

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

КВАНТОВЫЕ ВЫЧИСЛЕНИЯ
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студентки 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные
технологии
факультета КНиИТ
Пригариной Валерии Александровны

Научный руководитель
доцент, к. т. н.

В. М. Соловьев

Заведующий кафедрой
к. ф.-м. н., доцент

С. В. Миронов

ВВЕДЕНИЕ

В настоящий момент квантовые вычисления являются перспективным направлением в сфере информационных технологий. Дело в том, что классические компьютеры не могут с достаточной эффективностью обрабатывать большие массивы данных, в таких случаях целесообразно прибегнуть к квантовым вычислениям. Современные компьютеры, обладая большим быстродействием, способны решать сложнейшие задачи при физико-математических расчетах. Однако существуют задачи, которые не могут быть решены на обычных классических компьютерах, основанных на полу-проводниковых элементах (в дальнейшем просто классический компьютер). Достаточно наглядным примером является задача коммивояжера. В ее условии фигурирует N городов, которые нужно объехать, вернувшись в исходный, но так, чтобы общий путь оказался минимальным. Решение этой задачи на компьютере пока возможно только путем полного перебора всех вариантов. А время, требуемое на полный перебор, с увеличением числа N растет катастрофическими темпами: при $N = 100$, по сравнению с $N = 99$, оно возрастет в 100 раз. Для решения подобных задач классическому компьютеру требуется большое количество времени либо он вообще не может решить их из-за своих технических особенностей. В основе всего здесь лежит сам принцип хранения информации в современных процессорах.

Квантовые компьютеры находят применение в криптографии, могут использоваться для решения оптимизационных задач, квантовые алгоритмы используются в машинном обучении, научных исследованиях для проведения сложных расчетов.

Целью работы является:

1. Рассмотрение принципов работы квантового компьютера.
2. Изучение парадигм программирования квантовых алгоритмов.
3. Реализация алгоритма Шора, который используется в криптографии для расшифровки шифра RSA.
4. Программная реализация модели, которая бы имитировала работу квантового компьютера.
5. Реализация алгоритма поиска Гровера.

1 Краткое содержание работы

Работа состоит из введения, 4 разделов, заключения, списка использованных источников, 4 приложений.

Первый раздел посвящен истории развития квантовых компьютеров. Только к середине 1990-х годов теория квантовых компьютеров и квантовых вычислений утвердилась в качестве новой области науки. Как это часто бывает с великими идеями, сложно выделить первооткрывателя. По-видимому, первым обратил внимание на возможность разработки квантовой логики венгерский математик И. фон Нейман.

Большое внимание к проблеме разработки квантовых компьютеров привлек лауреат Нобелевской премии по физике Р. Фейнман. Благодаря его авторитетному призыву число специалистов, обративших внимание на квантовые вычисления, увеличилось во много раз. [1]

И все же долгое время оставалось неясным, можно ли использовать гипотетическую вычислительную мощь квантового компьютера для ускорения решения практических задач. Но в 1994 году американский математик, сотрудник фирмы Lucent Technologies (США) П. Шор предложил квантовый алгоритм, позволяющий проводить быструю факторизацию больших чисел. По сравнению с лучшим из известных на сегодня классических методов квантовый алгоритм Шора дает многократное ускорение вычислений, причем, чем длиннее факторизируемое число, тем значительней выигрыш в скорости.

Прототипы квантовых компьютеров существуют уже сегодня. Множество компаний уже имеют сервисы квантовых вычислений. Так у компании IBM есть сервис IBM Quantum Experience - онлайн-платформа, обеспечивающая публичный и премиальный доступ к облачным сервисам квантовых вычислений.

Компания Google завершила расширение исследовательского центра Quantum AI. Расширенный центр позволит увеличить штат сотрудников и умножить объём исследовательских работ. Это и предыдущие разработки Google в области квантовых вычислений позволят компании представить коммерческий универсальный квантовый компьютер к 2029 году. [2]

Росатом запустил проект по созданию квантового компьютера. Проект рассчитан на срок до 2024 года, он должен объединить усилия ключевых организаций и команд, занимающихся квантовыми вычислениями в России.

