

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»
ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ
Кафедра интеллектуальных информационных технологий

Отчет
по дисциплине
«Современные методы защиты информации»
по лабораторной работе № 5
«Алгоритм Гровера»

Выполнила:
студентка 4
курса группы
ИИ-22
Сокол С.М.
Проверила:
Хацкевич А.С.

Цель: ознакомление с алгоритмом Гровера и его реализацией в квантовых системах.

Постановка задачи: Найдите количество итераций с наибольшей амплитудой при запуске алгоритма Гровера с одним решением в базе данных с $N=2^7$.

Ход работы:

Код:

```
from qiskit import QuantumCircuit, transpile
from qiskit_aer import AerSimulator # Используем AerSimulator для симуляции измерений

import matplotlib
matplotlib.use('QtAgg')
import matplotlib.pyplot as plt

# Определим целевое состояние и количество кубитов
state = '1111' # Целевое состояние должно совпадать с количеством кубитов
n = 4

def oracle(circuit):
    for i, bit in enumerate(state):
        if bit == '0':
            circuit.x(i)

    circuit.h(n - 1)
    circuit.mcx(list(range(n - 1)), n - 1)
    circuit.h(n - 1)

    for i, bit in enumerate(state):
        if bit == '0':
            circuit.x(i)

    circuit.barrier()

def diffuser(circuit):
    circuit.h(range(n))
    circuit.x(range(n))
    circuit.h(n - 1)
    circuit.mcx(list(range(n - 1)), n - 1)
    circuit.h(n - 1)
    circuit.x(range(n))
    circuit.h(range(n))
    circuit.barrier()

def main(debug=False):
    # Используем AerSimulator для запуска квантовой схемы с измерениями
    simulator = AerSimulator()
```

```

correct = []
iterations = range(15)

for iteration in iterations:
    qc = QuantumCircuit(n)

    # Начальная суперпозиция
    qc.h(range(n))
    qc.barrier()

    # Применяем oracle и diffuser заданное количество раз
    for _ in range(iteration):
        oracle(qc)
        diffuser(qc)

    # Добавляем измерения
    qc.measure_all()

    # Транспилируем схему для симулятора и запускаем симуляцию
    transpiled_qc = transpile(qc, simulator)
    result = simulator.run(transpiled_qc, shots=1024).result()

    # Получаем количество успешных измерений
    counts = result.get_counts()
    correct_value = counts.get(state[::-1], 0) # Переворачиваем строку состояния для
соответствия результату
    correct.append(correct_value)

    # Отладочный вывод
    print(f"Итерация {iteration}: Состояние после измерения - {counts}")
    print(f"Количество успешных измерений для состояния '{state}': {correct_value}")

# Построение графика
plt.figure(figsize=(12, 6))
plt.bar(iterations, correct, color='skyblue', edgecolor='navy')
plt.xlabel('# of iterations')
plt.ylabel('# of times the solution was obtained')
plt.title("Grover's Algorithm Performance (Database size: 2^4)")
plt.xticks(iterations)
plt.grid(axis='y', linestyle='--', alpha=0.7)

for i, v in enumerate(correct):
    plt.text(i, v, str(v), ha='center', va='bottom')

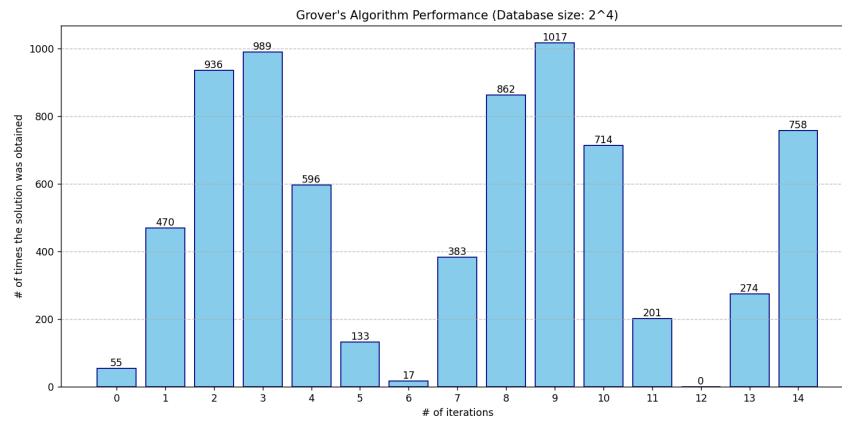
plt.tight_layout()
plt.show()

if __name__ == "__main__":

```

```
main(True)
```

Вывод программы:



Вывод: ознакомилась с алгоритмом Гровера и его реализацией в квантовых системах