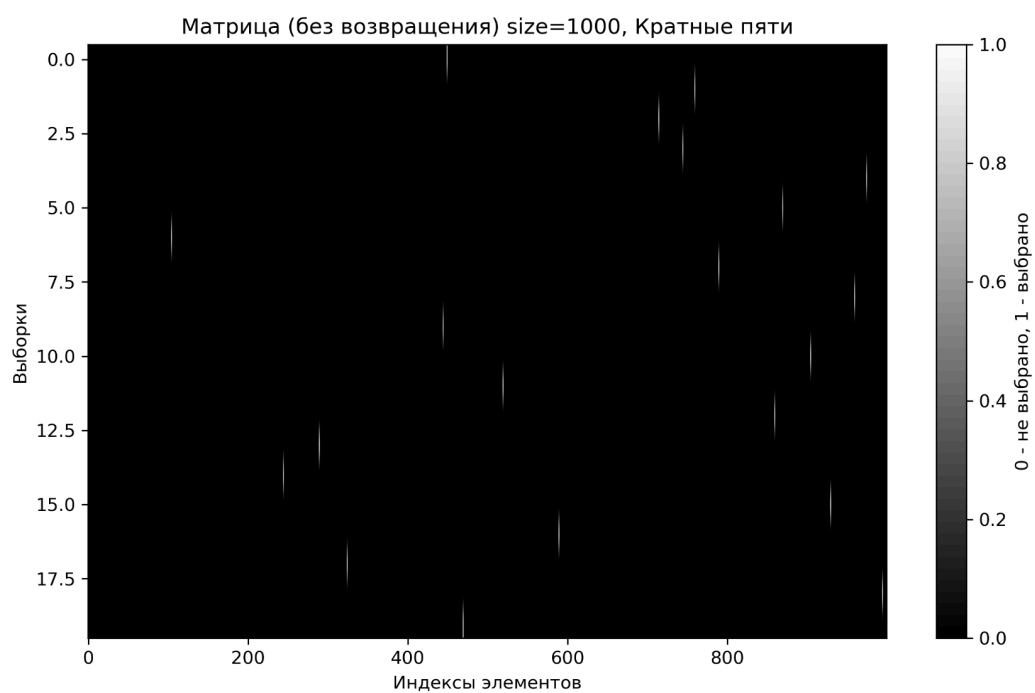
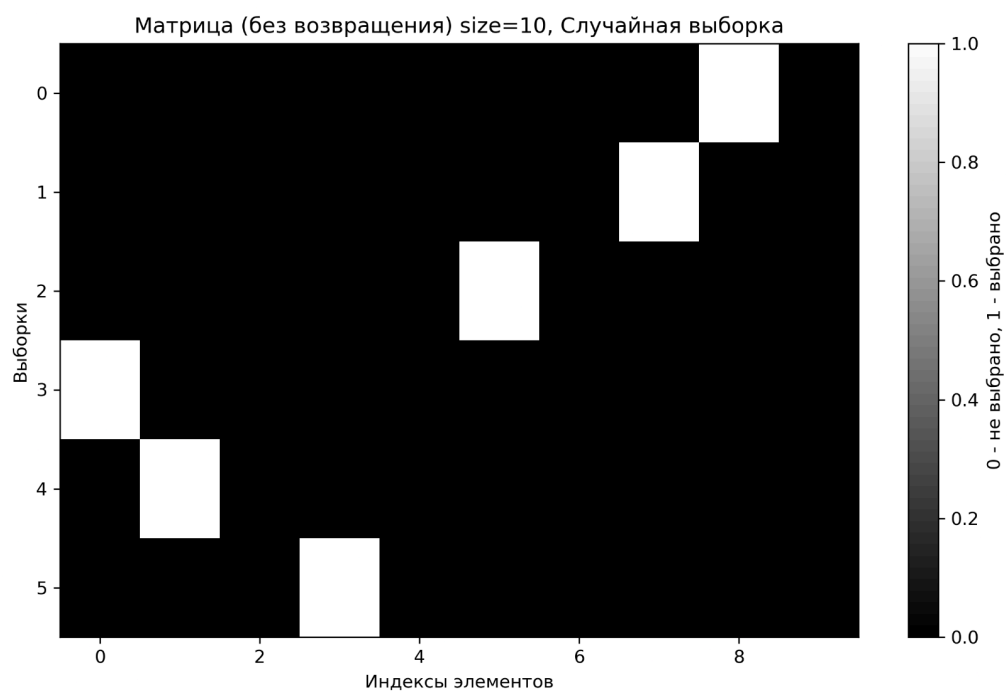
    plt.close()

```

Примеры построенных графиков (часть от всех сгенерированных):







Вывод в консоль:

Выборка (6 элементов) из 10 по критерию: Случайная выборка
[1, 2, 6, 7, 8, 10]

Пример размещений элементов выборки:

Размещение с возвратом: [2, 7, 7, 1, 6, 6]

Размещение без возвращения: [7, 2, 10, 1, 8, 6]

Матрица с возвратом:

