

**КВАНТОВЫЙ АЛГОРИТМ ДЛЯ ПОИСКА
САМОГО ДЛИННОГО ОТРЕЗКА ЭЛЕМЕНТОВ,
УДОВЛЕТВОРЯЮЩИХ ПРЕДИКАТУ**

К. Р. Хадиев, Т. Е. Бикмуллин (Казань)

Рассматривается задача поиска самого длинного отрезка элементов удовлетворяющих некоторому предикату. Формально, дана функция $f : \{0, \dots, n-1\} \rightarrow \{0, 1\}$ для некоторого целого положительного n . Задача состоит в том, чтобы найти ℓ и r такие, что $\ell \leq r$, $f(\ell) = f(\ell+1) = \dots = f(r-1) = f(r) = 1$; при этом нет таких ℓ', r' , что выполнялись бы следующие условия: $\ell' \leq r'$, $f(\ell') = f(\ell'+1) = \dots = f(r'-1) = f(r') = 1$ и $r' - \ell' > r - \ell$. Задача имеет множество приложений, например, в задаче обнаружения области свободной памяти максимального размера, в анализе данных и т. д.

Задача рассматривается с точки зрения квантовых алгоритмов, в частности, в широко распространенной вычислительной модели, модели запросов [1]. В качестве времени работы алгоритма в такой модели считается число запросов к памяти.

Теорема. *Для задачи поиска самого длинного отрезка, удовлетворяющего предикату, существует квантовый алгоритм со временем работы $O(\sqrt{n} \log n)$ и вероятностью ошибки не более 0.1. В то же время любой классический алгоритм (вероятностный или детерминированный) будет работать за время $\Omega(n)$.*

Нижняя оценка для классического случая строится путем сведения задачи к задаче поиска единицы в неструктурированном наборе булевских данных. Для нее известна линейная нижняя оценка [2].

Опишем основные идеи квантового алгоритма. Прежде всего, заметим, что существует квантовый алгоритм который находит минимальный и максимальный $i \in \{a, \dots, b\}$ такой, что $f(i) = 0$. Алгоритм представлен в работах [3-5] и базируется на Алгоритме Гровера [6]. Время работы такого алгоритма составляет $O(\sqrt{i-a})$ для случая минимального i , и $O(\sqrt{b-i})$ для случая максимального i .

Для некоторого заранее заданного $d > 0$, используя приведенный выше алгоритм, мы можем построить вероятностный алгоритм, который равновероятно выбирает элемент $i \in \{0, \dots, n-1\}$ и ищет ближайшее справа и слева индексы c и t такие, что $|t-i|, |c-i| \leq d$ и $f(c) = f(t) = 0$. Это можно сделать за время $O(\sqrt{d})$. Далее мы применяем к этому вероятностному алгоритму технику усиления амплитуды [7] и метод бинарного поиска для d . В результате получаем желаемую сложность алгоритма.

Работа выполнена за счёт средств субсидии, выделенной Казанскому федеральному университету для выполнения государственного задания в сфере научной деятельности, проект 0671-2020-0065.

Список литературы

1. Ablayev F., Ablayev M., Huang J. Z., Khadiev K., Salikhova N., Wu D. On quantum methods for machine learning problems part I: Quantum tools // Big Data Mining and Analytics. — 2019. — Vol. 3, No 1. — P. 41–55.
2. Bennett C. H. et al. Strengths and weaknesses of quantum computing // SIAM journal on Computing. — 1997. — Vol. 26, No. 5. — P. 1510–1523.
3. Kothari R. An optimal quantum algorithm for the oracle identification problem // 31st International Symposium on Theoretical Aspects of Computer Science. — 2014. — P. 482.
4. Durr, C., Heiligman, M., Hoyer, P., Mhalla, M. Quantum query complexity of some graph problems // SIAM Journal on Computing. — 2006. — Vol. 35, No. 6. — P. 1310-1328.
5. Kapralov, R., Khadiev, K., Mokut, J., Shen, Y., Yagafarov, M. Fast Classical and Quantum Algorithms for Online k -server Problem on Trees // CEUR Workshop Proceedings. — 2020. — 3072. — P. 287–301.
6. Grover L. K. A fast quantum mechanical algorithm for database search // Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. — 1996. — P. 212–219.
7. Brassard, G., Hoyer, P., Mosca, M., Tapp, A.. Quantum amplitude amplification and estimation // Contemporary Mathematics. — 2002. — Vol. 305. — P. 53–74.

DOI: 10.20948/dms-2022-24

КВАНТОВЫЙ АЛГОРИТМ ДЛЯ ПОИСКА КВАДРАТА НАИБОЛЬШЕГО РАЗМЕРА НА ДВУХМЕРНОЙ КАРТЕ

К. Р. Хадиев, В. С. Ремидовский (Казань)

В рамках данной работы рассматривается задача поиска самого большого квадрата в 0-1-матрице состоящая из единиц. Пусть для