Во второй главе описываются основы квантовых вычислений. Главный постулат квантовой механики состоит в том, что вся динамика любой системы определяется ее волновой функцией, которая является комплексной функцией от координат всех частиц, составляющих эту систему:

$$\psi(t, r_1, r_2, \dots, r_n)$$

Здесь r_i есть координаты частицы j (имеются в виду не только пространственные координаты частиц, но и их спиновые координаты). Эта волновая функция должна рассматриваться как вектор в гильбертовом пространстве состояний n частичной системы. Значения этой функции называются амплитудами, соответствующими пребыванию частицы в данный момент времени в таком состоянии, при котором для каждого $j = 1, 2, \dots, n$ частица j) имеет координаты r_j . Такая трактовка состояния в виде вектора сразу ведет к нетривиальному следствию: любая линейная комбинация состояний снова является некоторым возможным физическим состоянием данной системы. Таким образом, множество состояний обладает свойством линейности, и это означает, что любое уравнение, которому подчиняется вектор ψ , должно быть линейным. Этот принцип называется принципом суперпозиции, и из него вытекает существование особого процесса, называемого интерференцией амплитуд, который не имеет прямого аналога в классической физике. [3]

Гильбертово пространство, математическое понятие, обобщающее понятие евклидова пространства на бесконечномерный случай. Возникло на рубеже 19 и 20 вв. в виде естественного логического вывода из работ немецкого математика Гильберта в результате обобщения фактов и методов, относящихся к разложениям функций в ортогональные ряды и к исследованию интегральных уравнений. Постепенно развиваясь, понятие находило все более широкие приложения в различных разделах математики и теоретической физики; оно принадлежит к числу важнейших понятии математики. [4]

Интерференцию амплитуд проще всего продемонстрировать, применяя матрицы. Представим себе, что мы выбрали базис в гильбертовом пространстве состояний, и представляем всякий вектор ψ в виде некоего столбца координат этого вектора в данном базисе. Тогда из принципа суперпозиции вытекает, что состояние в следующий момент времени $t + \sigma t$ можно найти, применив к состоянию в момент времени t некий линейный оператор U , называемый оператором унитарной эволюции. [5]

Для нахождения амплитуды в некоторой точке в следующий момент времени надо просуммировать все амплитуды во всех точках в предыдущий момент, предварительно умножив их на соответствующие амплитуды перехода из этих точек в исходную. Значит, движение квантовой частицы можно представлять себе как движение некоей среды, где амплитуда в любой точке складывается из вкладов, которые вносят в эту точку движения этой частицы из всех других точек. При этом каждый вклад берется с комплексным весом, соответствующим описанному переходу из точки в точку. [3]

Квантовая частица может двигаться, вообще говоря, вдоль произвольной траектории, а не только по прямой, и ее амплитуда в любой точке есть результат суммирования амплитуд по всем путям, ведущим из каждой точки в данную. При этом вклад в сумму каждого пути получается умножением элементов матрицы эволюции I , соответствующих всем последовательным частям этого пути (мы представляем путь в виде ломаной и части - это ее звенья). Таким образом, амплитуда считается как сумма по всем путям, а вдоль каждого пути это - произведение амплитуд переходов по всем его последовательным частям.

Это правило справедливо везде, в том числе и в квантовой электродинамике где процессы описываются диаграммами. Оно в точности соответствует формуле для полной вероятности сложного события в теории вероятностей, с той лишь разницей, что в теории вероятности величины вещественные и неотрицательные, а здесь - комплексные. [6]

В третьей главе описывается структура квантового компьютера, принцип работы квантовых вентилях, а также приведена программная реализация модели квантового компьютера. Квантовый компьютер (в отличие от обычного) в качестве носителей информации использует квантовые объекты, а для проведения вычислений квантовые объекты должны быть соединены в квантовую систему. [3]

Квантовый объект — объект микромира (квантового мира), который проявляет квантовые свойства:

1. Имеет определенное состояние с двумя граничными уровнями
2. Находится в суперпозиции своего состояния до момента измерения
3. Запутывается с другими объектами для создания квантовых систем
4. Выполняет теорему о запрете клонирования (нельзя скопировать состояние объекта)

Любой объект, для которого выполняются вышеуказанные свойства и который мы можем создать и управлять, может использоваться как носитель информации в квантовом компьютере.

