Journal of Theory, Mathematics and Physics

Vol. 2, No. 11, 2023 ISSN: 2181-4376

Innova Science

ALGORITHM FOR SIMULATING QUANTUM GROVER'S ALGORITHM

Annotation: *The article is devoted to the study of Grover's quantum algorithm. This algorithm is designed to search for the value of a certain parameter in a given unordered space. As the study showed, the use of Grover's algorithm allows one to obtain a quadratic speedup compared to classical search algorithms. An analysis of the fundamental principles of quantum computing is also carried out: quantum bit, superposition, basic quantum elements.*

Keywords: *quantum bit, qubit, quantum computer, Grover's quantum algorithm.*

Information about the authors

Rasulov Xaydar Raupovich
Associate Professor of the Department of Mathematical Analysis of Bukhara State University, Uzbekistan

Raupova Mokhinur Haydar kizi
Teacher of the Department of Algebra and Mathematical Analysis of Chirchik State Pedagogical University, Uzbekistan

The quantum computing model has attracted close attention from scientists in recent decades, namely its implementation as a quantum computer. At the moment, it is known that a quantum computer is capable of computing complex problems for which there are no effective solution algorithms on a classical computer. For example, using classical calculations it is impossible to effectively solve the factorization problem. In quantum computing, Shor's algorithm is used, which allows solving the factorization problem in polynomial time. At the same time, other problems were discovered that could be solved more efficiently on quantum computers.

The basis of the quantum computing model is a quantum bit (qubit), which is an analogue of a classical bit. A qubit, like a bit, can be in the state $|0\rangle$, $|1\rangle$. However, unlike a bit, it can also be in a superposition of states, which is a linear combination of states of a quantum bit. For a quantum system consisting of one qubit, the superposition can be represented as follows:

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, (1)

where numbers α and β are complex coefficients satisfying the condition:

$|\alpha|^2+|\beta|^2=1.$

Another difference between a qubit and a bit is that it cannot be measured to determine its state. Only more limited information about its quantum state can be obtained. When measuring a qubit, you will get 0 with probability $|\alpha|^2$ or 1 with probability $|\beta|^2$. After the measurement, the qubit goes into a state that corresponds to the measurement, i.e. it collapses from a superposition into a certain state

In a quantum computer, the state of a qubit changes under the influence of various quantum elements. The following can be distinguished as standard quantum elements: the NOT quantum element and the Hadamard transform.

The quantum NOT gate is an analogue of the classical NOT gate. Its action is to transfer the state $\alpha|0\rangle + \beta|1\rangle$ to a state in which $|0\rangle$ and $|1\rangle$ are swapped.

The Hadamard transformation allows you to transform the basic state into an equally probable one, i.e. when measuring; any result can be obtained with equal probability. Many quantum algorithms use the Hadamard transform as the initial and final step.

To solve a problem, the computer needs to perform a certain sequence of operations. The description of this sequence is called an algorithm for solving the problem. To solve a problem, quantum algorithms are created on a quantum computer, which, unlike classical ones, take into account the laws of quantum physics. At the moment, about sixty quantum algorithms have been developed.

Let's consider Grover's quantum algorithm designed to search for the value of a certain parameter in a given unordered space. Let a Boolean function $f(x)$, $x\in\{0,1\}^n$, be given, which is represented as a black box. The goal of Grover's algorithm is to find x such that $f(x)=1$ (the function is given in the form of an oracle).

Grover's algorithm can be represented as follows:

- Initialization of the initial state. At this stage, it is necessary to prepare an equally probable superposition of the states of all input qubits.
- Applying Grover iteration. The Grover iteration consists of an oracle and a Grover diffusion operator (conditional phase shift). This stage is repeated $\sqrt{N}$ times.
- Measurement. At this stage, the output qubit register is measured.

The main part of the algorithm under consideration is the Grover iteration, which is divided into four steps: applying an oracle (a gate that calculates a given function.); application of the Hadamard transform; applying a conditional phase shift to the register; application of the Hadamard transform. The last three steps are combined into the Grover diffusion operator.

In the analyzed algorithm, the oracle is designed to recognize the solution to the search problem. When the input of the function f is given a value x at which $f(x) = 1$, the oracle marks this solution by shifting the phase of the quantum state that corresponds to the value x.

As you can see, the classical algorithm solves the search problem using the brute force method, i.e. in the best case, x will be found on the first try, and in the worst case, you will have to go through 2n options. Consequently, x can be found using this method in O(N) operations, where $N=2^n$. Grover's algorithm allows you to speed up the search method - up to $O(\sqrt{N})$ operations.

Thus, based on the above, we can conclude that Grover’s quantum algorithm makes it possible to find the value of a certain parameter in a given unordered space in $O\sqrt{N})$ calls to the oracle, i.e. gives a quadratic speedup compared to the classical algorithm.