```
0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 1 0 0 0
1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0 0 0
0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 1 0 0 0
1 0 0 0 0 0 0 0 0 0
```

Матрица без возвращения:

```
0 0 0 0 0 0 1 0 0 0
0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 1 0 0 0
0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0 0 0
```

Графики сохранены: plots\matrix_WR_10_Случайная_выборка.png, plots\matrix_WOR_10_Случайная_выборка.png

Выборка (4 элементов) из 10 по критерию: Простые числа
[2, 3, 5, 7]

Пример размещений элементов выборки:

Размещение с возвратом: [7, 5, 7, 3]

Размещение без возвращения: [3, 2, 7, 5]

Матрица с возвратом:

```
0 0 0 0 0 0 1 0 0 0
0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0
0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0
0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0
0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 1 0 0 0
0 0 0 0 1 0 0 0 0 0
```

Матрица без возвращения:

```
0 0 1 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0
0 0 0 0 1 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0
0 0 0 0 1 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0
```

Графики сохранены: plots\matrix_WR_10_Простые_числа.png, plots\matrix_WOR_10_Простые_числа.png

Исходный массив (size=100): [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100]

Выборка (20 элементов) из 100 по критерию: Кратные пяти

[5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100]

[illegible]

- Анализ распределения частот включений элементов.
- Применение других статистических критериев выборки.
- Генерация более сложных зависимостей между элементами массива.

Приложение.

Листинг программы

```
import numpy as np
import random
import matplotlib.pyplot as plt
import os

def generate_numbers(size, seed=42):
    np.random.seed(seed)
    numbers = np.arange(1, size + 1)
    return numbers

def sample_with_replacement(numbers, sample_size):
    return [int(random.choice(numbers)) for _ in range(sample_size)]

def sample_without_replacement(numbers, sample_size):
    return list(map(int, random.sample(list(numbers), sample_size)))

def build_matrix(numbers, sample, passes=3):
    n = len(sample) * passes # Увеличиваем количество строк
    m = len(numbers)
    matrix = np.zeros((n, m), dtype=int)

    # Заполняем матрицу за несколько проходов
    for pass_idx in range(passes):
        for i, value in enumerate(sample):
            idx = np.where(numbers == value)[0][0]
            matrix[pass_idx * len(sample) + i, idx] = 1

    return matrix

def save_matrix_plot(matrix, title, filename):
    plt.figure(figsize=(10, 6))
    plt.imshow(matrix, cmap="gray", aspect="auto")
    plt.xlabel("Индексы элементов")
    plt.ylabel("Выборки (несколько проходов)")
    plt.title(title)
    plt.colorbar(label="0 - не выбрано, 1 - выбрано")
    plt.savefig(filename, dpi=300)
    plt.close()

def print_matrix(matrix):
    for row in matrix:
        print(" ".join(str(x) for x in row))

output_dir = "plots"
os.makedirs(output_dir, exist_ok=True)

# Основные параметры
sizes = [10, 100] # Размеры исходных массивов
sample_criteria = [
    ("Кратные пяти", lambda x: x % 5 == 0),
    ("Простые числа", lambda x: all(x % i != 0 for i in range(2, int(np.sqrt(x)) + 1))
    and x > 1),
    ("Случайная выборка", lambda x: random.random() > 0.5)
]

for size in sizes:
    numbers = generate_numbers(size)
    print(f"\nИсходный массив (size={size}): {numbers.tolist()}")

    for criterion_name, criterion in sample_criteria:
        sample = [int(x) for x in numbers if criterion(x)]
        print(f"\nВыборка ({len(sample)} элементов) из {size} по критерию: {criterion_name}")
        print(sample)
```

```

if not sample:
    print(f"Для критерия {criterion_name} выборка пуста")
    continue

# Выборки с возвращением и без
sample_wr = sample_with_replacement(sample, min(len(sample), 20))
sample_wor = sample_without_replacement(sample, min(len(sample), 20))

print("\nПример размещений элементов выборки:")
print(f"    Размещение с возвращением: {sample_wr}")
print(f"    Размещение без возвращения: {sample_wor}")

# Строим матрицы (3 прохода)
matrix_wr = build_matrix(numbers, sample_wr, passes=3)
matrix_wor = build_matrix(numbers, sample_wor, passes=3)

# Выводим часть матрицы в консоль
print("\nМатрица с возвращением:")
print_matrix(matrix_wr[:min(10, len(matrix_wr))]) # Выводим первые 10 строк
print("\nМатрица без возвращения:")
print_matrix(matrix_wor[:min(10, len(matrix_wor))])

# Сохраняем графики
filename_wr = os.path.join(output_dir,
f"matrix_WR_{size}_{criterion_name.replace(' ', '_')}.png")
filename_wor = os.path.join(output_dir,
f"matrix_WOR_{size}_{criterion_name.replace(' ', '_')}.png")

save_matrix_plot(matrix_wr, f"Матрица (с возвращением) size={size},
{criterion_name}", filename_wr)
save_matrix_plot(matrix_wor, f"Матрица (без возвращения) size={size},
{criterion_name}", filename_wor)

print(f"Графики сохранены: {filename_wr}, {filename_wor}")

```