Квантовая система — система запутанных квантовых объектов, обладающая следующими свойствами:

1. Квантовая система находится в суперпозиции всех возможных состояний объектов, из которых она состоит
2. Нельзя узнать состояние системы до момента измерения
3. В момент измерения система реализует один из возможных вариантов своих граничных состояний

В квантовом компьютере вместо битов — кубиты. Кубиты — это квантовые частицы, у которых есть интересная особенность: кроме стандартных 0 и 1 кубит может находиться между нулём и единицей — это называют суперпозицией. Нагляднее это видно на рисунке ??.

Ещё одна особенность кубитов — зависимость значения от измерения. Это значит, что программист не узнает значение кубита до тех пор, пока его не измерит, а сам факт измерения тоже влияет на значение кубита. [7]

Именно благодаря тому, что кубит находится во всех состояниях одновременно до тех пор, пока его не измерили, компьютер мгновенно перебирает все возможные варианты решения, потому что кубиты связаны между собой. Получается, что решение становится известно сразу, как только введены все данные. Суперпозиция и даёт ту параллельность в вычислениях, которая ускоряет работу алгоритмов в разы.

Вся сложность в том, что результат работы квантового компьютера — это правильный ответ с какой-то долей вероятности. И нужно строить алгоритмы таким образом, чтобы максимально приблизить вероятность правильного ответа к единице.

Квантовые вентили воздействуют на кубит и меняют его значение. Так вентиль Адамара переводит кубит в суперпозицию. Вентиль X выполняет простое инвертирование, а вентиль Y выполняет инвертирование и изменяет относительную фазу кубита.

Модель квантового компьютера состоит из двух частей: классической [8] и квантовой. Классическая часть состоит из регистров, в которых указаны номера элементарных унитарных операций из некоторого списка U_1, U_2, \dots

простых 1-2 или 3 кубитных унитарных операторов, и указателей, то есть стрелок, которые указывают, к каким кубитам надлежит применить данный оператор. Кроме этого, классическая часть содержит два особых регистра: регистр конца вычисления и регистр вопроса к оракулу.

Квантовая часть компьютера - это лента, в ячейках которой стоят кубиты. [9] Потенциально лента не ограничена в том смысле, что к ней при необходимости можно всегда добавлять новые кубиты, инициализированные состоянием $|0\rangle$. Если лента содержит n кубит, то ее квантовые состояния заполняют 2^n мерное гильбертово пространство состояний. Общий вид состояния квантовой части компьютера таков: $|\psi\rangle = \sum_{j=0}^{N-1} \lambda_j |j\rangle$

где $N = 2^n$, а коэффициенты λ_j есть комплексные числа, называемые амплитудами соответствующих состояний $|\lambda_j\rangle$. Таким образом, мы всегда можем считать, что эволюция состояния квантовой ленты происходит в конечномерном

пространстве состояний, размерность которого N нам недоступна, хотя число кубитов n есть вполне доступная величина. [10]

Квантовый алгоритм - это классический алгоритм, задающий изменение во времени состояния классической части компьютера. [3] Вычисление на квантовом компьютере - это последовательность унитарных преобразований над состоянием квантовой части, которая задается состоянием классической части компьютера. То есть на каждом шаге j над состоянием $|\psi_j\rangle$ выполняется преобразование вида: $U \otimes V \otimes W \dots$ где данные элементарные операторы $U, V, W \dots$ коды которых стоят в регистрах классической части выполняются над теми кубитами, на которые указывают стрелки, выходящие из соответствующего регистра.

В четвертой главе рассмотрены алгоритмы Шора и Гровера, а также произведен обзор наиболее популярных языков программирования для квантовых компьютеров. Квантовый алгоритм представляет собой классический алгоритм, который задает последовательность унитарных операций (гейтов, или вентилях) с указанием, над какими именно кубитами их надо совершать. Квантовый алгоритм задается либо в виде словесного описания таких команд, либо с помощью их графической записи в виде системы вентилях (quantum gate array). [11]

Результат работы квантового алгоритма носит вероятностный характер.