To implement Grover's quantum algorithm, a service-oriented architecture was chosen. All business logic, which represents quantum computing, is implemented by a set of services, and verification of the entered data, their interpretation and output are carried out on the client side. The data entered by the user is transmitted via the protocol to the service, which performs quantum calculations. The result obtained is transmitted to the client.

Figure 1 shows the architecture of the designed system.

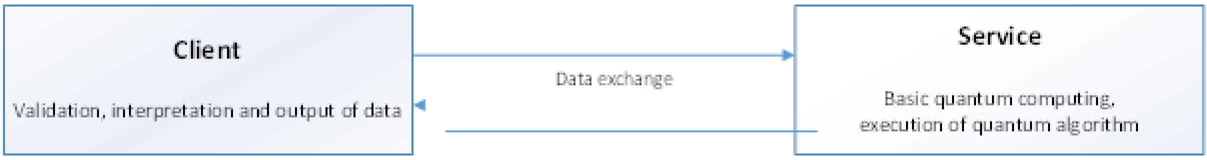

Client
Validation, interpretation and output of data
Data exchange
Service
Basic quantum computing,
execution of quantum algorithm

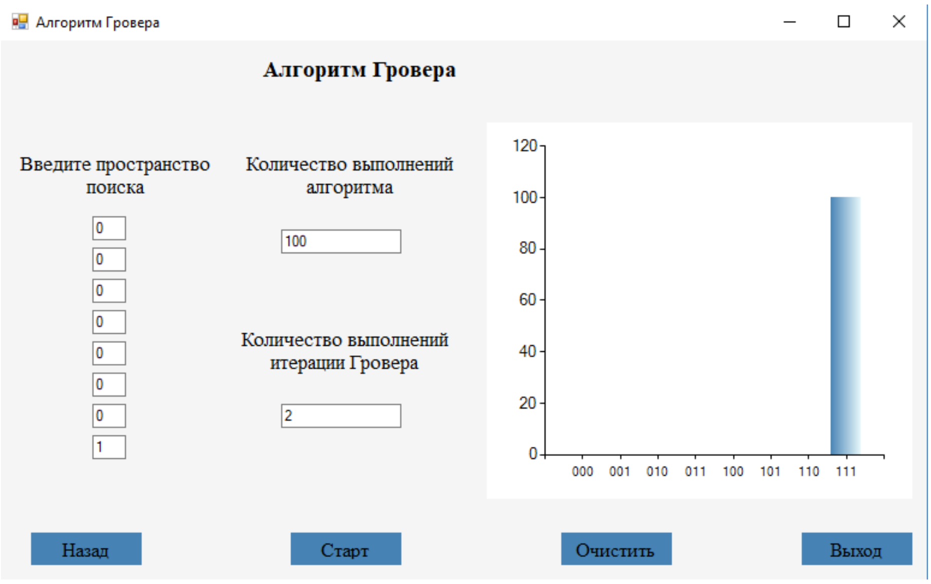
Figure 1 – Service-oriented architecture

When implementing Grover's algorithm, two classes were used: *StartGrover* and *Grover*. *StartGrover* checks and prepares input data, as well as displays the results in the form of a graph. Grover executes the quantum portion of Grover's algorithm.

The execution of Grover's algorithm begins with data input. The system then prepares a quantum register, which should consist of three qubits on which the Hadamard transform is performed.

The next stage begins with the design of the oracle and the Grover diffusion operator, which are built based on the data that was entered by the user. Control flow then proceeds to execute the Grover iteration.

The final stage of the quantum part of the algorithm is measuring the quantum register and storing the answer in an array. When the quantum part of Grover's algorithm has been executed N times, the system proceeds to construct a graph that displays all the answers (Figure 2).



Алгоритм Гровера

Введите пространство поиска
0
0
0
0
0
0
0
1

Количество выполнений алгоритма
100

Количество выполнений итерации Гровера
2

Figure 2 - Testing Grover's algorithm

This article presents a simulation of Grover's quantum algorithm on a classical computer. To implement the algorithm, the quantum computing library in C# - Quantum.NET, which was released in 2017, was used. This library allows you to manipulate qubits and simulate quantum circuits. It greatly simplifies the design of registers and oracles.

Thus, we can conclude that despite the absence of a full-fledged quantum computer, quantum algorithms can be described, studied and simulated on a classical computer.

LITERATURES

1. Raupova M.H. “Benefits of computerized learning systems in mathematics”, Pedagogical Acmeology, 2022, pp 133-137.

2. Ainley, J., & Pratt, D. (1995). Planning for portability: Integrating mathematics and technology in the primary curriculum. In L. Burton & B. Jaworski (Eds.), Technology in mathematics teaching: A bridge between teaching and learning (pp. 435-448). Bromley, UK: Chartwell-Bratt.

3. Rasulov X.R., Raupova M.H. General algorithm on fuzzy subclasses of k-valued logic for some issues. European Journal of Higher Education and Academic Advancement, 1:2 (2023), p. 212-215.