

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

**Кафедра ИИТ**

**ОТЧЁТ**  
**По лабораторной работе №5**  
**«Алгоритм Гровера»**

Выполнил:  
Студент группы ИИ-22  
Борейша О.С.  
Проверил:  
Хацкевич А.С.

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

### Задачи:

Найдите количество итераций с наибольшей амплитудой при запуске алгоритма Гровера с одним решением в базе данных с  $N=2^7$ . Как показано выше, измените количество итераций и проверьте усиление. Ответ должен быть целым числом.

### Ход работы

```
from qiskit import QuantumCircuit, QuantumRegister, ClassicalRegister
from qiskit import BasicAer, execute
import numpy as np
import matplotlib.pyplot as plt

backend = BasicAer.get_backend('qasm_simulator')
prob_of_ans = []

for x in range(15):
    database = QuantumRegister(7)
    oracle = QuantumRegister(1)
    auxiliary = QuantumRegister(6)
    cr = ClassicalRegister(7)
    qc = QuantumCircuit(database, oracle, auxiliary, cr)

    qc.h(database[:])
    qc.x(oracle[0])
    qc.h(oracle[0])

    for j in range(x):
        qc.mct(database[:], oracle[0], auxiliary[:], mode='basic')

        qc.h(database[:])
        qc.x(database[:])
        qc.h(database[6])
        qc.mct(database[0:6], database[6], auxiliary[:], mode='basic')
        qc.h(database[6])
        qc.x(database[:])
        qc.h(database[:])

    qc.h(oracle[0])
    qc.x(oracle[0])
    qc.measure(database, cr)

    qc = qc.reverse_bits()

    job = execute(qc, backend=backend, shots=1000, seed_simulator=1234,
backend_options={"fusion_enable":True})
    result = job.result()
    count = result.get_counts()
    answer = count.get('1111111', 0)
    prob_of_ans.append(answer)
```

```

iteration = list(range(15))
correct = prob_of_ans

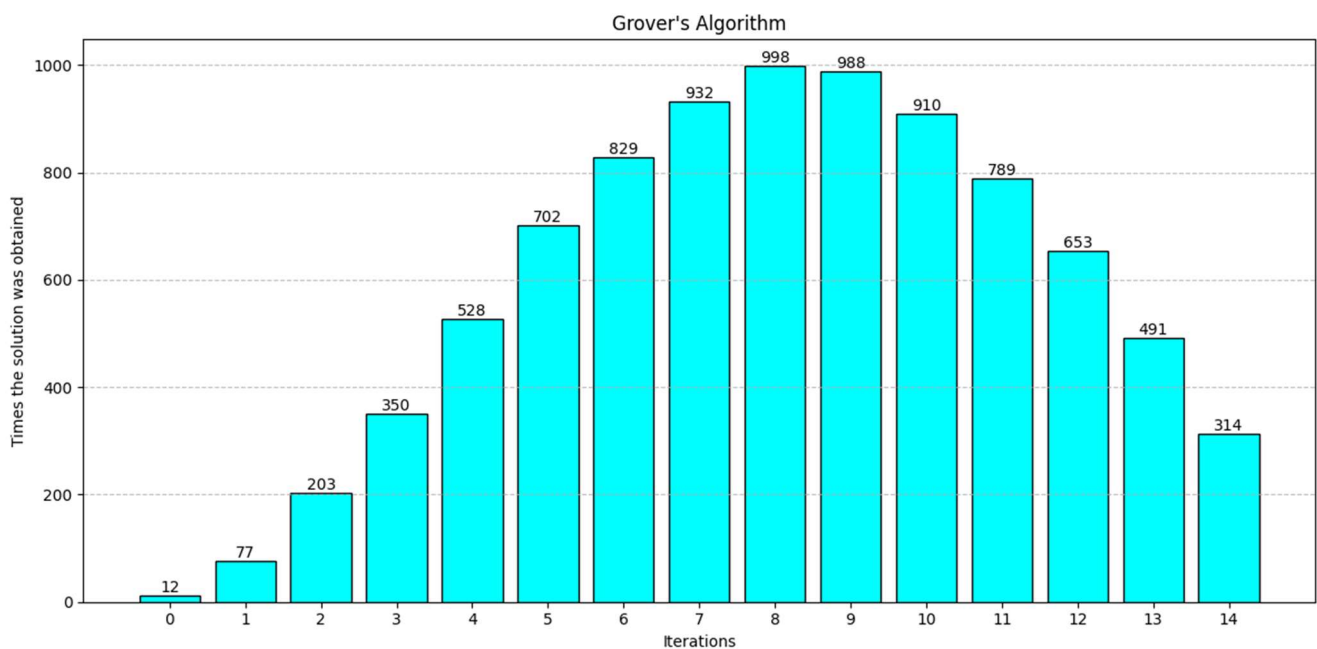
plt.figure(figsize=(12, 6))
plt.bar(iteration, correct, color='aqua', edgecolor='black')
plt.xlabel('Iterations')
plt.ylabel('Times the solution was obtained')
plt.title("Grover's Algorithm")
plt.xticks(iteration)
plt.grid(axis='y', linestyle='--', alpha=0.8)

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

plt.tight_layout()
plt.show()

```

## Результат работы:



**Вывод:** ознакомился с алгоритмом Гровера и его реализацией в квантовых системах.