За счёт небольшого увеличения количества операций в алгоритме можно сколько угодно приблизить вероятность получения правильного результата к единице. [3]

Ускорение на квантовом компьютере не связано с тактовой частотой процессора. Оно основано на квантовом параллелизме. Один шаг квантового вычисления совершает гораздо большую работу, чем один шаг классического.

Алгоритм Гровера - один из самых известных алгоритмов в квантовых вычислениях. Проблему, которую он решает, часто называют "поиском в базе данных но правильнее думать об этом с точки зрения проблемы поиска.

Любая задача поиска может быть математически сформулирована с помощью абстрактной функции $f(x)$ которая принимает элементы поиска x . Если элемент x является решением задачи поиска, то $f(x) = 1$. Если элемент x не является решением задачи поиска, то $f(x) = 0$. Задача поиска состоит в нахождении x такого, что $f(x) = 1$. [3]

Алгоритм выполняет два шага. [7] На первом переворачивается знак амплитуды вероятности, связанной с местом, которое мы пытаемся отыскать. Второй усиливает эту амплитуду вероятности.

После передачи через вентили Адамара два верхних кубита получают состояние $\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$ а нижний кубит имеет состояние $\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$. Объединенное квантовое состояние можно записать как 1

$$\begin{aligned} \frac{1}{2}|00\rangle \otimes \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle + |01\rangle \otimes \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle + \\ + |10\rangle \otimes \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle + |11\rangle \otimes \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \quad (1) \end{aligned}$$

Затем кубиты проходят через оракул .

Шаги алгоритма:

1. Для начала проинициализируем регистр из n кубит состоянием $|0\rangle$
2. Переведем каждый кубит в регистре в суперпозицию

$$|\psi\rangle = \frac{1}{N^{1/2}} \sum_{x=0}^{N-1} |x\rangle$$

3. Применим следующие операции к регистру N раз

- a) Сдвиг на -1 для элементов решения
- б) Переведем каждый кубит в регистре в суперпозицию
- в) Применить фазовый сдвиг на -1 для каждого вычисленного базиса,

за исключением $|0\rangle$

а) Переведем каждый кубит в регистре в суперпозицию

4. Измерим регистр, чтобы получить индекс элемента, который является решением с наибольшей вероятностью.
5. Проверить правильность решения.

Алгоритм Питера Шора — это квантовый алгоритм разложения чисел на простые множители, то есть факторизации. Суть алгоритма заключается в сведении задачи факторизации к поиску периода функции. Если известен ее период, то факторизация осуществляется при помощи алгоритма Евклида за полиномиальное время на классическом компьютере. Таким образом, алгоритм Шора включает в себя две части: классическую и квантовую. Квантовая часть занимается поиском периода функции, а классическая часть сначала подготавливает эту функцию, а потом проверяет период, найденный квантовой частью. Если период найден правильно, то задача будет решена. [12]

Важной частью алгоритма является вентиль, который называется вентилем квантового преобразования Фурье. Его можно рассматривать как обобщенный вариант вентиля Адамара. Фактически для одного кубита вентиль квантового преобразования Фурье в точности повторяет вентиль H . Основное отличие $H^{\otimes n}$ от матрицы квантового преобразования Фурье состоит в том, что в последнем случае элементы матрицы обычно являются комплексными числами — точнее, они являются комплексными корнями из 1. Они являются двумя возможными квадратными корнями из 1. Попытавшись отыскать корни четвертой степени из 1, мы снова получим ± 1 , если использовать только действительные числа, но при использовании комплексных чисел мы получим два других корня. В общем случае 1 имеет n комплексных корней n -й степени. Матрица квантового преобразования Фурье для n кубитов включает все комплексные корни 2^n -й степени из 1. [3]

Итак, мы знаем число N и хотим разложить его на простые сомножители p и q . Алгоритм выбирает число a , удовлетворяющее условию $1 < a < N$. Затем проверяет, имеют ли a и N общие делители, и если имеют, то можно сделать вывод, что a является множителем числа p или q . Отсюда легко завершить разложение. Если a не имеет общих делителей с N , тогда вычисляется $a \bmod(N)$, $a^2 \bmod(N)$, $a^3 \bmod(N)$, и так далее, где $a^i \bmod(N)$ означает вычисление a^i и затем взятие остатка от деления на N . Поскольку эти числа являются остатками,

они гарантированно будут меньше N . Следовательно, в какой-то момент последовательность чисел начнет повторяться. В результате будет найдено такое число r , для которого выполняется равенство $a^r \bmod(N) = a \bmod(N)$. Число r можно рассматривать как период, и именно это число находит квантовая часть алгоритма Шора. Как только будет найдено число r , классические алгоритмы смогут использовать его для разложения числа N .

Компания Microsoft разработала новый продукт — язык программирования для квантовых вычислений под названием Q#. Его предварительная версия, а также набор необходимых инструментов доступны для скачивания на официальном сайте компании. Помимо этого, предлагается симулятор, позволяющий разработчикам тестировать и отлаживать квантовые алгоритмы.

В настоящее время доступ к квантовым компьютерам имеется только у довольно узкого круга. С учётом этого Microsoft создала симулятор, который даёт возможность запуска квантовых программ. С помощью его локальной версии поддерживаются программы, использующие до 32 кубитов и 32 ГБ оперативной памяти. У симулятора существует также облачная Azure-версия, работающая с 40 кубитами.

Исходный код OpenQASM был выпущен как часть программного обеспечения IBM Quantum Information Software Kit (QISKit) для использования с квантовой вычислительной платформой Quantum Experience. OpenQASM имеет общие черты со специализированными языками программирования (такими, как Verilog), используемыми для описания структуры и поведения электронных схем. [13]

Программы QASM фактически всегда начинаются одинаково: мы определяем все биты, которые нам понадобятся — как квантовые, так и нормальные. Ниже приведен пример исходного кода OpenQASM. Программа добавляет два четырехбитовых номера.

Платформа LIQUi, созданная командой Quantum Architectures and Computation Group в Microsoft Research, включает язык программирования, алгоритмы оптимизации и планирования, а также несколько квантовых симуляторов. LIQUi может использоваться для перевода квантового алгоритма, написанного в виде программы высокого уровня на языке F# из семейства .NET Framework, в низкоуровневые команды для квантового компьютера.

LIQUi позволяет моделировать до 30 кубитов на одной машине с 32 ГБ

оперативной памяти. Платформу можно использовать для определения, выполнения и отображения в различных графических форматах квантовых схем. С помощью LIQUi можно имитировать простую квантовую телепортацию, алгоритм факторизации Шора, квантовую ассоциативную память, квантовую линейную алгебру.

QCL, или Quantum Computation Language создан Бернхардом Омером в 1998 году. Развитие языка продолжается и сейчас: существует эмулятор, который позволяет запускать квантовые программы на классическом компьютере. Конечно, эмулятор не может обеспечить ускорение квантового параллелизма; с другой стороны, он предлагает программисту некоторые полезные функции, такие как команды для проверки внутреннего состояния кубитов (что крайне трудно сделать на реальном квантовом оборудовании).

QCL заимствует синтаксис C и Java, которые иногда описываются как «императивные» языки, потому что они полагаются на прямые команды для установки и сброса значений переменных. Такие команды обычно запрещены в квантовом вычислении, поэтому основные части программы QCL работают только на классическом оборудовании. Квантовая система служит «оракулом», отвечающим на вопросы, которые могут быть заданы в формате, подходящем для вычислений кубитов. Каждый запрос к оракулу должен иметь требуемую архитектуру дымоходной трубы, но он может быть встроен в цикл во внешнем классическом контексте.

Язык Quipper был реализован в виде встроенного языка программирования, для которого основным языком является язык Haskell. Поэтому язык Quipper можно рассматривать как совокупность типов данных, комбинаторов и библиотек функций языка Haskell, однако совместно с дополнительной идиомой, то есть предпочтительным стилем написания программ на этом встроенном языке.

Язык Quipper использует расширенную схемную модель в рамках квантовых вычислений. В рамках квантовой схемы позволена как квантовые, так и классические данные и операции над ними. Квантовыми операциями можно управлять с помощью классических данных, но не наоборот. Квантовые данные могут быть явно измерены, в результате чего будут созданы классические данные. Схемная модель языка Quipper также включает в себя явно определённые области действия вспомогательных кубитов, что позволяет для

них быть использованными только в той части схемы, в которых они действительно используются. Это достигается путём позволяя явной инициализации и уничтожения кубита в цепи.

Несмотря на то, что при помощи языка Qirreg можно описывать квантовые схемы, на практике часто удобнее размышлять о квантовых программах в терминах применения гейтов к кубитам (или битам), которые содержатся в переменных. Эта процедурная парадигма лежит в основе разработки квантовых алгоритмов на языке Qirreg, и на ней строятся все остальные высокоуровневые операции. Вычисления на языке Qirreg имеют вид функций.

ЗАКЛЮЧЕНИЕ

Квантовые вычисления являются одним из самых перспективных и быстро развивающихся направлений в сфере информационных технологий. В будущем задачи, решение которых сейчас невозможно будут решаться на квантовых компьютерах с высокой эффективностью. Уже создан квантовый компьютер, который решил задачу по отбору проб гауссовских бозонов всего за пару минут, в то время как самый современный суперкомпьютер решал бы эту же задачу около двух с половиной миллиардов лет. Однако квантовые компьютеры скорее всего не смогут полностью заменить классические. Они будут использоваться для решения специфических задач.

В результате работы были рассмотрены принципы работы квантовых компьютеров. На языках Q# и C# была реализована модель, имитирующая работу квантового компьютера, реализованы квантовые алгоритмы, а именно алгоритм Шора и Гровера. Были рассмотрены базовые квантовые вентили и квантовые операции. В ходе работы рассмотрены языки программирования для квантовых компьютеров, структура квантового компьютера. Была изучена область применения квантовых компьютеров.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 *Steane, A.* Quantum computing / A. Steane // *Reports on Progress in Physics*. — 1998. — Т. 61, № 2. — С. 117–118.
- 2 *Ааронсон, С.* Квантовые вычисления со времен Демокрита / С. Ааронсон. — Москва: Альпина Паблишер, 2017.
- 3 *Бернхард, К.* Квантовые вычисления для настоящих айтишников / К. Бернхард. — Санкт-Петербург: Питер, 2020.
- 4 *Шерер, В.* Математика квантовых вычислений / В. Шерер. — Москва: Спрингер, 2019.
- 5 *Хренников, А. Ю.* Квантовая теория информации / А. Ю. Хренников. — Москва: Физматлит, 2008.
- 6 *Емельянов, В. И.* Квантовая физика. Биты и кубиты / В. И. Емельянов, Ю. В. Владимирова. — Москва: Физический факультет МГУ им. М. В. Ломоносова, 2012.
- 7 *Ожигов, Ю. И.* Квантовые вычисления, учебное пособие / Ю. И. Ожигов. — Москва: ВМК МГУ, 2003.
- 8 *Китаев, А.* Классические и квантовые вычисления / А. Китаев, А. Шень, М. Вялый. — Москва: МЦНМО-ЧеРо, 1999.
- 9 *Shemyakina, M.* Quantum computing. classification of quantum algorithms / M. Shemyakina // *Modern Science*. — 2019. — Т. 3, № 12. — С. 642–645.
- 10 *Кайе, Ф.* Введение в квантовые вычисления / Ф. Кайе, Р. Лафлам, М. Моска. — Санкт-Петербург: Питер, 2000.
- 11 *Preskill, J.* Quantum computing in the nisq era and beyond / J. Preskill // *Quantum*. — 2018. — Т. 2, № 1. — С. 78–80.
- 12 *Jurcevic, P.* Demonstration of quantum volume 64 on a superconducting quantum computing system / P. Jurcevic // *Quantum Science and Technology*. — 2021. — Т. 6, № 2. — С. 100–105.
- 13 *Моран, К.* Квантовые вычисления с использованием IBM QX / К. Моран. — Москва: Раст, 2019.